



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

Dirección General de Estudios de Posgrado  
Facultad de Ingeniería de Sistemas e Informática  
Unidad de Posgrado

**Modelo de medición de la productividad para fábricas  
de software**

**TESIS**

Para optar el Grado Académico de Doctor en Ingeniería de  
Sistemas e Informática

**AUTOR**

Pedro Segundo CASTAÑEDA VARGAS

**ASESOR**

David Santos MAURICIO SÁNCHEZ

Lima, Perú

2019



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Castañeda, P. (2019). *Modelo de medición de la productividad para fábricas de software* [Tesis de doctorado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática / Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

---



Universidad Nacional Mayor de San Marcos  
Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

**SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR EN INGENIERÍA DE SISTEMAS E INFORMÁTICA**

En la Ciudad Universitaria, a los quince (15) días del mes de marzo del 2019, siendo las 19:05 horas, se reunieron en el Auditorio de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, el Jurado de Tesis conformado por los siguientes docentes:

*Dra Rosa Sumactika Delgadillo Ávila (Presidente)*  
*Dr. Glen Darío Rodríguez Rafael (Miembro)*  
*Dra. Luz Sussy Bayona Oré (Miembro)*  
*Dr. David Santos Mauricio Sánchez (Asesor)*

Se inició la Sustentación invitando al candidato a Doctor **Pedro Segundo Castañeda Vargas**, para que realizara la exposición oral y pública de la tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas e Informática, siendo la Tesis intitulada:

**“Modelo de Medición de la Productividad para Fábricas de Software”**

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

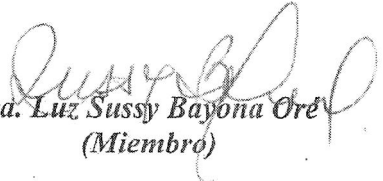
.....  
DIECISIETE (17)  
.....

Por tanto el Presidente del Jurado, de acuerdo al Reglamento de Grados y Títulos, otorga al Magister **Pedro Segundo Castañeda Vargas** el Grado Académico de Doctor en Ingeniería de Sistemas e Informática.

Siendo las 20:30 horas, el Presidente del Jurado de Tesis da por concluido el acto académico de Sustentación de Tesis.

  
**Dra. Rosa Sumactika Delgadillo Ávila**  
(Presidente)

  
**Dr. Glen Darío Rodríguez Rafael**  
(Miembro)

  
**Dra. Luz Sussy Bayona Oré**  
(Miembro)

  
**Dr. David Santos Mauricio Sánchez**  
(Asesor)

# **DEDICATORIA**

Este trabajo está dedicado a toda mi familia, en especial a mi pequeño hijo Nicolás y a mi madre María Clemencia que desde el cielo me guía e ilumina en el largo camino de la vida.

A mis hermanos y a mi amor Sandra por la paciencia y el apoyo constante en todo este tiempo dedicado al logro de este objetivo.

# AGRADECIMIENTOS

Al profesor Dr. David Santos Mauricio Sánchez, por la orientación, revisión y dedicación para que este trabajo cumpla con los objetivos trazados.

A todos los docentes del programa doctoral, por sus grandes enseñanzas que sirvieron de mucho en la elaboración de la presente tesis.

Al consejo editorial de *International Journal of Information Technologies and Systems Approach* (IJITSA) por la aceptación y publicación de los artículos de investigación que afianzaron la propuesta de tesis planteada.

A Jorge Fernández, por el apoyo en el análisis estadístico de los resultados.

A Karin Salvatierra, por la revisión lingüística.

A mis colegas y amigos del programa doctoral, por sus observaciones y porque en todo momento me incentivaron para que culmine este trabajo.

Al jurado revisor, por sus revisiones y recomendaciones de la presente tesis.

A todas aquellas personas que indirectamente me ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

Y por encima de todo doy gracias a Dios.

# TABLA DE CONTENIDO

<b>DEDICATORIA .....</b>	<b>II</b>
<b>AGRADECIMIENTOS.....</b>	<b>III</b>
<b>TABLA DE CONTENIDO .....</b>	<b>IV</b>
<b>LISTA DE FIGURAS.....</b>	<b>VII</b>
<b>LISTA DE TABLAS .....</b>	<b>VIII</b>
<b>RESUMEN .....</b>	<b>XI</b>
<b>ABSTRACT .....</b>	<b>XII</b>
<b>CAPÍTULO 1: INTRODUCCIÓN .....</b>	<b>13</b>
1.1. Antecedentes.....	13
1.2. Definición del Problema .....	17
1.3. Importancia .....	17
1.4. Motivación .....	19
1.5. Objetivos .....	21
1.6. Propuesta .....	21
1.7. Organización de la Tesis.....	22
<b>CAPÍTULO 2: MARCO TEÓRICO.....</b>	<b>24</b>
2.1. Fábricas de Software.....	24
2.2. Productividad en Fábricas de Software .....	27
2.3. Análisis Envoltorio de Datos (DEA) .....	28

<b>CAPÍTULO 3: ESTADO DE ARTE .....</b>	<b>32</b>
3.1. Introducción .....	32
3.2. Fábrica de Software .....	34
3.3. Metodología de Investigación .....	40
3.4. Planificación de la Revisión.....	40
3.5. Conducción de la Revisión.....	42
3.6. Estadísticas .....	44
3.7. Análisis .....	45
3.8. Discusión .....	66
3.9. Conclusión.....	69
<b>CAPÍTULO 4: NUEVOS FACTORES QUE INFLUYEN EN LA PRODUCTIVIDAD DE LAS FÁBRICAS DE SOFTWARE .....</b>	<b>71</b>
4.1. Introducción .....	71
4.2. Modelo Conceptual .....	73
4.3. Metodología .....	86
4.4. Resultados .....	88
4.5. Discusión .....	94
4.6. Conclusiones .....	97
<b>CAPÍTULO 5: MODELO DEA PROPUESTO PARA LA MEDICIÓN DE LA PRODUCTIVIDAD DE LA FÁBRICA DE SOFTWARE .....</b>	<b>99</b>
5.1. Introducción .....	99
5.2. Modelos de Productividad DEA.....	100
5.3. Metodología .....	105
5.4. Modelo DEA .....	107



5.5.	Fuente de Datos .....	111
5.6.	Resultados Empíricos e Interpretaciones.....	117
5.7.	Procedimiento de Medición .....	127
5.8.	Conclusiones .....	131
<b>CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS .....</b>		<b>134</b>
6.1.	Conclusiones .....	134
6.2.	Limitaciones del Trabajo.....	136
6.3.	Trabajos Futuros.....	137
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>138</b>
<b>ANEXO A: FACTORES DE LA REVISIÓN DE LA LITERATURA.....</b>		<b>152</b>
<b>ANEXO B: ENCUESTA DE INVESTIGACIÓN PARA DETERMINAR NUEVOS FACTORES QUE INFLUYEN EN PRODUCTIVIDAD .....</b>		<b>162</b>
<b>ANEXO C: RESULTADOS DEL ANÁLISIS ENVOLTORIO DE DATOS (SOFTWARE DEAP VERSIÓN 2.1) .....</b>		<b>169</b>
<b>ANEXO D: PUBLICACIONES .....</b>		<b>252</b>
<b>ANEXO D.1: CARTAS DE ACEPTACIÓN Y/O PUBLICACIÓN .....</b>		<b>253</b>
<b>ANEXO D.2: A REVIEW OF LITERATURE ABOUT MODELS AND FACTORS OF PRODUCTIVITY IN THE SOFTWARE FACTORY .....</b>		<b>257</b>
<b>ANEXO D.3: NEW FACTORS AFFECTING PRODUCTIVITY OF THE SOFTWARE FACTORY .....</b>		<b>258</b>
<b>ANEXO D.4: A MODEL BASED ON DATA ENVELOPMENT ANALYSIS FOR THE MEASUREMENT OF PRODUCTIVITY IN THE SOFTWARE PRODUCTION COMPONENT OF THE SOFTWARE FACTORY .....</b>		<b>259</b>

## Lista de figuras

Figura 1.1. Empresas acreditadas por países (Capability Maturity Model Integration [CMMI] Institute, 2018).....	18
Figura 1.2. Modelo de medición de productividad basado en DEA.....	22
Figura 1.3. Secuencia de lectura de tesis .....	23
Figura 3.1. Modelo conceptual de la productividad. Adaptado de Arcudia-Abad <i>et al.</i> (2007) .....	33
Figura 3.2. Estructura de Fábrica de Software. Adaptado de Nomura <i>et al.</i> (2006) .....	39
Figura 3.3. Proceso de revisión de literatura .....	43
Figura 3.4. Número de publicaciones sobre Productividad de Software por año y base de datos.....	44
Figura 4.1. Estructura de Fábrica de Software adaptado de Nomura <i>et al.</i> (2006).....	78
Figura 4.2. Modelo conceptual inicial de los factores que afectan la productividad .....	86
Figura 4.3. Datos de la caracterización de la muestra .....	89
Figura 4.4. Estadísticas de fiabilidad.....	89
Figura 4.5. Modelo final de factores que afectan la productividad .....	96
Figura 5.1. Parámetros para el Modelo DEA.....	107
Figura 5.2. Fábricas de software considerados en la muestra .....	114
Figura 5.3. Línea de regresión estadística para fábricas de software .....	115
Figura 5.4. Resumen de eficiencias .....	118
Figura 5.5. Resultados para SF004.....	119
Figura 5.6. Conjunto de referencias VRS para la fábrica de software .....	121
Figura 5.7. Cantidad de veces que un proyecto es referencia de otro proyecto .....	122
Figura 5.8. Proceso para la medición de la productividad.....	129

## Lista de tablas

Tabla 1.1. Técnicas/Modelos sobre Productividad de Software .....	19
Tabla 1.2. Caracterización de Modelos de Productividad de Software.....	20
Tabla 2.1. Aproximaciones al enfoque de fábrica de software .....	25
Tabla 2.2. Definición de productividad de software .....	27
Tabla 3.1. Estructura organizacional de fábricas de software.....	37
Tabla 3.2. Cadenas de búsqueda utilizada en las bases de datos.....	41
Tabla 3.3. Criterios de Selección y Exclusión.....	42
Tabla 3.4. Estudios potencialmente elegibles y seleccionados .....	43
Tabla 3.5. Distribución por tipo de publicación y base de datos.....	45
Tabla 3.6. Clasificación de factores por categoría propuestos en Wagner y Ruhe (2008a); Trendowicz (2007) .....	49
Tabla 3.7. Cantidad de factores por componente y categoría que impactan en la productividad.....	51
Tabla 3.8. Modelos de medición de la productividad de software .....	52
Tabla 3.9. Otros modelos de medición de la productividad de software, no seleccionados (Sudhakar et al., 2012).....	59
Tabla 3.10. Evaluación de Modelos de Productividad de Software .....	61
Tabla 3.11. Modelos de Medición de la Productividad de Software por Componente..	62
Tabla 3.12. Factores incluidos en los Modelos de Productividad de Software por Componente.....	63
Tabla 4.1. Componentes de la fábrica de software.....	74
Tabla 4.2. Unidades de trabajo de los componentes de la Fábrica de Software.....	76
Tabla 4.3. Nuevos factores que impactan en la productividad.....	81
Tabla 4.4. Hipótesis planteadas para la definición de factores que afectan la productividad en el desarrollo de software .....	85

Tabla 4.5. Regresión de la variable dependiente (ADI) con todas las variables independientes .....	90
Tabla 4.6. Regresión de la variable dependiente (ADI) con las variables independientes significativas.....	91
Tabla 4.7. Regresión de la variable dependiente (PRO) con todas las variables independientes .....	92
Tabla 4.8. Regresión de la variable dependiente (PRO) con las variables independientes significativas.....	92
Tabla 4.9. Regresión de la variable dependiente (TES) con todas las variables independientes .....	93
Tabla 4.10. Regresión de la variable dependiente (TES) con las variables independientes significativas.....	93
Tabla 5.1. Estudios de Proyectos de Software aplicando DEA.....	101
Tabla 5.2. Información acerca de Fábricas de Software.....	110
Tabla 5.3. Fábricas de software considerados en la muestra.....	112
Tabla 5.4. Estadísticas descriptivas para variables de entrada y salida – Fábricas de Software.....	113
Tabla 5.5. Proyectos de fábricas de software considerados en la muestra .....	116
Tabla 5.6. Estadísticas descriptivas para variables de entrada y salida – Proyectos de Fábricas de Software .....	116
Tabla 5.7. Tabla de abreviaturas.....	117
Tabla 5.8. Eficiencia DEA VRS-I – Fábrica de Software .....	118
Tabla 5.9. Eficiencia DEA VRS-I – Fábrica de Software .....	120
Tabla 5.10. Eficiencia DEA VRS-I – Proyectos de Fábrica de Software .....	121
Tabla 5.11. Eficiencia promedio DEA VRS.....	122
Tabla 5.12. Conjunto de referencias VRS para los proyectos de fábrica de software..	124
Tabla 5.13. Conjunto de referencias VRS para los proyectos de fábrica de software..	126
Tabla 5.14. Responsabilidades de los participantes en el procedimiento.....	128

Tabla 5.15. Actividades del proceso de medición de la productividad .....	130
---------------------------------------------------------------------------	-----

## Resumen

La productividad en las fábricas de software es dado por el esfuerzo realizado para la producción del software, siendo muy importante porque permite que las organizaciones logren una mayor eficiencia y eficacia en sus actividades. Uno de los pilares de la competitividad es la productividad, la cual está relacionada al esfuerzo requerido para cumplir con las tareas asignadas, sin embargo, no existe una forma estándar de medirla. En este trabajo, se presenta un modelo basado en Análisis Envoltorio de Datos (DEA, por las siglas del inglés *Data Envelopment Analysis*) para evaluar la eficiencia relativa de las fábricas de software y sus proyectos, a fin de medir la productividad en la Componente de Producción de Software de la Fábrica de Software a través de las actividades que se realizan en sus diferentes unidades de trabajo. El modelo propuesto consta de dos fases, en la cual se evalúa, respectivamente, la productividad de la fábrica de software y la productividad de los proyectos que esta realiza. Pruebas numéricas sobre 6 fábricas de software con 160 proyectos implementados en el Perú muestran que el modelo propuesto permite determinar las fábricas de software y los proyectos más eficientes.

Palabras clave: *Productividad, Fábrica de Software, Producción de Software, Benchmarking, DEA*

## Abstract

Productivity in software factories is given by the effort made for the production of software, being very important because it allows organizations to achieve greater efficiency and effectiveness in their activities. One of the pillars of competitiveness is productivity, and, it is related to the effort required to accomplish the assigned tasks, however there is no standard way of measuring it, which makes it difficult to establish policies and strategies to improve the factory. In this paper, we present a model based on *Data Envelopment Analysis* (DEA) to evaluate the relative efficiency of software factories and their projects, in order to measure productivity in the Component of Software Production of the Software Factory through the activities carried out in its different work units. The proposed model consists of two phases, in which the productivity of the software factory is evaluated respectively, and the productivity of the projects it carries out. Numerical tests on 6 software factories with 160 projects implemented show that the proposed model allows to determine the software factories and the most efficient projects.

Keywords: *Productivity, Software Factory, Software Production, Benchmarking, DEA*

# Capítulo 1: Introducción

## 1.1. Antecedentes

Medición de la productividad es un gran esfuerzo que las organizaciones de software están realizando para mejorar su proceso de desarrollo de software, con el fin de ser más eficientes y competitivos en el mercado. Siendo la premisa de que la productividad está relacionada al esfuerzo requerido para completar las tareas asignadas, es necesario establecer claramente cómo se define la productividad en la industria.

En economía, un productor es definido como un agente que transforma un conjunto de entradas (materia prima o insumos) en un conjunto de salidas (bienes y servicios) y el proceso productivo es definido por la función de producción. Un similar concepto es usado en ingeniería, donde un proceso es definido como un conjunto de tareas interrelacionadas que, juntas, transforman entradas en salidas (TechAmerica, 1999). El diseño, control y optimización de procesos de ingeniería, generalmente, requieren determinación de mediciones de desempeño como eficiencia o productividad, sin embargo, muchos modelos de medición del desempeño del proceso usualmente han derivado de observaciones empíricas y han llegado a ser inexactos.

En un negocio o en el contexto industrial, la productividad está referida a un ratio del resultado de producción entre el esfuerzo requerido (Tonkay, 2008). En el proceso de ingeniería de software, las entradas y las salidas son multidimensionales y frecuentemente dificultan la medición. En muchos casos, el esfuerzo de desarrollo ha sido medido en horas hombre como una razonable medida de esfuerzo (entrada del proceso). En su libro sobre medición de software, Fenton y Pfleeger (1997) discuten la medición de la productividad basado en la siguiente definición de productividad de software:

$$Productividad = \frac{tamaño}{esfuerzo}$$

La necesidad y motivación por medir se ha incrementado notablemente, lo cual se ve reflejado en la preocupación de las organizaciones por alcanzar mayores niveles de madurez y las consiguientes certificaciones basadas en modelos como el Modelo de Madurez de



Capacidades Integrado (CMMI, por las siglas del inglés *Capability Maturity Model Integration*) y las normas de la Organización Internacional de Normalización (ISO, por las siglas del inglés *International Organization for Standardization*).

Dada la importancia y tamaño de la industria de software, existe un gran interés de las organizaciones por mejorar su proceso de desarrollo de software, lo cual conlleva a la aplicación de metodologías y modelos que les permitan conocer las tendencias de la productividad y, en particular, determinar si la industria como un todo está mejorando en el tiempo, de tal manera que sea más eficiente y eficaz en su modelo productivo. Obviamente, esta es una pregunta compleja por alguna de las siguientes razones: primero, la productividad es difícil de medir porque la definición tradicional (ratio de salidas y entradas) requiere que se cuente con métodos objetivos de medición para ambas variables y, desafortunadamente, para el software la noción de salida (producto resultante) no es sencillo, ya que los diferentes trabajos que se han realizado a la fecha difieren sobre este concepto (algunos utilizan líneas de código, puntos de función, etc.); segundo, la productividad es impactada por un gran número de factores, los cuales muchas veces son difíciles de evaluar, por ejemplo, la dificultad de la tarea, las habilidades del equipo del proyecto, la facilidad de interacción con el cliente y el nivel de requerimientos no funcionales expresadas como confiabilidad y rendimiento; tercero, es necesario considerar las diferentes perspectivas desde la cual se mide la productividad, por ejemplo, desde el punto de vista de desarrolladores, una medida de la productividad podría ser la cantidad de código producido por el sistema de software y, en el otro lado, desde la perspectiva de usuario, esto podría ser interpretado como el grado de funcionalidad lograda por el sistema de software.

Según Premraj, Shepperd, Kitchenham, y Forselius (2005), la productividad es un aspecto muy importante para la gestión de proyectos de software y la predicción de efectividad en los costos, sin embargo, el hecho de definirla es una problemática en la industria, siendo algunas de las razones las que se indican a continuación:

- Primero, debemos considerar que existen diferentes formas de medir la productividad, así, se hallan estudios que se enfocan sobre el problema de definición de una forma efectiva para medir la producción de software, por ejemplo, en sus trabajos, Behrens

(1983) o Dale y Van der Zee (1992) hacen comentarios generales sobre el contexto del negocio para mediciones de la productividad.

- Segundo, se han realizado estudios sobre productividad para facilitar la predicción del esfuerzo, por ejemplo Jeffery y Low (1990) y Jorgensen, Indahl y Sjoberg (2003) usan la productividad para construir herramientas de predicción de esfuerzo. Otro más complejo enfoque es diseñado por Sentas, Angelis, Stamelos, y Bleris (2005) quienes usan productividad y regresión para evaluar la fiabilidad de la clasificación de costos de un proyecto dado.
- Tercero, como una forma de comparación, por ejemplo, haciendo comparaciones internacionales como Cusumano, Cormack, Kemerer, y Crandall (2003), ellos analizan la productividad y prácticas de desarrollo de software a través de cuatro regiones (India, Japón, Estados Unidos y Europa).
- Cuarto, para explorar empíricamente qué factores influyen en la productividad, se debe seguir un objetivo a largo plazo a fin de guiar a la industria del software a prácticas más efectivas.

Asimismo, muchas investigaciones se han enfocado en medir la productividad de los proyectos de desarrollo de software y no se ha llegado a un consenso sobre cuál debería ser la unidad de medida a utilizar en las entradas y salidas del proceso. De acuerdo a la definición tradicional de productividad, los diversos métodos comparan el ratio de resultados (salidas) y entradas. Sin embargo, los autores consideran que, generalmente, los métodos no toman en cuenta todas las salidas y todas las entradas involucradas en el proceso, tal como lo sugieren (Scacchi, 1995; Asmild, Paradi & Kulkarni, 2006; Moreira, Carneiro, Pires & Bessa, 2010; Yilmaz & O'Connor, 2011; Machek, Hnilica & Hejda, 2012; Cheikhi, Al-Qutaish & Idri, 2012; Khan, Ahmed & Faisal, 2014), quienes consideran que los métodos actuales de medición realizan mediciones muy simplistas de entradas y salidas. Tradicionalmente, la salida es representada por líneas de código (SLOC, por las siglas del inglés *Source Lines of Code*) o puntos de función (FP, por las siglas en inglés *Function Points*), pero esta medida es simple, porque no captura otros aspectos relevantes como calidad u otra posible salida, como reusabilidad del código producido, su complejidad, etc., los cuales no son evidentes a simple vista. Asimismo, el conjunto de entradas es frecuentemente reducido a horas hombre de trabajo, lo cual consideran que es una medida parcial de productividad, ya que no toman

en cuenta los conocimientos y habilidades de los miembros del equipo de desarrollo. También consideran que capital, materiales y servicios son generalmente omitidos en el análisis de productividad, pero el desarrollo de software es un típico activo intangible que está sujeto a amortización, lo cual representa una importante entrada del proceso; la elección del lenguaje y técnicas de programación afectan no solamente el número de SLOC o FP, sino también su calidad. Los autores consideran que estos factores podrían representar el origen de la inexactitud de la medición, ya que sesgan las estimaciones de productividad.

A fin de poder solucionar esta problemática, se ha encontrado diversos estudios que se han realizado a la fecha, los mismos que se agrupan en las siguientes categorías:

- Estudios sobre factores que afectan la productividad, los cuales han sido obtenidos a través del análisis de data histórica publicada en el ISBSG (siglas del inglés *International Software Benchmarking Standards Group*) o en algunos casos se utiliza data histórica que ha sido generada en algunos proyectos que son objetos de estudio.
- Estudios sobre modelos que miden la productividad en el desarrollo de software, los cuales están basados en diversos enfoques, tales como Análisis Envoltorio de Datos (DEA, por las siglas del inglés *Data Envelopment Analysis*) (Asmild *et al.*, 2006; Cao, Ching Gu & Thompson, 2012; Pai, Subramanian & Pendharkar, 2015), CMMI & Six Sigma (Moreira, *et al.*, 2010), Lógica Difusa (López, Kalichanin, Meda & Chavoya, 2010), Modelo de Ecuaciones Estructuradas (SEM, por las siglas del inglés *Structural Equation Modeling*) (Yilmaz & O'Connor, 2011), Reutilización de Software (Nwelih & Amadin, *et al.*, 2008), Factor Total de Productividad (Machek *et al.*, 2012), ISO 9126-4 IEEE Std.1045 (Cheikhi *et al.*, 2012); y otras técnicas estadísticas planteadas por Unluturk y Kurtel (2015).

## **1.2. Definición del Problema**

El problema de la presente tesis es determinar la efectividad con que las organizaciones vienen utilizando los recursos productivos en su proceso de desarrollo de software para el cumplimiento de sus objetivos organizacionales, por lo cual están en la búsqueda de dar respuesta a la siguiente interrogante: “¿Cómo medir la productividad en la fábrica de software?”.

## **1.3. Importancia**

De acuerdo a lo indicado por Bonilla (2011), en los estudios realizados sobre la productividad es muy común decir “hay que ser competitivos”, pero no se destaca la importancia de la productividad como componente fundamental de la competitividad.

Según Echeverry y Hernández (2005), la mayoría de los estudios la han analizado de manera agregada, y de estos no se pueden tomar recomendaciones para cada sector, ya que cada sector industrial presenta una dinámica diferente.

En 1997, E. Prescott, Premio Nobel de Economía en el 2004, enfatiza la importancia de la productividad como fuente de crecimiento, así, destaca que su estudio es insuficiente.

Según Reyes (2010), el comportamiento de esta variable explica el 80% del crecimiento económico en los países desarrollados y cerca del 40% en los del Tercer Mundo.

Paul Krugman señala que para lograr competitividad en los mercados internacionales se requiere fortalecer las economías domésticas y, por tanto, tener en cuenta la productividad sujeta tanto al capital humano como a los recursos tecnológicos de que se disponga (Krugman, 1997). Michael Porter afirma que “la productividad es, a largo plazo, el determinante primordial del nivel de vida de un país y del ingreso nacional por habitante”. Asimismo, Romo y Andel (2005) asocian la productividad y la competitividad a “la capacidad de crear un entorno que favorezca el crecimiento sostenido de la productividad y que se refleje en niveles de vida más elevados de la población. Esto incorpora niveles macro, meso y microeconómicos en un marco de integración de la economía global”.

En (Banco Interamericano de Desarrollo [BID], 2010), se plantea que el bajo crecimiento de la productividad es la raíz del deficiente crecimiento económico de América Latina.

Asimismo, el crecimiento de la industria del software ha conllevado a que las organizaciones se tornen cada vez más competitivas y agresivas en la búsqueda de la medición de la productividad y la eficiencia en ambientes de desarrollo de software, así, se pone énfasis en aumentar su eficiencia, reducir costos y sobretodo estimar el tiempo y recursos que deberán asignar a sus proyectos.

La medición de la productividad ha sido siempre una preocupación en las organizaciones, por lo cual la industria del software no es ajena a esta realidad. En la actualidad, pocos dudan de la importancia de la medición si se quiere mejorar la calidad y la productividad en el proceso de desarrollo del software. La necesidad y motivación por medir se ha incrementado notablemente con la preocupación de las organizaciones por alcanzar mayores niveles de madurez y las consiguientes certificaciones basadas en modelos y normas como ISO 9000, ISO 15504 o CMMI, tal como se puede apreciar en la Figura 1.1.

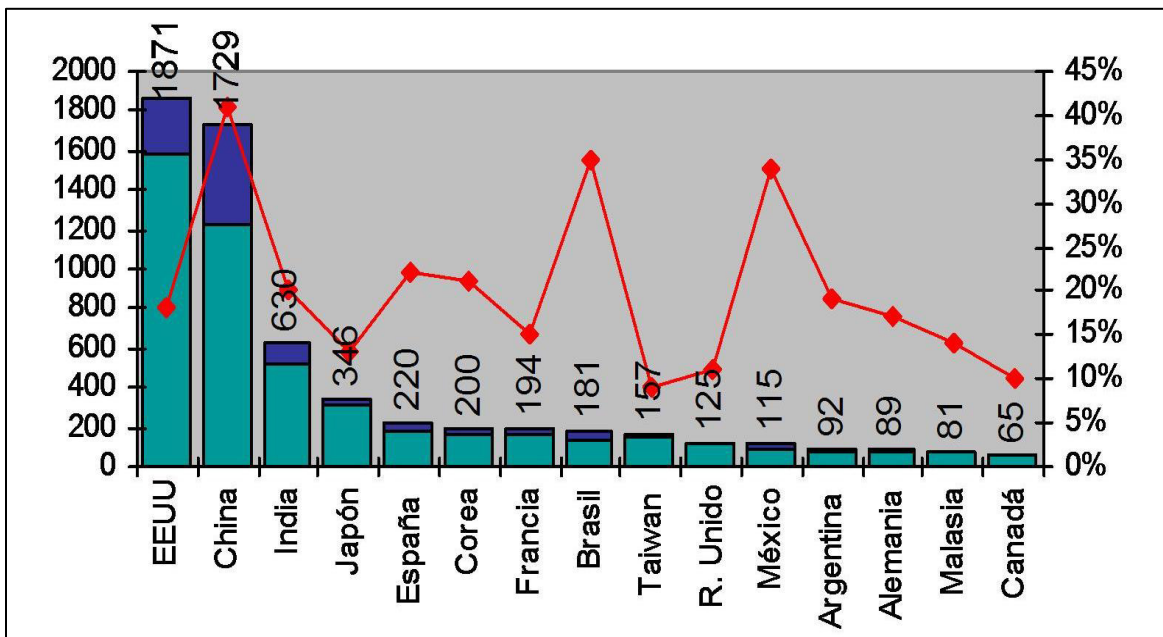


Figura 1.1. Empresas acreditadas por países (Capability Maturity Model Integration [CMMI] Institute, 2018)

Contar con un modelo de medición de la productividad permitirá una gestión eficaz y eficiente de los recursos, logrando de esta manera:

- Identificar la fábrica de software y los proyectos más eficientes.
- Identificar buenas prácticas y lecciones aprendidas en los proyectos que conllevan a un mejor desempeño en la ejecución de sus actividades.
- Diseñar estrategias para la mejora de la productividad en los proyectos de desarrollo de software.
- Realizar un benchmarking entre las fábricas de software y los proyectos implementados.

## 1.4. Motivación

En la Tabla 1.1, se muestra la recopilación de algunas de las técnicas o modelos que han desarrollado diversos investigadores sobre productividad de software durante el transcurso de los años.

Tabla 1.1.

Técnicas/Modelos sobre Productividad de Software

Id	Enfoque	Referencias
M1	Análisis Envoltorio de Datos (DEA)	Asmild <i>et al.</i> , 2006
M2	CMMI & Six Sigma	Moreira <i>et al.</i> , 2010
M3	Lógica Difusa	López <i>et al.</i> , 2010
M4	Modelo de Ecuaciones Estructurales (SEM)	Yilmaz & O'Connor, 2011
M5	Factor Total de Productividad	Machek <i>et al.</i> , 2012
M6	ISO 9126-4 IEEE Std. 1045	Cheikhi <i>et al.</i> , 2012
M7	DEA con rendimientos variables de escala (DEA VRS)	Cao <i>et al.</i> , 2012
M8	Reutilización del Software	Nwelih & Amadin, 2008
M9	DEA VRS	Pai <i>et al.</i> , 2015
M10	Técnica Estadística	Unluturk & Kurtel, 2015

A continuación, consideramos los siguientes criterios para la evaluación de los modelos presentados:

- **Formulación de la productividad.** Explicita una función para medir la productividad.

- **Cobertura del modelo.** Especifica si el modelo cubre parte o todas las unidades de trabajo de la fábrica de software (Análisis & Diseño, Programación, y Pruebas de Software).
- **Sustento teórico.** Especifica si el modelo presenta sustento o fundamento en teorías.
- **Validación.** Especifica si el modelo evaluado ha sido probado en proyectos reales de una organización o por repositorios de datos.
- **Pre-procesamiento.** El modelo describe un procedimiento para la captura y pre-procesamiento de los datos para medir la productividad.

Seguidamente, en la Tabla 1.2, se especifica los modelos de productividad y las características que cada uno de ellos cumple.

Tabla 1.2.

Caracterización de Modelos de Productividad de Software

<b>Criterio</b>	<b>M1</b>	<b>M2</b>	<b>M3</b>	<b>M4</b>	<b>M5</b>	<b>M6</b>	<b>M7</b>	<b>M8</b>	<b>M9</b>	<b>M10</b>
<i>Formulación de Productividad</i>	√	√	√	No	No	No	No	Si	No	√
<i>Cobertura del Modelo</i>	PRO	PRO - TES	PRO	PRO	PRO	PRO	AD	PRO	PRO - TES	PRO - TES
<i>Sustento Teórico</i>	√	√	√	√	√	√	√	√	√	√
<i>Validación</i>	NE	√	√	NE	NE	NE	√	NE	√	√
<i>Pre-Procesamiento</i>	NE	NE	NE	NE	NE	NE	NE	NE	√	NE

√ Si cumple el criterio establecido NE: No Especifica PRO: Programación TES: Pruebas AD: Análisis & Diseño

Tal como se puede apreciar en la Tabla 1.2, existen discrepancias entre los diferentes estudios que se han realizado hasta la fecha sobre productividad de software, siendo algunas de ellas las que se describen a continuación:

- *La mayor parte de los modelos se orientan a la unidad de trabajo de Programación.* Al medir solo una unidad de la fábrica de software, no se podría tener una medición real de la productividad, ya que existen otras unidades de importancia por analizar, como son Análisis & Diseño y Pruebas.
- *No se ha estandarizado claramente los factores que afectan la productividad.*

- *Los modelos no miden la productividad en forma comparativa.* Los modelos se basan en las siguientes teorías: modelo económico de productividad total de factores, modelos matemáticos como el uso de sistemas de ecuaciones estructurales, modelos estadísticos de regresión lineal y múltiple, modelos tradicionales como el Modelo Constructivo de Costos (COCOMO, por las siglas del inglés *Constructive Cost Model*), Six Sigma, entre otros.
- *Los modelos solo dan respuesta a la interrogante ¿Qué?, más no han establecido la forma de ¿Cómo se realizará la medición de la productividad en fábricas de software?*

## **1.5. Objetivos**

### **1.5.1. Objetivo general**

OG1: Diseñar un modelo para medir la productividad de la fábrica de software, a fin de determinar la efectividad del uso de los recursos productivos para el cumplimiento de los objetivos organizacionales.

### **1.5.2. Objetivos específicos**

OE1: Identificar los factores que afectan a la productividad de las fábricas de software y modelos para su medición.

OE2: Identificar nuevos factores que afectan en la medición de la productividad en la Componente de Producción de Software de las fábricas de software, específicamente en las unidades de trabajo de Análisis & Diseño, Programación y Pruebas.

OE3: Diseñar y validar un modelo para la medición de la productividad de las fábricas de software y sus respectivos proyectos.

## **1.6. Propuesta**

Se presenta un modelo de medición de la productividad basado en Análisis Envoltorio de Datos (DEA), el cual permite medir la productividad en la fábrica de software y en sus respectivos proyectos.



Debido a que el 90% del esfuerzo de la Fábrica de Software es absorbida por la componente de Producción de Software, y de esta las unidades de trabajo: Análisis & Diseño, Programación y Pruebas, consumen el 85% del esfuerzo de esta componente (Jacobson, Booch & Rumbaugh, 2000), el alcance de la presente tesis está orientado a la medición de la productividad en esta componente y las unidades de trabajo de Análisis & Diseño, Programación y Pruebas.

Asimismo, como parte del presente trabajo, se busca determinar los factores que afectan la productividad en las fábricas de software.

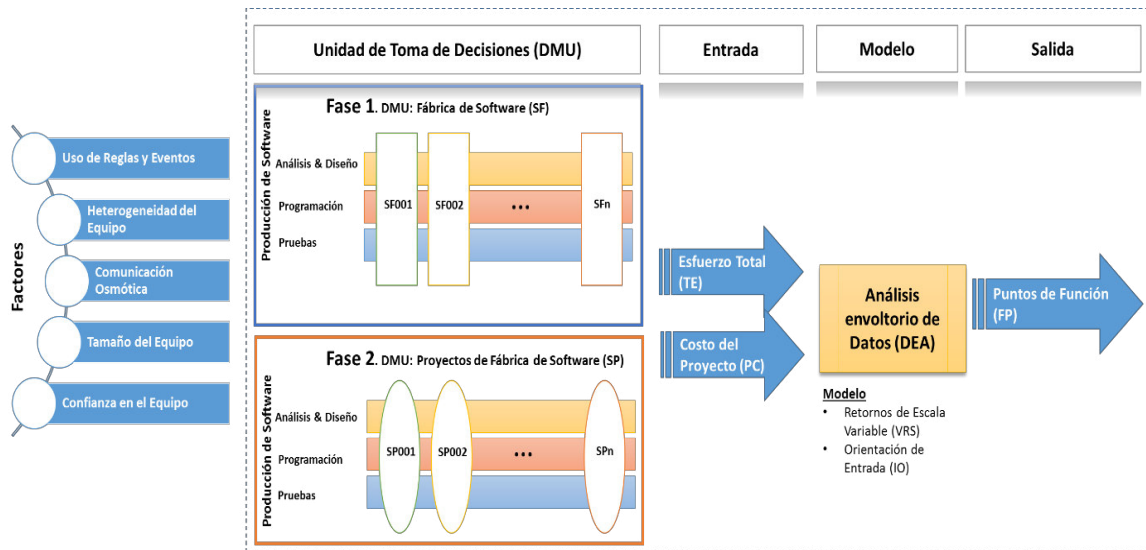


Figura 1.2. Modelo de medición de productividad basado en DEA

## 1.7. Organización de la Tesis

Esta tesis está organizada en 6 capítulos, los cuales se sintetizan a continuación:

En el Capítulo 2, se desarrolla de manera exhaustiva los conceptos fundamentales de la Fábrica de Software, la productividad y DEA.

En el Capítulo 3, se hace una revisión de la taxonomía existente sobre modelos de productividad de software, a fin de identificar los diferentes estudios que se han realizado a la fecha y ver cómo estos han evolucionado en el tiempo.

En el Capítulo 4, se presentan nuevos factores que afectan la productividad en las fábricas de software.

En el Capítulo 5, se introduce el modelo de productividad basado en DEA, desarrollándose dos aplicaciones DEA. Por un lado, se expone el caso de la medición de la eficiencia de las fábricas de software, y por otro, se busca determinar los proyectos más eficientes que sirvan como guía para los proyectos de fábricas de software.

En el Capítulo 6, se presentarán las conclusiones de esta tesis doctoral, así como las posibles líneas de investigación que quedan abiertas tras ella.

Con el fin de facilitar la lectura de la tesis se propone la secuencia de lectura descrita en la Figura 1.3.

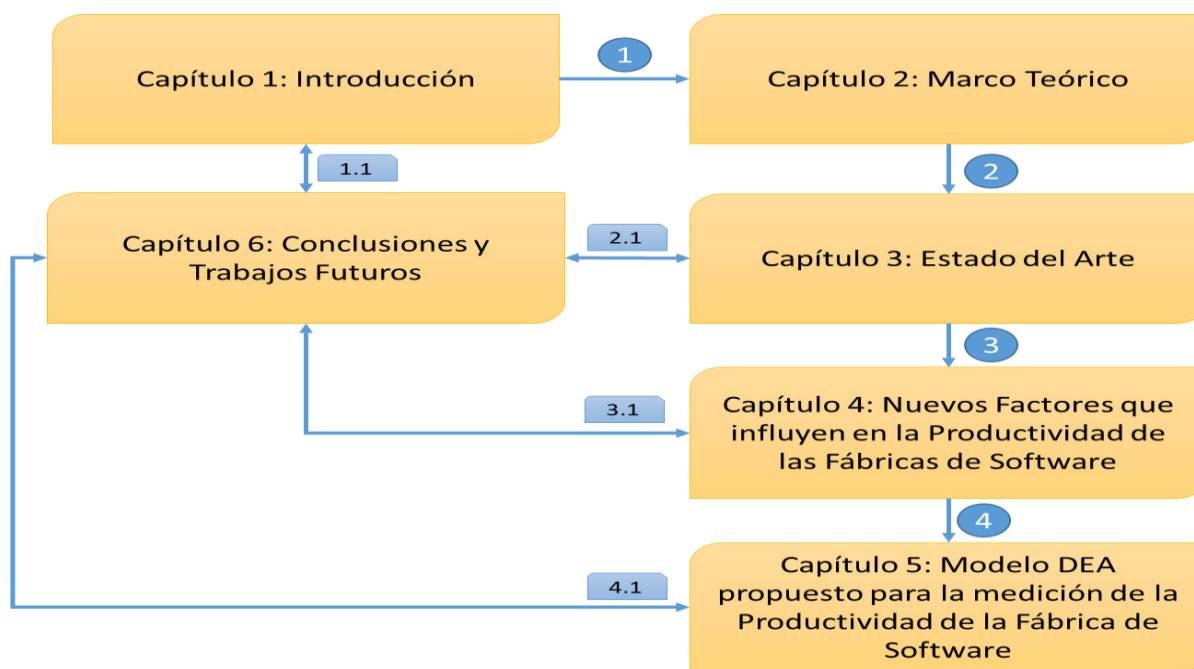


Figura 1.3. Secuencia de lectura de tesis

## Capítulo 2: Marco Teórico

### 2.1. Fábricas de Software

El término fábrica de software ha estado presente a lo largo de la evolución de la Ingeniería de Software, habiéndose establecido diversos enfoques que han estado alineados al desarrollo y aparición de nuevas metodologías y/o procesos de software. Hoy en día, este término se ha vuelto común, ya que muchas organizaciones están interesadas en ser más eficientes y es por ello que buscan adoptar este modelo que les permitiría el aprovechamiento de capacidades y habilidades, reduciendo costos y mejorando la productividad.

En Piattini & Garzás (2010) se hace una revisión de las diferentes etapas por las cuáles ha ido evolucionando el concepto de fábrica de software. Los autores mencionan que Bemer utiliza este término por primera vez en el año 1968 en el congreso IFIP (International Federation of Information Processing) definiendo que *“una fábrica, tiene más que supervisión humana. Mide y controla la productividad y calidad. Se mantienen registros financieros para coste y planificación”*. También mencionan que en 1969, Hitachi es la primera empresa que utilizó este término en la implementación de sus proyectos.

En la Tabla 2.1 se muestra las diferentes aproximaciones que Piattini & Garzás (2010) plantean en la evolución del enfoque de fábrica de software.

Tabla 2.1.

Aproximaciones al enfoque de fábrica de software

(Año) Descripción	Enfoque	Objetivo
(Años 90) <i>Fábricas basadas en Entornos de Desarrollo Integrados</i>	Uso de herramientas CASE (Piattini & Daryanani, 1995), Entornos Integrados de Desarrollo de Software (ISDE, por las siglas del inglés <i>Integrated Software Development Environments</i> ), y Entornos de Ingeniería del Software orientados al Proceso (PSEE, por las siglas del inglés <i>Process-centered Software Engineering Environment</i> ).	Producción de una arquitectura y un marco de trabajo para los ISDE.
(Años 90) <i>Fábrica de componentes basadas en experiencia</i>	Experimento de nuevas tecnologías en entornos productivos, tal como la experiencia desarrollada en el Laboratorio de Ingeniería de Software (SEL, por las siglas del inglés <i>Software Engineering Laboratory</i> ) de la NASA por Basili (Basili, 1989; Basili, 1993; Basili, Caldiera & Cantone, 1992)	Mejora de la eficacia del proceso. Reducción del reproceso Reutilización de productos
(Años 90) <i>Fábrica de software basada en la madurez de procesos</i>	Basado en el Modelo de Madurez de Capacidades (CMM, por las siglas del inglés <i>Capability Maturity Model</i> )	Creación de un entorno para la mejora de procesos de software

(Año) Descripción	Enfoque	Objetivo
(Años 90) <i>Fábrica de software basada en la reutilización</i>	Se plantea la reutilización mediante la idea de la “fábrica de software flexible” (Griss, 1993)	Reducción del reproceso de ingeniería.
(Años 90) <i>Fábricas de renovación de software</i>	Debido a la agudización del problema de mantenimiento del software (Polo, Piattini & Ruiz, 2003), surge el concepto de líneas de ensamblado (Brunekreef & Diertens, 2002). En (Van den Brand, Sellink & Verhoef, 2000) se presenta los componentes para la fábrica como son: transformadores de código, regeneradores, reestructuradores, migradores, entre otros.	Producción basado en concepto de líneas de ensamblado.
(Años 90) <i>Fábricas enfocadas en otras técnicas de gestión de la calidad</i>	Aplicación de Gestión de la Calidad Total (TQM, por las siglas del inglés <i>Total Quality Management</i> ), reutilización, generadores de código y herramientas CASE (Swanson, Kent, McComb & Dave, 1991). También se enfocan en sistemas de producción “esbeltos” ( <i>lean</i> ) y otras técnicas de gestión basadas en el tiempo (Aoyamaa, 1996).	Flexibilidad en las fábricas de software Eliminación del desperdicio en el proceso productivo.
(Años 2000) <i>Componentes, Modelos y Líneas de Productos</i>	En (Li, Li & Li, 2001) se propone un modelo de fábrica de software para organizaciones chinas, en la que se integran ISO9000, CMM y PSP/TSP (Piattini, Caballero & García, 2006). (Greenfield, Short, Cook, Kent & Crupi, 2004) conceptualizan un enfoque de desarrollo basado en componentes, el desarrollo dirigido por modelos y las líneas de producto de software.	Desarrollo basado en componentes.

## 2.2. Productividad en Fábricas de Software

La productividad en la fábrica de software es un aspecto muy importante a tener en cuenta, ya que se orienta a que las organizaciones busquen la eficiencia en el logro de sus objetivos.

La productividad es definida como la razón entre una salida que se produce y la entrada que se usa para producir dicha salida (Coelli, Rao, O'Donnell, & Battese, 2005).

En la Tabla 2.2, se presentan las definiciones sobre la productividad del software.

Tabla 2.2.

Definición de productividad de software

Definición	Referencias
<i>Productividad = (Tamaño de aplicación desarrollada)/(Mano de Obra consumida durante el desarrollo)</i>	Banker & Kauffman, 1991
<i>La relación entre producto de trabajo y esfuerzo de trabajo.</i>	Institute of Electrical and Electronics Engineers (IEEE, 1992)
<i>Productividad = (Producto de Trabajo) / (Esfuerzo de Trabajo)</i>	
<i>La relación entre el tamaño del producto y el esfuerzo de trabajo.</i>	Fenton & Pfleeger, 1997
<i>Productividad=Tamaño/Esfuerzo</i>	
<i>Productividad= Salida/Entrada</i>	Scacchi, 1995; Basili, Briand & Melo, 1996; Blackburn, Scudder & Van Wassenhove, 1996; Briand, El-Emam & Bomarius, 1998; Card, 2006; Jeffery, Ruhe & Wiczorek, 2001; Maxwell, 2001; Wagner & Ruhe, 2008; Nwelih & Amadin, 2008
<i>donde:</i>	
<i>Salida: FP, SLOC</i>	
<i>Entrada: horas-persona (PH), mes-persona (PM)</i>	

No hay un estándar en la definición de la productividad, pero a pesar de ello los autores tienen un aspecto común, ya que consideran que la productividad es el ratio del producto obtenido

entre el esfuerzo, sin embargo, los estudios realizados sobre la productividad de una fábrica de software se orientan a medir la productividad en la unidad de trabajo de programación de software, tal como se puede apreciar en las entradas y salidas que han sido evaluadas (SLOC, FP, PH por el desarrollador, PM por el desarrollador), lo cual ha conllevado a que se hayan desarrollado muchos modelos de medición del desempeño que han llegado a ser inexactos. No obstante, los estudios no han revisado de manera integral a la fábrica de software, así, obvian otras unidades de trabajo también importantes como: Análisis & Diseño y Pruebas.

En los estudios se habla muy poco de las fábricas de software, a pesar de ser un modelo que se viene trabajando en la industria hace muchos años y, por ende, no se tiene una definición clara de la productividad. Es en este contexto que definimos la productividad de las fábricas de software como “un indicador de la eficiencia de los recursos utilizados sobre las diferentes unidades de trabajo de la fábrica de software para el logro del producto final” (Castañeda & Mauricio, 2018). Se ha planteado esta definición, ya que considera todas las unidades de trabajo que participan en el desarrollo de software, y, además, permite evaluar los diferentes factores que influyen en la producción del software.

### **2.3. Análisis Envoltorio de Datos (DEA)**

DEA es una técnica de programación matemática ampliamente utilizada que fue desarrollada originalmente por Charnes, Cooper y Rhodes (1978) y fue extendido por Banker, Charnes y Cooper (1984) para incluir retornos de escala variable. DEA generaliza la medida de eficiencia técnica de Farrell (1957) para evaluar la eficiencia relativa de unidades pares con respecto a múltiples medidas de rendimiento (Charnes, Cooper, Lewin & Seiford, 1994; Cooper, Seiford & Tone, 2006). Las unidades bajo evaluación en DEA se llaman unidades de toma de decisiones (DMU). Una DMU se considera eficiente cuando ninguna otra unidad DMU puede producir más salidas usando una cantidad igual o menor de entradas. Asimismo, se definen las siguientes ventajas:

- No requiere la imposición de una función específica de producción que relacione las variables independientes con las variables dependientes. DEA asigna matemáticamente valores ponderados para las variables de entrada y de salida, lo que evita que un “experto” asigne los valores ponderados (Nyhan & Martin, 1999).

- Realiza comparaciones simultáneas de procesos con múltiples variables de entradas y salidas y produce una “eficiencia” agregada para cada institución.
- Puede calcular la cantidad de recursos que se pueden ahorrar, o la cantidad de recursos adicionales para aquellas instituciones que no son eficientes.
- Puede utilizar para determinar la eficiencia técnica y/o económica, si la información es proporcionada.

Sin embargo, también DEA tiene sus limitantes, como los que se mencionan a continuación:

- DEA no tiene indicadores para medir error.
- DEA no es apropiado para probar hipótesis.

### **2.3.1. Conceptos Fundamentales**

A continuación se explican los conceptos fundamentales sobre los cuáles se basan los modelos DEA (Villa, 2003).

#### *2.3.1.1. Unidad productiva*

Es cualquier organización que produzca consumiendo ciertos recursos, con la capacidad de poder modificar tanto los recursos consumidos (entradas) como la producción creada (salidas).

#### *2.3.1.2. Productividad*

La productividad, según Farrel (1957), de una determinada unidad productiva se define como la relación existente entre los resultados que obtiene y los recursos empleados en su producción. Es una forma de medir cómo se están aprovechando dichos recursos. Para el caso de una sola salida y una sola entrada:

$$Productividad = \frac{Producción\ creada}{Recurso\ consumido} = \frac{Salida}{Entrada}$$

En la medida de la productividad es importante destacar la dificultad de determinar los factores que son realmente relevantes a la hora de definir las entradas y las salidas de la unidad productiva. En algunos casos, los recursos o los productos no son fácilmente mensurables y a veces ni siquiera es sencillo obtener información cuantitativa sobre ellos. Para el caso de varias entradas y varias salidas la expresión matemática es:



$$Productividad = \frac{\text{Suma ponderada de salidas}}{\text{Suma ponderada de entradas}}$$

Por lo tanto, si denotamos como  $x_{ij}$  a la cantidad de entrada o recurso “i” utilizado por la unidad “j”, y como  $y_{kj}$  a la cantidad de salida o resultado “k” que produce la misma unidad, se obtiene la expresión:

$$Entrada\ virtual_j = \sum_{i=1}^m u_{ij} x_{ij}$$

$$Salida\ virtual_j = \sum_{k=1}^s v_{kj} y_{kj}$$

En donde los términos  $u_{ij}$  y  $v_{kj}$  son, respectivamente, los pesos correspondientes a cada entrada y salida, que adimensionalizan las expresiones de entrada y salida virtuales, “m” el número total de entradas consideradas y “s” el número de salidas de la unidad. Con estos nuevos conceptos se puede definir la productividad como:

$$Productividad_j = \frac{\sum_{k=1}^s v_{kj} y_{kj}}{\sum_{i=1}^m u_{ij} x_{ij}}$$

Con esta expresión se puede determinar la productividad de una unidad productiva. Sin embargo, lo interesante es tener algún índice que nos permita comparar unas unidades con otras similares, por eso debemos introducir en este punto un nuevo concepto: eficiencia relativa.

### 2.3.1.3. Eficiencia relativa

La expresión que define la eficiencia relativa es:

$$Eficiencia_j = \frac{Productividad_j}{Productividad_{max}} = \frac{Salida\ virtual_j / Entrada\ virtual_j}{Salida\ virtual_{max} / Entrada\ virtual_{max}}$$

Donde el subíndice “j” indica la unidad que se va a estudiar, y el subíndice “max” la unidad de máxima productividad. Se pueden distinguir varios tipos de eficiencias relativas en función de la unidad de referencia que se utilice.

Se habla de la eficiencia global cuando se escoge como unidad de referencia la de mayor productividad de entre las que están en estudio. La eficiencia técnica se utiliza cuando se elige como unidad de referencia la de mayor productividad de entre las unidades de su tamaño. Se define la eficiencia de escala como el cociente entre la eficiencia global y la eficiencia técnica.

## Capítulo 3: Estado de Arte

En el presente capítulo hacemos una revisión de la literatura sobre los factores que influyen en la productividad de la fábrica de software. Para este fin, revisamos el concepto de fábrica de software y productividad, y seguimos una metodología de tres fases, lo cual nos permite identificar aspectos a considerar en la medición de la productividad: factores que influyen y modelos para medir la productividad.

### 3.1. Introducción

Hernández, Colomo y García (2012) indican que en la ingeniería de software la medición de la productividad se ha centrado en la productividad de la entrega del producto, quizá influenciada en parte por las fórmulas utilizadas para estimar los proyectos de software. La estimación del esfuerzo de desarrollo de software es el proceso de predecir la cantidad de esfuerzo más realista requerida para desarrollar o mantener el software, mientras que, según IEEE (1992), la productividad del software se define como la relación entre el producto de trabajo y el esfuerzo de trabajo.

Existe un gran interés de las organizaciones por medir la productividad de la fábrica de software, ya que permitirá tener un panorama claro de la capacidad para utilizar sus recursos disponibles con el fin de producir bienes o servicios rentables que permitan dar satisfacción a las necesidades de sus clientes. Medir la productividad es medir el rendimiento de una organización, tanto interna como externa con el fin de evaluar su progreso.

Al medir la productividad, una organización cuenta con indicadores que le permiten compararse en el mercado, a fin de plantearse acciones que incrementen su eficiencia global, lo que le permitirá utilizar todos los recursos de una manera eficaz y eficiente para obtener los mejores resultados posibles.

En otras palabras, los directivos, necesitan conocer lo que están haciendo, comparándose con el desempeño de períodos anteriores, surgiendo preguntas como: ¿Se está incrementando, disminuyendo, avanzando o se está retrocediendo? ¿Cuál es la magnitud de ese avance o de ese retroceso? ¿Son eficaces los programas?

De acuerdo a Petersen (2011), el enfoque del proceso de mejora del software frecuentemente es mejorar la productividad, lo cual implica realizar una adecuada medición, siendo por ello necesario no solo recopilar información de diversas fuentes y luego establecer resultados, sino que demanda un mayor esfuerzo en la comprensión de los diferentes elementos que la conforman, tal como se puede visualizar en el modelo conceptual para la medición de la productividad de Arcudia-Abad, Solís-Carcaño y Cuesta-Santos (2007), que se muestra en la Figura 3.1. El proceso de productividad está compuesto de diversos tipos de factores (entrada, salida, contexto, proceso) que la influyen, siendo por ello determinante homogeneizar la unidad de medida a utilizarse, de tal manera que los datos obtenidos permitan diseñar indicadores de medición que brinden resultados coherentes y comparables bajo diferentes entornos, que puedan conllevar a establecer adecuadas estrategias de mejora del proceso.

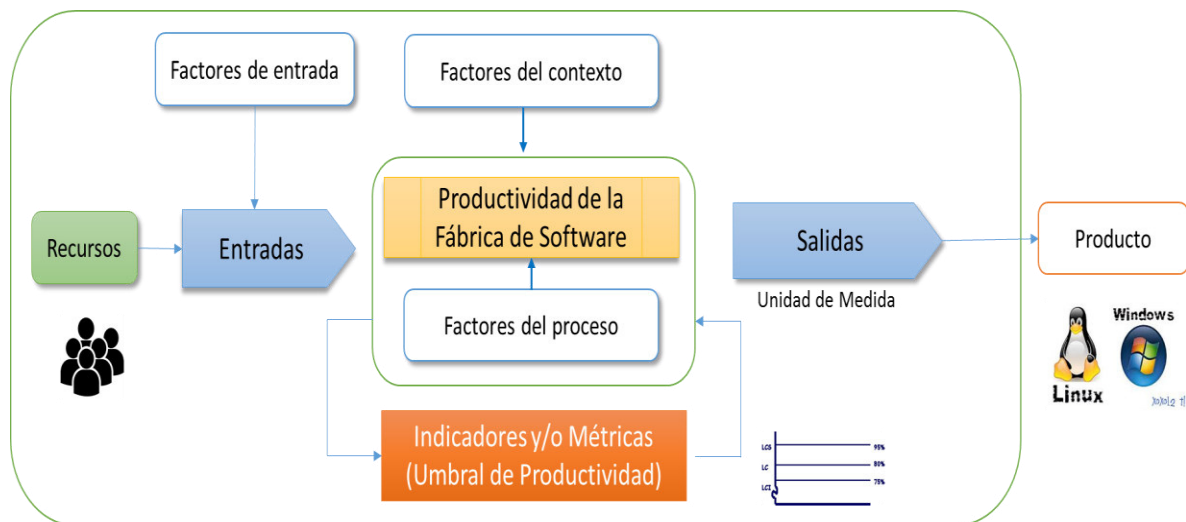


Figura 3.1. Modelo conceptual de la productividad. Adaptado de Arcudia-Abad *et al.* (2007)

Es preciso mencionar que antes de iniciar el proceso de medición de la productividad es necesario que se establezca el objetivo de la medición, ya que ello permitirá establecer el nivel en el cual se quiere realizar la medición; por ejemplo, se podría requerir medir la productividad de la fábrica de software para realizar una comparación con el mercado, o se podría medir la productividad en solo algunos componentes de la fábrica de software para verificar posibles falencias que estaría generando cuellos de botella en la producción final, o se podría medir la productividad de los roles que participan en el proceso de desarrollo de software.

Medir la productividad es una actividad compleja porque no hay un consenso sobre qué medir, ya que incorpora diversos factores que en muchos casos no son tomados en cuenta (Scacchi, 1995; Asmild, Paradi & Kulkarni, 2006; Moreira *et al.*, 2010; Yilmaz & O'Connor, 2011; Machek *et al.*, 2012; Cheikhi *et al.*, 2012; Khan *et al.*, 2014). También, los diversos estudios no han definido el ámbito de medición, dificultando la detección de oportunidades de mejora en las organizaciones, por lo cual nos planteamos la siguiente pregunta de investigación: ¿Cómo se viene midiendo la productividad en las fábricas de software?

En el presente capítulo presentaremos una revisión sistemática de los diferentes modelos que miden la productividad en las fábricas de software, así como de los factores que la afectan, pero a su vez será clasificado de acuerdo a la unidad de producción de la fábrica de software en la cual interviene, de tal manera que permita a los investigadores sobre esta temática contar con información actualizada sobre los diferentes estudios realizados a la fecha.

### **3.2. Fábrica de Software**

De acuerdo a estudios realizados por Clements y Northrop (2001), el concepto de fábrica de software trae grandes ventajas para las áreas de desarrollo de software, tales como:

- Disminución en costos de producción por producto hasta en un 60%; ahorro del tiempo de puesta en el mercado hasta en un 98%; y reducción del requerimiento en mano de obra hasta en un 60%.
- Mejoramiento de la productividad en alrededor de 10 veces y la calidad de cada producto derivado en hasta 10 veces menos errores, lo que aumenta así el portafolio de productos y servicios ofrecidos por una fábrica de software, y la posibilidad de ganar nuevos mercados.

El término Fábrica de Software ha sido usado desde los años 60 y 70 en Estados Unidos y Japón. Cusumano la define como “una empresa productora de software que responda a características como: producción de software en gran escala, estandarización de tareas, estandarización del control, división del trabajo, mecanización y automatización, puede ser considerada una factoría de software. El desarrollo de una factoría implica que las buenas prácticas de Ingeniería de Software sean aplicadas sistemáticamente” (Cusumano, 1989, p. 23).

Basili *et al.* (1992) propone dividir la factoría de software en dos sub-áreas: la de producción de software y la de producción de componentes. La sub-área de producción de software realiza las solicitudes de productos (componentes para la producción del software), de datos (estadísticas para la estimación de costos y plazos) y de planos (modelos, métodos para el análisis y diseño del software). La sub-área de producción de componentes posee una base de componentes reutilizables, los cuales pueden ser también estadísticas y datos históricos de los cuales se apoya para dar respuesta a las solicitudes hechas por la sub-área de producción de software.

Li *et al.* (2001) organiza la factoría de software basado en el Modelo Integrado de Madurez de Capacidades (CMMI, por las siglas del inglés Capability Maturity Model Integration) y la Organización Internacional de Normalización 9001 (ISO, por las siglas del inglés International Organization for Standardization) en cinco entidades: técnicas, proceso, trabajadores involucrados, gestión de la factoría y activos del proceso, herramientas y componentes de código.

En Kruchten (2004), el Proceso Unificado de Rational (RUP, por las siglas del inglés *Rational Unified Process*) describe el proceso de desarrollo de software en términos de disciplinas, estructurándolos en Modelado de Negocios, Requisitos, Análisis y Diseño, Implementación, Pruebas y Despliegue; y procesos de soporte, tales como Configuración y Gestión del Cambio, Gestión del Proyecto y Entorno.

Rockwell y Gera (1993) adapta el modelo Eureka (proceso de desarrollo de software distribuido) para el desarrollo de componentes en dos capas (servicios e interfaz), donde establece un desarrollo de software por capas. En Fernandes y Teixeira (2004), se propone un modelo que clasifica la factoría de acuerdo al alcance o fases de desarrollo definidas en el proceso, así, se ofrece una idea sobre cuál debe ser el ciclo de vida de un proyecto, clasificándolas en Factoría de Proyectos Ampliada, Factoría de Proyectos de Software, Factoría de Proyectos Físicos y Factoría de Programas.

En Yanosky (2005), se plantea que una fábrica de software debe poseer un modelo de organización de la producción, una unidad de producción de componentes y una unidad de producción de software. Sus deficiencias radican en que no se contempla la gestión de proyecto y no utiliza estándares de calidad.

En Fabri, Trindade, Begosso, Lerario, Silveira y Pessoa (2004); Fabri, Trindade, L'erário, Pessoa y Spinola (2004a); Fabri, Trindade, Durscki, Spinola y Pessoa (2005); Fabri, Trindade, Begosso, Lerario y Pessoa (2007); Fabri, Trindade, Silveira y Pessoa (2007a); Fabri, Scheible, Moreira, Trindade, Begosso, Braun y Pessoa (2007b); Fabri, Trindade, Begosso, Pessoa y L'erário (2007c); y Pessoa, Fabri, L'erário, Spinola y Begosso (2004) definen el proceso de producción de software como un proceso con características de fabricación, así, establecen que una fábrica de software posee dos unidades: producción de software y producción de componentes.

Nomura, Spinola, Hikage y Tonini (2006) plantean una estructura de fábrica de software basada en los conceptos de reutilización de la Fábrica de Experiencia de Basili *et al.* (1992); la gestión operacional y división organizacional de la Fábrica de Software de Fernandes y Teixeira (2004); la aplicación de actividades de Ingeniería de Software citada por Swanson, McComb, Smith y McCubbrey (1991); y en la mejora de los métodos de trabajo citada por Cusumano (1989), así como prácticas de ingeniería mencionadas en la Guía de Fundamentos para la Dirección de Proyecto (PMBOK, por las siglas del inglés *Project Management Body of Knowledge*) y conceptos y terminologías de RUP.

Trujillo (2007) propone un modelo que considera los principales elementos a definir para establecer un proceso de desarrollo con base en una metodología acorde a la magnitud del proyecto y al personal con que cuenta el mismo.

En la Tabla 3.1, se hace resume acerca de los estudios sobre la estructura organizacional de la fábrica de software realizado a la fecha y se realiza una analogía entre los diferentes modelos propuestos.

Tabla 3.1.

Estructura organizacional de fábricas de software

Autor	Unidad de Trabajo							
<i>Basili, et al., 1992</i>	Producción de Software							Producción de Componentes
<i>Li, et al., 2001</i>	Gestión de la Factoría y Activos del Proceso	Proceso y Trabajadores Involucrados					Técnicas, Herramientas y Componentes de Código	
<i>Rockwell &amp; Gera, 1993</i>	Servicios e Interfaz							Configuración y Gestión del Cambio, Despliegue y Entorno
<i>Kruchten, 2004</i>	Project Management	Modelado de Negocio	Requisitos	Análisis & Diseño	Implementación	Pruebas		Configuración y Gestión del Cambio, Despliegue y Entorno
<i>Fernandes &amp; Teixeira, 2004</i>	Arquitectura de Solución	Proyecto Conceptual	Especificación Lógica	Proyecto Detallado	Construcción y Pruebas	Pruebas de Integración	Pruebas Aceptación	Configuración y Gestión del Cambio, Despliegue y Entorno



Autor	Unidad de Trabajo		
<i>Yanosky, 2005</i>	Organización de la Producción	Producción de Software	Producción de Componentes
<i>Nomura, et al., 2006</i>	Project Management	Producción de Software	Soporte
<i>Trujillo, 2007</i>		Fases de Metodología de Desarrollo de Software	
<i>Fabri et al., 2004; Fabri et al., 2004a; Fabri et al., 2005; Fabri et al., 2007; Fabri et al., 2007a; Fabri et al., 2007b; Fabri et al., 2007c; Pessoa et al., 2004</i>		Producción de Software	Producción de Componentes

De la Tabla 3.1, se puede mencionar que el modelo de Nomura *et al.* (2006) plantean una estructura de fábrica de software, que cubre las distintas disciplinas del desarrollo de software, tal como se visualiza en la Figura 3.2.

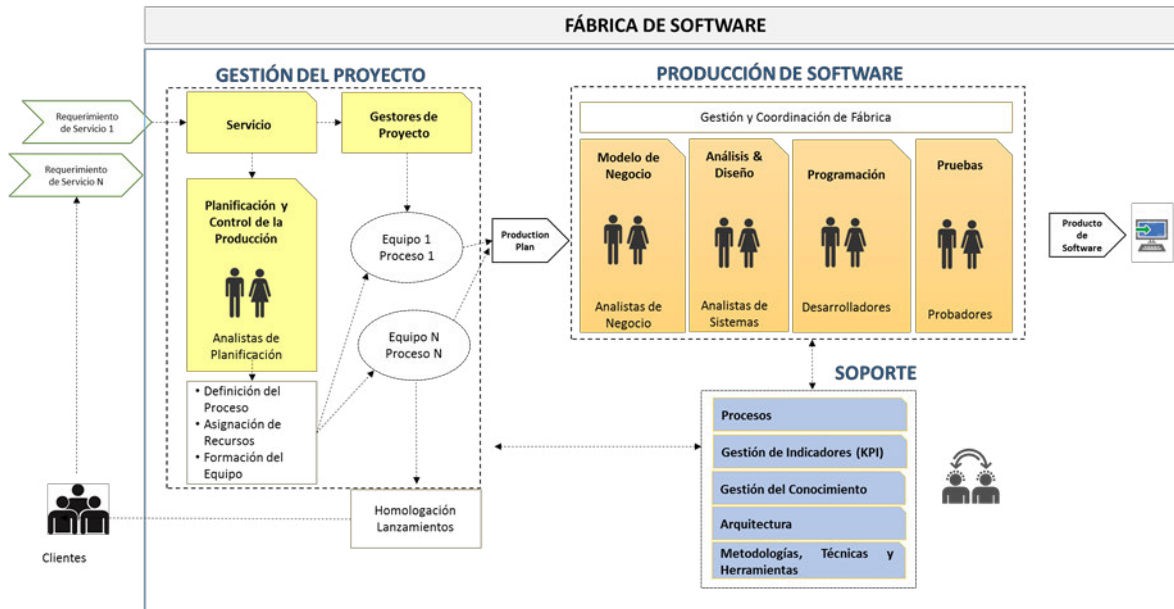


Figura 3.2. Estructura de Fábrica de Software. Adaptado de Nomura *et al.* (2006)

Tal como plantean Nomura *et al.* (2006), una fábrica de software tiene tres grandes componentes: Gestión de Proyecto, Producción de Software y Soporte. El componente de Gestión del Proyecto es responsable de la relación con el cliente, negociación, planificación, elaboración y gerenciamiento de contratos de acuerdo con el planeamiento estratégico de la empresa, así como de la gerencia, elaboración, programación y distribución de las órdenes de servicio y asignación de los recursos necesarios para la ejecución del servicio, estructurándose en las siguientes unidades de trabajo: Servicio, Gestión de Proyectos y Planificación y Control de la Producción. El componente de Producción de Software es el principal, en el cual se desarrollan todas las actividades propias del proceso de desarrollo, la misma que está estructurada en las siguientes unidades de trabajo: Modelo de Negocio, Análisis & Diseño, Programación y Pruebas. Asimismo, el componente de Soporte, proporciona apoyo para la integración y desarrollo de actividades del proceso de desarrollo, incluyéndose la gestión de procesos, la infraestructura y el soporte, el diseño de interfaces, la seguridad, la documentación, entre otros.

El presente trabajo estará orientado a la búsqueda de estudios que se han realizado en la componente de Producción de Software, principalmente en las unidades de trabajo de Análisis & Diseño, Programación y Pruebas.

### **3.3. Metodología de Investigación**

La revisión sistemática considera el procedimiento propuesto por Kitchenham y Charters (2007), el mismo que ha sido adaptado para desarrollar el presente trabajo e involucra las siguientes fases:

- Planificando la revisión: en esta etapa se plantea las preguntas de investigación y se define el protocolo de revisión.
- Conduciendo la revisión: en esta etapa se ejecuta el plan y se selecciona los estudios primarios de acuerdo a los criterios de inclusión y exclusión establecidos para el efecto.
- Reportando la revisión: se presenta los resultados estadísticos y el análisis de los documentos encontrados y seleccionados, los mismos que son presentados posteriormente.

### **3.4. Planificación de la Revisión**

Para contestar la pregunta principal de investigación, nos planteamos las siguientes preguntas sobre productividad en las fábricas de software:

- Q1: ¿Qué es la productividad de fábricas de software?
- Q2: ¿Qué factores influyen en la medición de la productividad y cómo se clasifican?
- Q3: ¿Qué modelos se han desarrollado?
- Q4: ¿Qué factores incluyen los modelos?

Se ha considerado las bases de datos IEEE Explore, ACM Digital Library, ScienceDirect, Springer, DOAJ (Directory of Open Access Journals), Taylor & Francis y otros buscadores (SCOPUS, Web of Knowledge). Los trabajos seleccionados responden a las cadenas de búsqueda dadas en la Tabla 3.2, que han sido aplicados al título, resumen y palabras claves,

y en el período comprendido de enero 2005 a abril 2017. Además, estos términos fueron adaptados para que coincidan con cada una de las preguntas de investigación y las necesidades individuales de cada uno de los buscadores.

Tabla 3.2.

Cadenas de búsqueda utilizada en las bases de datos

Bases de Datos		Cadena de Búsqueda
<i>IEEE Explore</i>		((("software factory" OR "software process" OR "software development" OR "software development process" OR "software engineering")) AND (productivity OR "development efficiency" OR "development effectiveness" OR "development performance" OR "models performance" OR "project performance" OR "productivity model" OR "productivity factors")) and refined by Content Type: Conference Publications Journals & Magazines Year: 2005-2017
<i>ACM Library</i>	<i>Digital</i>	"query": {(+"software +factory" +OR +"software +process" +OR +"software +development" +OR +"software +development +process" +OR +"software +engineering") +AND +(productivity +OR +"development efficiency" +OR +"development effectiveness" +OR +"development performance" +OR +"models performance" +OR +"project +performance" +OR +"productivity +model" +OR +"productivity +factors"))} "filter": {"publicationYear":{ "gte":2005 }}, {owners.owner=ACM
<i>ScienceDirect</i>		("software factory" OR "software process" OR "software development" OR "software development process" OR "software engineering") AND (productivity OR "development efficiency" OR "development effectiveness" OR "development performance" OR "models performance" OR "project performance" OR "productivity model" OR "productivity factors" OR "efficiency" OR "effectiveness" OR "performance")
<i>Springer</i>		
<i>DOAJ</i>		
<i>Taylor &amp; Francis,</i>		
<i>SCOPUS, Web of Knowledge</i>		

También se han considerado los criterios de selección y exclusión establecidos en la Tabla 3.3. Con respecto a las fuentes de búsqueda, se han incluido revistas, conferencias y actas.

Tabla 3.3.

Criterios de Selección y Exclusión

<b>Criterios de Selección</b>	<b>Criterios de Exclusión</b>
<ul style="list-style-type: none"> <li>• Publicados en fuentes con factor de impacto SJR</li> <li>• Presentan modelos, factores, técnicas y/o metodologías sobre productividad de software</li> <li>• Responden directamente a las preguntas de investigación</li> </ul>	<ul style="list-style-type: none"> <li>• Estudios orientados a costos y estimación de esfuerzo</li> <li>• Artículos que mencionan productividad de software, pero no aportan al tema planteado</li> <li>• Artículos que no están dentro del contexto de fábrica de software</li> <li>• Libros, tesis, posters, relatorios técnicos</li> </ul>

### **3.5. Conducción de la Revisión**

Los estudios primarios potenciales identificados en el proceso de búsqueda de acuerdo a la estrategia planteada fueron sometidos a un proceso de selección considerando los criterios de inclusión y exclusión establecidos; para ello fue necesario realizar una revisión previa a su contenido para poder determinar su relevancia para el presente estudio y, principalmente, para determinar si estos estudios se aplican a los aspectos de la productividad de fábricas de software; la mayoría de los artículos fueron descartados porque correspondían a la estimación de costos y esfuerzos para el desarrollo de software, a los factores para la estimación de esfuerzo, o artículos que no tenían factor de impacto SJR. El proceso aplicado se representa en la Figura 3.3. Luego, se procedió a analizar los artículos para contestar las preguntas de investigación.

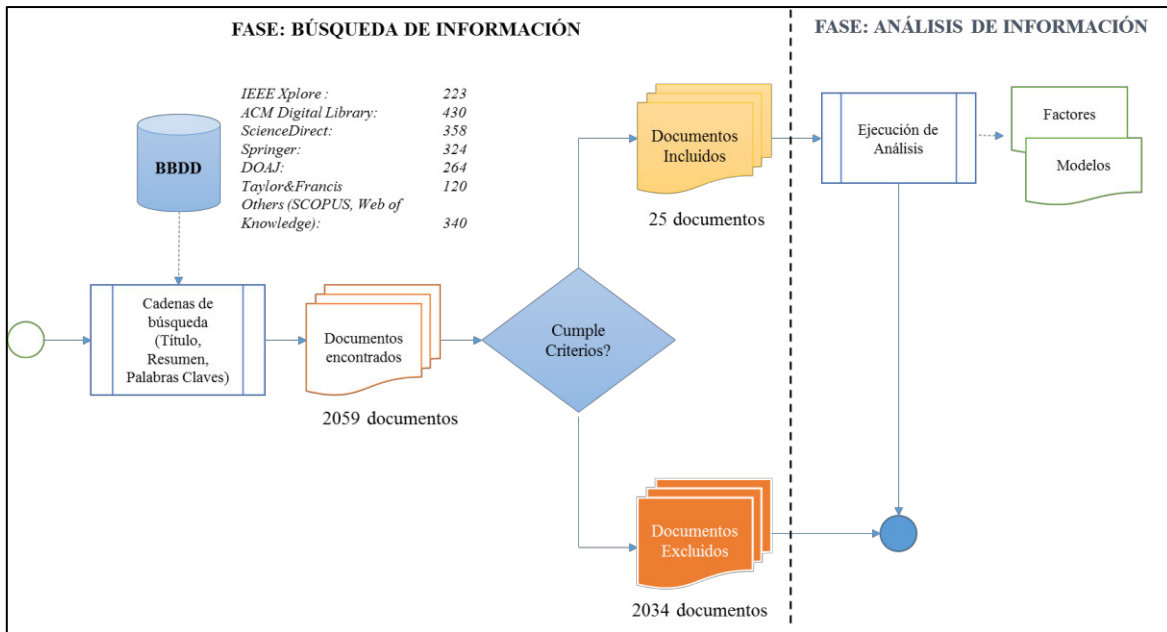


Figura 3.3. Proceso de revisión de literatura

El desarrollo de la revisión sistemática dio como resultado 2059 artículos, de los cuales 25 investigaciones fueron seleccionadas de acuerdo a los criterios de inclusión y exclusión (Ver Tabla 3.4).

Tabla 3.4.

Estudios potencialmente elegibles y seleccionados

Base de Datos	N.º de estudios potencialmente elegibles	N.º de estudios seleccionados
<i>IEEE Xplore</i>	223	1
<i>ACM Digital Library</i>	430	2
<i>ScienceDirect</i>	358	5
<i>Springer</i>	324	5
<i>DOAJ</i>	264	3
<i>Taylor &amp; Francis</i>	120	0
<i>Others</i>	340	9
<b>TOTAL</b>	<b>2059</b>	<b>25</b>

De la tabla anterior, ScienceDirect y Springer son las que contienen la mayor cantidad de publicaciones, la misma que concentra el 40%. Asimismo, en Taylor & Francis, no se seleccionaron artículos debido a que no cumplían con los criterios de selección y exclusión.

### 3.6. Estadísticas

#### 3.6.1. Tendencia temporal de publicaciones

Si revisamos los estudios de productividad de software que se han realizado a través de los años, se puede notar, en la Figura 3.4, que se han venido realizando publicaciones en las diferentes bases de datos, lo cual demostraría el interés que los investigadores han dado al tema de la productividad en la industria del software y, en particular, a la fábrica de software en las empresas.

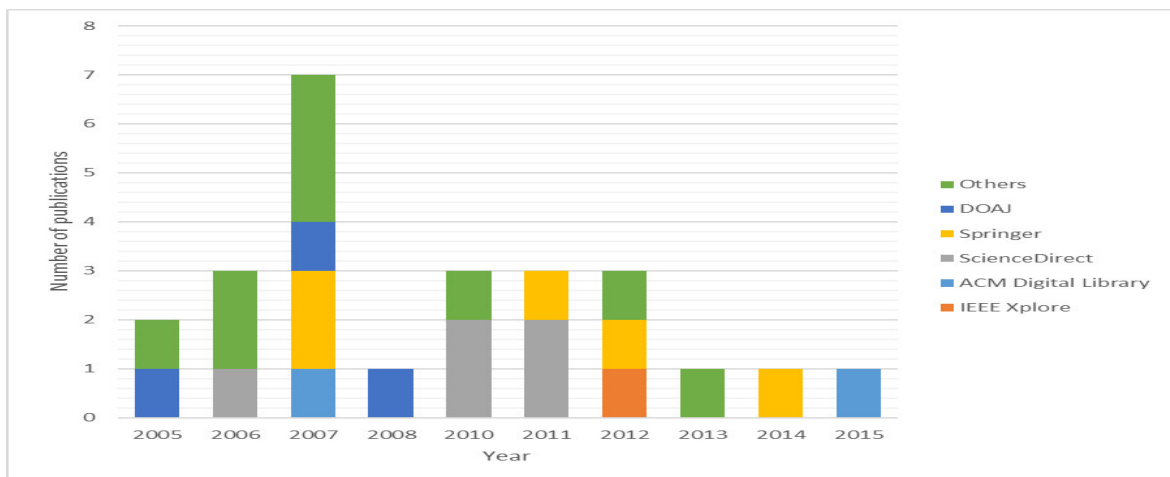


Figura 3.4. Número de publicaciones sobre Productividad de Software por año y base de datos

#### 3.6.2. Número de Publicaciones seleccionados

En la Tabla 3.5, se muestra la distribución de los estudios por tipo de publicación y base de datos. Observe que la mayor cantidad de publicaciones se dan en revistas y, en cuanto a base de datos, se han realizado mayores publicaciones en ScienceDirect y Springer, con 40% del total de publicaciones y 48% de las publicaciones corresponden a actas y conferencias.

Tabla 3.5.

Distribución por tipo de publicación y base de datos

Tipo de Publicación	Base de Datos						Total
	IEEE Xplore	ACM Digital Library	ScienceDirect	Springer	DOAJ	Others	
<i>Revistas</i>	1	1	4	2	2	3	<b>13</b>
<i>Actas</i>	0	1	1	2	0	4	<b>8</b>
<i>Conferencias</i>	0	0	0	1	1	2	<b>4</b>
<i>Total</i>	<b>1</b>	<b>2</b>	<b>5</b>	<b>5</b>	<b>3</b>	<b>9</b>	<b>25</b>

### 3.7. Análisis

#### 3.7.1. Q1: ¿Qué es la productividad de fábricas de software?

Banker y Kauffman (1991) establecen que la productividad de software es definida por la siguiente relación:

Productividad = (Tamaño de aplicación desarrollada) / (Mano de obra consumida durante el desarrollo)

De acuerdo a la IEEE (1992), se define a la productividad de software como la relación entre el producto del trabajo y el esfuerzo de trabajo. En Fenton y Pfleeger (1997) se discute la medición de la productividad basado en la siguiente definición de productividad de software:

Productividad=Tamaño/Esfuerzo

El concepto que muchos autores (Scacchi, 1995; Basili *et al.*, 1996; Blackburn *et al.*, 1996; Briand *et al.*, 1998; Jeffery *et al.*, 2001; Maxwell, 2001; Wagner & Ruhe, 2008a; Nwelih & Amadin, 2008) toman en cuenta para medir la productividad es el que está basado en el modelo CRS, para lo cual se ha definido que:

Productividad=Salida/Entrada

Donde *salida* está representado por puntos de función (FP) y líneas de código (SLOC), mientras que *entrada* es el esfuerzo del proyecto en hora persona (PH) o mes persona (PM).



De acuerdo a Wagner y Ruhe (2008a), el número de SLOC escrito o el número de FP implementados por hora persona del desarrollador es la medida de la productividad.

También, Card define la productividad como “*la relación entre los productos producidos y los recursos consumidos*” (Card, 2006, p1).

En síntesis, los estudios realizados sobre la productividad de una fábrica de software se orientan a medir la productividad en la unidad de trabajo de programación de software, tal como se puede apreciar en las entradas y salidas que han sido evaluadas (SLOC, FP, PH del desarrollador, entre otros), lo cual ha conllevado a que se hayan desarrollado muchos modelos de medición del desempeño que han llegado a ser inexactos. Sin embargo, los estudios no han revisado de manera integral a la fábrica de software, ya que su estudio considera, además, otras unidades de trabajo como Análisis & Diseño y Pruebas, y su productividad depende de diversos factores que influyen en el proceso, contexto, entre otros (Arcudia-Abad *et al.*, 2007; Nomura *et al.*, 2006).

### 3.7.2. Q2: *¿Qué factores influyen en la productividad y cómo se clasifican?*

Premraj *et al.* (2005) muestran que el Sector de Negocio influye en la productividad de las personas, siendo los sectores más influyentes la Banca, las Telecomunicaciones y los Seguros.

Moreira *et al.* (2010) plantean un modelo de desempeño basado en CMMI - Six Sigma y su aplicación para la predicción de la productividad en proyectos de una organización de software, considerándose en el modelo los siguientes factores: *Densidad de Defectos en Pruebas Sistémicas, Nivel de Inestabilidad de los Requerimientos, Nivel de Utilización de Integración Continua, Nivel de Experiencia, Entorno de Desarrollo, Porcentaje de Defectos en Revisiones Técnicas y Cobertura de Pruebas Unitarias.*

Rodger, Pankaj y Nahouraii (2011) estudian la vigencia de los factores propuestos en Subramanian y Zarnich (1996) y Foss (1993), así, demuestran que los factores más significativos que afectan la productividad son el *Tamaño del Equipo, Volatibilidad en Tipo de Desarrollo de Software (Nuevo, Mantenimiento), Plataforma de Desarrollo y el Tipo de Lenguaje de Programación.* Asimismo, se determina que las herramientas de Ingeniería de

Software Asistida por Computadora (CASE, por las siglas del inglés *Computer Aided Software Engineering*) no tienen impacto en la productividad de software.

Rodríguez, Sicilia, García y Harrison (2011) demuestran que existe una correlación estadística significativa entre *Tamaño del Equipo, Esfuerzo, Productividad y Duración del Proyecto*.

Yilmaz y O'Connor (2011) demuestran la correlación entre la productividad de software, la productividad social y capital social, así, se establece los siguientes factores: *Motivación, Proceso, Complejidad, Reutilización, Tamaño del Equipo, Liderazgo del Equipo, Resultados Colectivos, Conocimiento de la Información, Transparencia de Comunicación, Relaciones Sociales y Reuniones Frecuentes*.

Çetin y Alabaş-Uslu (2015) realizan un estudio en compañías de software de Turquía, y a través de análisis de regresión múltiple demuestran que existe una relación entre el *Tipo de Proyecto* y los factores *Severidad, Número Total de Defectos y Tamaño del Equipo del Proyecto*. También demuestran la relación entre *Duración del Proyecto* y los factores *Metodología, Número de Líneas Bases, Diferencia entre fechas actuales y planificadas (línea base), Duración de Análisis, Duración de Desarrollo, Duración de Pruebas, Número Total de Defectos, Número Total de Riesgos y Complejidad del Proyecto*.

Otros factores que impactan en la productividad son *Complejidad* (Zhan, Zhou & Zhao, 2012; Machek *et al.*, 2012; Yilmaz & O'Connor, 2011; Khan, *et al.*, 2014) y *Nivel de Experiencia* (Moreira *et al.*, 2010).

Por otro lado, existen trabajos que realizan un ranking de los factores, tal como se menciona en (Paiva, Barbosa, Lima & Alburqueque, 2010), donde demuestra que la *Motivación* y el *Compromiso* tienen una alta influencia positiva.

Wagner y Ruhe (2008a) realizan la clasificación de factores en ocho categorías, las cuales al final se agrupan en factores blandos y técnicos. Los autores definen a los factores blandos como “*todos los factores no técnicos que influyen en la productividad. Estos factores provienen principalmente del equipo y su entorno de trabajo*” (Wagner & Ruhe, 2008, p.12); y a los factores técnicos referidos a “*contiene todos los factores que tienen una relación directa con el producto, los aspectos técnicos del proceso y las herramientas que el desarrollador utiliza en el proyecto*” (Wagner & Ruhe, 2008, p. 12). Trendowicz (2007)

identifica varios contextos asociados a los factores, tales como Factores del Contexto en General, Factores específicos del Modelo, Factores específicos del Tipo de Desarrollo, Factores específicos del Dominio y Factores específicos de Reutilización, definiendo a los factores como “*factores que se incluyen en el modelo para explicar la variación de la productividad dentro de un contexto determinado*” (Trendowicz, 2007, p. 10). En la Tabla 3.6, se visualiza la clasificación de los factores propuestos por ambos autores.

Tal como se puede apreciar en la Tabla 3.6, la clasificación de Wagner y Ruhe (2008a) contempla categorías que en muchos casos no han sido cubiertos por Trendowicz (2007), tales como *Entorno de Desarrollo, Cultura Corporativa, Cultura del Equipo y Entorno*, sin embargo, estas categorías son necesarias, ya que agrupan a factores que son transversales a toda la organización y no son exclusivas del producto o proyecto. Es por ello que en el presente artículo se estará utilizando la categoría propuesta por Wagner y Ruhe (2008a) para la agrupación de factores.

En este contexto, se han recopilado de la literatura existente 74 factores que afectan la productividad, los cuales han sido clasificados en las siete categorías propuestas por Wagner y Ruhe (2008a): *Producto, Proceso, Entorno de Desarrollo, Cultura Corporativa, Cultura del Equipo, Capacidades y Experiencia, Proyecto*. En el Anexo A, se presentan los factores que corresponden a cada componente y categoría. La cantidad de factores y frecuencias por categorías está dado en la Tabla 3.7.

Tabla 3.6.

Clasificación de factores por categoría propuestos en Wagner y Ruhe (2008a); Trendowicz (2007)

<b>Categorías de Wagner &amp; Ruhe, 2008</b>		<b>Categorías de Trendowicz, 2007</b>	<b>Descripción</b>
<i>Técnicos</i>	Producto	Producto	<p>“...la categoría producto contiene todos los factores que tienen una relación directa con el producto...” (Wagner &amp; Ruhe, 2008, p. 12).</p> <p>“...características de los productos de software que se desarrollan a lo largo de todas las fases de desarrollo. Estos factores se refieren a productos tales como código de software, requisitos, documentación, etc., y sus características, como complejidad, tamaño, volatilidad, etc...” (Trendowicz, 2007, p. 10).</p>
	Proceso	Proceso	<p>“...la categoría proceso se compone de los aspectos técnicos del proceso...” (Wagner &amp; Ruhe, 2008, p.12).</p> <p>“...características de los procesos de software, así como métodos, herramientas y tecnologías aplicadas durante un proyecto de desarrollo de software. Incluyen, por ejemplo, la efectividad del aseguramiento de la calidad, la calidad de la prueba, la calidad del análisis y los métodos de documentación, la calidad y el uso de la herramienta, la calidad de la gestión del proceso o el grado de participación del cliente...” (Trendowicz, 2007, p. 10).</p>
	Entorno de Desarrollo		<p>“...la categoría entorno de desarrollo contiene factores acerca de las herramientas que el desarrollador usa en el proyecto...” (Wagner &amp; Ruhe, 2008, p. 12).</p>

<b>Categorías de Wagner &amp; Ruhe, 2008</b>		<b>Categorías de Trendowicz, 2007</b>	<b>Descripción</b>
<i>Blandos</i>	Cultura Corporativa		“...contiene los factores que están en un nivel más amplio para toda la empresa, mientras que...” (Wagner & Ruhe, 2008, p. 14).
	Cultura del Equipo		“...denota factores similares a nivel de equipo...” (Wagner & Ruhe, 2008, p. 14).
	Capacidades y Experiencia	Personal	“...en Capacidades y Experiencia son factores resumidos que está relacionados con las personas...” (Wagner & Ruhe, 2008, p. 14). “...características del personal involucrado en el Proyecto de desarrollo de software. Estos factores usualmente consideran la experiencia y capacidades de las partes interesadas del proyecto, como miembros del equipo de desarrollo (por ejemplo, analistas, diseñadores, programadores, gerentes de proyecto), así como usuarios de software, cliente, subcontratistas, etc...” (Trendowicz, 2007, p. 10).
	Entorno		“...representa las propiedades del entorno de trabajo...” (Wagner & Ruhe, 2008, p. 14).
	Proyecto	Proyecto	“...factores específicos del proyecto...” (Wagner & Ruhe, 2008, p. 14). “...considerar las diversas cualidades de la gestión y organización del proyecto, las limitaciones de desarrollo, las condiciones de trabajo o la rotación del personal...” (Trendowicz, 2007, p. 10).

Tabla 3.7.

Cantidad de factores por componente y categoría que impactan en la productividad

<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Número de Factor</b>	<b>Referencias</b>
<i>Programación</i>	Producto	17	Rodger <i>et al.</i> , 2011; Machek <i>et al.</i> , 2012; Yilmaz & O'Connor, 2011; Cheikhi <i>et al.</i> , 2012; Khan <i>et al.</i> , 2014; Çetin & Alabaş-Uslu, 2015; López <i>et al.</i> , 2010; Pai, <i>et al.</i> , 2015; Unluturk & Kurtel, 2015
	Proceso	13	Machek <i>et al.</i> , 2012; Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014; Cheikhi <i>et al.</i> , 2012; Çetin & Alabaş-Uslu, 2015
	Entorno de Desarrollo	1	Moreira <i>et al.</i> , 2010
	Cultura Corporativa	3	Premraj <i>et al.</i> , 2005; Khan <i>et al.</i> , 2014; Çetin & Alabaş-Uslu, 2015
	Cultura del Equipo	8	Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014
	Capacidades	6	Yilmaz & O'Connor, 2011; Moreira <i>et al.</i> , 2010; Khan <i>et al.</i> , 2014
	Experiencia		
	Entorno	0	
	Proyecto	12	Rodger <i>et al.</i> , 2011; Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014; Çetin & Alabaş-Uslu, 2015; Machek <i>et al.</i> , 2012; Cheikhi <i>et al.</i> , 2012
<i>Pruebas</i>	Producto	3	Moreira <i>et al.</i> , 2010; Pai <i>et al.</i> , 2015; Unluturk & Kurtel, 2015
	Proceso	2	Moreira <i>et al.</i> , 2010
<i>Análisis &amp; Diseño</i>	Producto	9	Cao, Ching & Thompson, 2012

### 3.7.3. Q3: ¿Qué modelos se han desarrollado?

Cardoso, Bert y Podestá (2010) afirman que un modelo es una representación simplificada de la realidad, definiéndose a través de los diferentes componentes y procesos que forman parte del sistema en estudio.

Se ha identificado diez modelos que miden la productividad en el desarrollo de software, los cuales están basados en diversos enfoques. En la Tabla 3.8, resumimos los modelos DEA, CMMI y Six Sigma, Lógica Difusa, Reutilización del Software, SEM, Factor Total de Productividad, IEEE Std. 1045 & ISO 9216-4.

Por otro lado, también existen trabajos que presentan modelos de evaluación de productividad de software, pero que no han sido seleccionados, por no cumplir los criterios del presente estudio. Entre ellos se encuentran los modelos descritos en Sudhakar, Farooq y Patnaik (2012), como el Modelo Simple de Productividad descrito en Card (2006); Tasa de Entrega de Productividad Normalizada (PDR, por las siglas del inglés *Normalized Productivity Delivery Rate*) de Jiang, Naud'é y Comstock (2007) y SEM de Khan (2014), los cuales se presentan en la Tabla 3.9.

Tabla 3.8.

Modelos de medición de la productividad de software

Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
M1	Análisis Envoltorio de Datos (DEA)	$Productividad = e^{1.95} FP^{0.7} (*)$ Donde: <ul style="list-style-type: none"> <li>• Productividad: La productividad es el Esfuerzo del Trabajo</li> <li>• FP: Puntos de Función</li> </ul>	Grandes proyectos en bancos de Canadá.	Los autores miden la productividad considerando como salida a los puntos de función (FP) y la entrada, el esfuerzo de desarrollo, utilizando DEA como técnica de medición.	Asmild <i>et al.</i> , 2006
M2	CMMI & Six Sigma	$Productividad = 32.087 - 3.637 DDST + 11.71$ $LRU - 9.451 LCIU - 0.8187 LEX DENV$ Donde: <ul style="list-style-type: none"> <li>• DDST: Densidad de Defectos en Pruebas Sistémicas <math>DDST = 1.8955 - 0.5087 PDTR - 1.6020 UTC</math></li> <li>• LRU: Nivel de Inestabilidad de los Requerimientos</li> </ul>	5 proyectos del Atlantic Institute, una empresa mediana, evaluada en CMMI nivel 3, cuyo objetivo real	Los autores proponen un modelo de productividad basado en CMMI y Six Sigma. En el artículo, ellos definen la productividad como la Productividad General de los Proyectos (GPP),	Moreira <i>et al.</i> , 2010



Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
M3	Lógica Difusa	<ul style="list-style-type: none"> <li>• LCIU: Nivel de Utilización de Integración Continua</li> <li>• LEX: Nivel de Experiencia</li> <li>• DENV: Entorno de Desarrollo</li> <li>• PDTR: Porcentaje de Defectos en Revisiones Técnicas</li> <li>• UTC: Cobertura de Pruebas Unitarias</li> </ul> <p><b>Productividad</b> = 10.3 + 0.31701* N&amp;C</p> <p>N&amp;C: Código Nuevo y modificado. Se considera como líneas físicas de código (LOC). N&amp;C es compuesta de código agregado y modificado.</p>	<p>es alcanzar el nivel de madurez 5</p> <p>Conjunto de datos de 140 pequeños programas desarrollados con prácticas basadas en el Proceso de Software Personal (PSP)</p>	<p>usando la técnica de regresión lineal para establecer su modelo.</p> <p>Los autores proponen que un modelo de lógica difusa se podría usar para estimar y predecir la productividad del desarrollo de software.</p>	López <i>et al.</i> , 2010
M4	Modelo de Ecuaciones	<p><b>Productividad</b> = <math>f</math> (Motivación, Proceso, Complejidad, Reutilización, Tamaño del Equipo, Productividad Social, Capital Social)</p>	<p>El análisis se realizó con</p> <p>227</p>	<p>Los autores proponen un modelo empíricamente validado para medir el</p>	Yilmaz & O'Connor, 2011

Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
Estructurales (SEM)	<ul style="list-style-type: none"> <li>Productividad Social = <math>f</math>(Liderazgo del Equipo, Resultados Colectivos, Conocimiento de la Información)</li> <li>Capital Social = <math>f</math> (Transparencia de la Comunicación, Relaciones Sociales, Reuniones Frecuentes)</li> </ul>	participantes. Alrededor del 65% de los participantes eran postgraduados.	capital social, la productividad social y la productividad del desarrollo de software.	Machek <i>et al.</i> , 2012	
M5	Factor Total de Productividad	<ul style="list-style-type: none"> <li>Definición de Salidas (<math>y</math>)</li> <li>Definición de Entradas (<math>x</math>)</li> <li>Determinación de Precios de Entradas y Salidas (<math>p, w</math>)</li> </ul> <p>Cálculo del Factor Total de Productividad (Índices de Productividad de Fisher o Törnqvist)</p>	No se explica en el artículo.	Modelo de medición de la productividad total del proceso de software basado en el Factor Total de Productividad.	Machek <i>et al.</i> , 2012
M6	ISO 9126-4 IEEE Std. 1045	<p>Modelo de productividad consolidado</p> <p><b>Productividad</b>=<math>f</math> (Eficiencia, Completitud, Precisión)</p> <p>Donde:</p> <ul style="list-style-type: none"> <li>Eficiencia=<math>f</math> (Tiempo de Tareas, Costo de Tarea, Frecuencia de Error, Eficiencia del usuario, Eficiencia del usuario en comparación con un experto, Eficiencia de la tarea (tiempo),</li> </ul>	No se explica en el artículo.	Los autores revisan los estándares internacional IEEE Std. 1045 e ISO 9216-4, definiendo cómo ambos estándares pueden usarse juntos para la medición de la productividad.	Cheikhi <i>et al.</i> , 2012

Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
M7	DEA con rendimientos variables de escala (DEA VRS, por las siglas del inglés Data Envelopment	<p data-bbox="583 207 1188 298">Eficiencia de la tarea (costo), Proporción del tiempo que el usuario es productivo)</p> <ul data-bbox="541 321 1188 899" style="list-style-type: none"> <li data-bbox="541 321 1188 516">• Completitud= f (Porcentaje de tareas realizadas, Porcentaje de usuarios que tuvieron éxito en completar la tarea, Logro del objetivo de porcentaje para cada unidad de tiempo)</li> <li data-bbox="541 539 1188 899">• Precisión=f (Porcentaje de tareas logradas correctamente, Porcentaje de usuarios que logran la tarea correctamente. ¿Con qué frecuencia encuentra el usuario resultados inexactos al realizar la tarea? ¿Con qué frecuencia realiza la tarea el usuario con una precisión inadecuada?)</li> </ul> <p data-bbox="541 922 1188 1062"><b>Productividad</b>= f(n (O<sub>T</sub>), n (R<sub>T</sub>), n (P<sub>T</sub>), P<sub>O</sub> (M<sub>T</sub>, o), P<sub>XO</sub> (M<sub>T</sub>), P<sub>R</sub> (M<sub>T</sub>, e), P<sub>XR</sub> (M<sub>T</sub>), R<sub>O</sub> (M<sub>T</sub>, o), R<sub>O</sub>(M<sub>T</sub>)/ (hora de trabajo variable)</p> <ul data-bbox="541 1084 1188 1341" style="list-style-type: none"> <li data-bbox="541 1084 1188 1127">• n (O<sub>T</sub>): Recuento de tipos de objeto por técnica</li> <li data-bbox="541 1149 1188 1230">• n (R<sub>T</sub>): Recuento de tipos de relación por técnica</li> <li data-bbox="541 1253 1188 1341">• n (P<sub>T</sub>): Recuento de tipos de propiedad por técnica</li> </ul>	Información sobre 25 proyectos SAD basados en UML. Esta información se obtuvo de	Un enfoque para evaluar la eficiencia relativa de los proyectos de software utilizando estas medidas de complejidad como salidas.	Cao <i>et al.</i> , 2012

Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
Analysis Variable Returns to Scale)	<ul style="list-style-type: none"> <li>• <math>P_O (M_T, o)</math>: Recuento de número de propiedades para un tipo de objeto dado</li> <li>• <math>P_{XO} (M_T)</math>: Promedio de número de propiedades para un tipo de objeto dado</li> <li>• <math>P_R (M_T, e)</math>: Número de propiedades de un tipo de relación y los tipos de roles que lo acompañan</li> <li>• <math>P_{XR} (M_T)</math>: Promedio de propiedades por tipo de relación</li> <li>• <math>R_O (M_T, o)</math>: Cantidad de tipos de relaciones que se pueden conectar a un determinado tipo de objeto</li> <li>• <math>R_O(M_T)</math>: Número promedio de tipos de relaciones que se pueden conectar a un determinado tipo de objeto</li> </ul>	diferentes empresas	El modelo se aplica en el proceso de análisis y diseño de sistemas (SAD, por las siglas del inglés Systems Analysis and Design).		
M8	Reutilización del Software	$\text{Productividad} = \sum_{i=1}^n (r_i + f_i + l_i + c_i) / \sum I$ <p>Donde:  <math>r_i</math> = Reutilización  <math>f_i</math> = Funcionalidad  <math>l_i</math> = Longitud</p>	En Nigeria, por instancia, con referencias especiales a	Este modelo considera factores como reutilización, complejidad, longitud, funcionalidad y esfuerzo.	Nwelih & Amadin, 2008

Id	Enfoque	Modelo	Contexto	Consideraciones	Referencias
M9	DEA VRS	$c_i = \text{Complejidad}$ $\sum I = \text{Esfuerzo}$ <b>Productividad</b> = FP, Calidad/ Esfuerzo Total (Esfuerzo de Desarrollo + EoC + EoNC) <ul style="list-style-type: none"> <li>• Esfuerzo de Conformidad (EoC) = (Evaluación + Prevención) Costos. Ejemplo: esfuerzo de revisión</li> <li>• Esfuerzo de No Conformidad (EoNC) (Interna + Externa) Costos de Fallas. Ejemplo: El esfuerzo de retrabajo, incluido el esfuerzo de falla</li> <li>• Calidad = <math>f(\text{Defectos})</math></li> </ul>	la industria de TI El modelo se aplica a los datos recogidos en 79 proyectos de desarrollo de software de una organización CMMI nivel 5.	Los autores desarrollan un modelo para medir la productividad de los proyectos de software e identificar las mejores prácticas. Este estudio muestra que incluir EoC y EoNC como entradas tiene un impacto positivo en la frontera de mejores prácticas.	Pai <i>et al.</i> , 2015
M10	Técnica Estadística	<b>Productividad</b> = $(\sum_{i=1}^n (\text{Calidad}_i * \text{Cantidad}_i) / \text{Horas Netas de Tareas}_i) * \text{Métodos Ponderados}$ <ul style="list-style-type: none"> <li>• Calidad = <math>1 - ((10 * \text{total \# de Defectos Serios} + 3 * \text{total \# de Defectos Medianos} + \text{total \# de Defectos Triviales}) / \text{LOC})</math></li> <li>• Cantidad = LOC</li> </ul>	Durante esta fase, tres estudiantes de nivel senior trabajaron en el proceso de	Un método para medir la productividad de los programadores de software de manera individual. Este método proporciona una opinión	Unluturk & Kurtel, 2015

<b>Id</b>	<b>Enfoque</b>	<b>Modelo</b>	<b>Contexto</b>	<b>Consideraciones</b>	<b>Referencias</b>
		<ul style="list-style-type: none"> <li>Métodos Ponderados = <math>(3 * (\text{total\# de constructores} + \text{total\# de destructores}) + 5 * \text{total \# de selectores} + 9 * \text{total \# de iteradores} + 15 * \text{total \# de modificadores}) * (1/N)</math></li> </ul> <p><i>donde N es el número total de métodos para un programador individual</i></p>	medición de dos proyectos.	común para comprender, controlar y mejorar las prácticas de ingeniería de software.	

(\*) Para una mejor comprensión del modelo se ha simplificado la fórmula.

Tabla 3.9.

Otros modelos de medición de la productividad de software, no seleccionados (Sudhakar et al., 2012)

<b>ID</b>	<b>Técnica/Modelo</b>	<b>Fórmula/Descripción</b>	<b>Consideraciones</b>	<b>Referencias</b>
<i>I</i>	Modelo Simple de Productividad	<ul style="list-style-type: none"> <li>Productividad Física = Número de LOC / horas hombre o días o meses</li> <li>Productividad Funcional = Número de Puntos de Función / horas hombre o días o meses</li> <li>Productividad Económica = Valor / Costo</li> </ul> <p>Donde Valor = f(Precio, Tiempo, Calidad, Funcionalidad)</p>	Este modelo considera las entidades tales como Producto, Proceso o Subproceso, Requisitos, Valor, Costo y Esfuerzo.	Card, 2006

ID	Técnica/Modelo	Fórmula/Descripción	Consideraciones	Referencias
2	Tasa de Entrega de Productividad Normalizada	$\log(\text{PDR}) = 2.8400 + 0.3659 \times \log(\text{Team Size}) - 0.6872 \times I(3\text{GL}) - 1.2962 \times I(4\text{GL}) - 1.3225 \times I(\text{ApG}) - 0.1627 \times I(\text{MR}) - 0.4189 \times I(\text{Multi}) - 0.3201 \times I(\text{PC}) - 0.4280 \times I(\text{OO}) - 0.2812 \times I(\text{Event}) + 0.7513 \times I(\text{OO:Event}) - 0.2588 \times I(\text{Business}) - 0.0805 \times I(\text{Regression}) + 1.0506 \times I(\text{Business:Regression})$ <p>Normalizada PDR = (Esfuerzo de Trabajo Normalizado) / (Puntos de Función Ajustados)</p>	Utiliza dos variables continuas: Tamaño promedio del equipo y PDR; y seis variables categóricas como Tipo de lenguaje, Tipo de desarrollo, Plataforma de desarrollo, Técnicas de desarrollo, Herramienta CASE utilizada y metodología.	Jiang <i>et al.</i> , 2007
3	Modelo de Ecuaciones Estructuradas	<p><b>Productividad</b> = f (Tecnología, Cultura de Trabajo, Interés en Trabajo Individual, Complejidad, Tamaño del Equipo, Productividad Humana)</p> <p>Productividad Humana = f (Habilidades del gerente, unidad de equipo, vida social, frecuencia de reunión)</p>	Los autores desarrollan un modelo de productividad empírico, así, muestran la correlación entre productividad de software y productividad humana.	Khan <i>et al.</i> , 2014

Para la evaluación de los modelos presentados, se ha considerado los siguientes criterios:

- **Formulación de la productividad.** Explicita una función para medir la productividad.
- **Cobertura del modelo.** Especifica si el modelo cubre parte o todas las unidades de trabajo de la fábrica de software (Análisis & Diseño, Programación, y Pruebas de Software).
- **Sustento teórico.** Especifica si el modelo presenta sustento o fundamento en teorías.
- **Validación.** Especifica si el modelo evaluado ha sido probado en proyectos reales de una organización o por repositorios de datos.
- **Pre-procesamiento.** El modelo describe un procedimiento para la captura y pre-procesamiento de los datos para medir la productividad.

Una vez que se han definido los criterios, en la Tabla 3.10 se puede visualizar los criterios de los modelos de productividad.

Tabla 3.10.

Evaluación de Modelos de Productividad de Software

Criterio	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
<i>Formulación</i>	√	√	√	No	No	No	No	Si	No	√
<i>de</i>										
<i>Productividad</i>										
<i>Cobertura del Modelo</i>	PRO	PRO - TES	PRO	PRO	PRO	PRO	AD	PRO	PRO - TES	PRO - TES
<i>Sustento Teórico</i>	√	√	√	√	√	√	√	√	√	√
<i>Validación</i>	NE	√	√	NE	NE	NE	√	NE	√	√
<i>Pre-Procesamiento</i>	NE	NE	NE	NE	NE	NE	NE	NE	√	NE

√ Sí cumple el criterio establecido NE: No Especifica PRO: Programación TES: Pruebas AD: Análisis & Diseño

En la Tabla 3.11, se agrupan los modelos revisados en cada uno de los componentes que conforman la fábrica de software.



Tabla 3.11.

Modelos de Medición de la Productividad de Software por Componente

<b>Unidad de Trabajo</b>	<b>Modelo</b>	<b>Enfoque</b>	<b>Referencias</b>
<i>Programación</i>	M1	Análisis Envoltorio de Datos (DEA)	Asmild <i>et al.</i> , 2006
	M2	CMMI & Six Sigma	Moreira <i>et al.</i> , 2010
	M3	Lógica Difusa	López <i>et al.</i> , 2010
	M4	Modelo de Ecuaciones Estructuradas	Yilmaz & O'Connor, 2011
	M5	Factor Total de Productividad	Machek <i>et al.</i> , 2012
	M6	ISO 9126-4 IEEE Std.1045	Cheikhi <i>et al.</i> , 2012
	M8	Reutilización de Software	Nwelih & Amadin, 2008
	M9	DEA con rendimientos variables de escala (DEA VRS)	Pai <i>et al.</i> , 2015
	M10	Técnicas Estadísticas	Unluturk & Kurtel, 2015
	<i>Pruebas</i>	M2	CMMI & Six Sigma
M9		DEA con rendimientos variables de escala (DEA VRS)	Pai <i>et al.</i> , 2015
M10		Técnicas Estadísticas	Unluturk & Kurtel, 2015
<i>Análisis &amp; Diseño</i>	M7	DEA con rendimientos variables de escala (DEA VRS)	Cao <i>et al.</i> , 2012

3.7.4. Q4: ¿Qué factores incluyen los modelos?

Tomando en consideración la información encontrada en la literatura, en la Tabla 3.12 se visualiza los factores incluidos en los distintos modelos, clasificándolos por Componente.

Tabla 3.12.

Factores incluidos en los Modelos de Productividad de Software por Componente

Unidad de Trabajo	Categoría	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
<i>Programación</i>	Producto			F10, F15, F16	F3, F4, F17	F1, F2, F3, F4, F5, F6, F7, F8, F17	F9, F10		F4, F5, F6, F17	F10	F9
	Proceso	F35, F37, F38, F39			F31, F32	F30, F34	F33				
	Entorno de Desarrollo Cultura Corporativa		F45								

Unidad de Trabajo	Categoría	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
	Cultura de Equipo				F49, F50, F51, F52, F53, F54						
	Capacidades y Experiencia		F58		F57, F59, F60, F61						
	Proyecto						F64, F65, F66, F67, F68, F69, F70, F71, F73	F66, F72			

<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>M1</b>	<b>M2</b>	<b>M3</b>	<b>M4</b>	<b>M5</b>	<b>M6</b>	<b>M7</b>	<b>M8</b>	<b>M9</b>	<b>M10</b>
<i>Pruebas</i>	Producto		F18, F19, F20							F18, F19	F18, F19
	Proceso		F43, F44								
<i>Análisis &amp; Diseño</i>	Producto							F21, F22, F23, F24, F25, F26, F27, F28, F29			

### 3.8. Discusión

De acuerdo a la literatura, existen diversas publicaciones sobre productividad de software que se han venido revisando en los últimos años, el cual puede ser justificado por el interés que los investigadores han dado al tema de la productividad en la industria del software, sin embargo, es muy escasa o casi nula la literatura acerca de la medición de la productividad en fábricas de software, cuya utilización del modelo ha traído grandes ventajas, tal como lo demuestra en su estudio (Clements & Northrop, 2001). Si bien es cierto, existen modelos de medición de la productividad del software (Asmild *et al.*, 2006; Nwelih & Amadin, 2008; Moreira *et al.*, 2010; Yilmaz & O'Connor, 2011; Machek *et al.*, 2012; Cheikhi *et al.*, 2012), la mayoría ha considerado solo la unidad de trabajo de programación, lo cual conlleva a un sesgo en la medición, que trae como consecuencia que no se cuente con indicadores de medición que faciliten la toma de decisiones en la organización.

Los estudios han sido orientados en dos vertientes principales: técnicas o modelos que miden la productividad y los factores que la afectan. Estos estudios representan al proceso de desarrollo como una unidad, dificultándose la medición si se quisiera realizar de manera independiente por cada uno de los componentes que participan en el ciclo de vida de desarrollo del software, no habiéndose encontrado evidencias de modelos que midan la productividad en todas las fases del proceso de desarrollo de software.

Asimismo, no existe un consenso de los autores sobre los factores que generan mayor impacto en la productividad y la unidad de medida a tenerse en cuenta para la medición de la misma.

Del total de documentos, más del 70% de los estudios están enfocados en los factores que afectan la productividad, de lo cual se puede inferir la importancia que han dado los investigadores al aspecto Factor, así, se busca explicar el impacto que tiene este aspecto en la medición. Existen pocos estudios (27,41%) enfocados a determinar modelos que midan la productividad en las diferentes unidades de la fábrica de software.

A través de este capítulo, se ha buscado dar respuesta a las siguientes interrogantes de investigación:

### 3.8.1. *¿Qué es la productividad de fábricas de software?*

Los estudios explican sobre la productividad de software, y estos a su vez se enfocan en la unidad de trabajo de Programación, sin embargo, es casi nula la información acerca de la productividad en la fábrica de software como un todo.

En los estudios se habla muy poco de las fábricas de software, a pesar de ser un modelo que se viene trabajando en la industria hace muchos años, y, por ende, no se tiene una definición clara de la productividad.

Si la medición de la productividad es muy importante para el desarrollo de una organización, esta se hace compleja si no se tiene una definición clara. En este contexto, es necesario definir la productividad de las fábricas de software como “*un indicador de la eficiencia de los recursos utilizados en diferentes unidades de trabajo de la fábrica de software para el logro del producto final*”.

Mediante este concepto, la productividad exige una buena gestión de los recursos asignados a fin de conseguir resultados que vuelvan eficientes todas las labores desarrolladas dentro de la organización, no solo en lo que respecta a la producción del software, sino también en lo referente a los métodos utilizados y a la relación interna de la organización.

### 3.8.2. *¿Qué factores influyen en la medición de la productividad y cómo se clasifican?*

Se han identificado 74 factores, los cuales han sido agrupados de acuerdo a las categorías propuestas por Wagner y Ruhe (2008a), ya que contempla categorías que en muchos casos no han sido cubiertos por Trendowicz (2009), tales como *Entorno de Desarrollo, Cultura Corporativa, Cultura del Equipo y Entorno*, sin embargo, estas categorías son necesarias ya que agrupan a factores que son transversales a toda la organización y no son exclusivas del producto o proyecto.

### 3.8.3. *¿Qué modelos se han desarrollado?*

Se ha identificado diez modelos de medición de la productividad de software que están basados en DEA (Asmild *et al.*, 2006; Cao *et al.*, 2012; Pai, *et al.*, 2015), CMMI & Six Sigma (Moreira, *et al.*, 2010), Lógica Difusa (López, *et al.*, 2010), Modelo de Ecuaciones Estructuradas (Yilmaz & O'Connor, 2011), Reutilización de Software (Nwelih & Amadin,

*et al.*, 2008), Factor Total de Productividad (Machek *et al.*, 2012), ISO 9126-4 IEEE Std.1045 (Cheikhi *et al.*, 2012); y otras técnicas estadísticas planteadas por Unluturk y Kurtel (2015). Si bien es cierto que se definen como modelos de medición de la productividad, muchos de ellos plantean una función, sin embargo, no derivan a una fórmula (Yilmaz & O'Connor, 2011; Machek *et al.*, 2012; Cheikhi *et al.*, 2012), solo mencionando diversos factores a tener en cuenta en la medición.

Los modelos solo han sido diseñados para realizar una medición de la productividad interna, pero a pesar de ello son incompletos, ya que han sido orientados en su mayoría para la unidad de trabajo de programación (Asmild *et al.*, 2006; Nwelih & Amadin, 2008; Moreira *et al.*, 2010; Yilmaz & O'Connor, 2011; Machek *et al.*, 2012; Cheikhi *et al.*, 2012), y muy pocos para pruebas y análisis & diseño (Moreira *et al.*, 2010; Pai *et al.*, 2015; Unluturk & Kurtel, 2015; Cao *et al.*, 2012), lo cual genera un sesgo en la medición, ya que no permitiría establecer la productividad en los otros componentes de la fábrica de software. No se ha encontrado modelos que midan la productividad en toda la fábrica de software.

Los modelos no miden la productividad externa, es decir, no realizan una evaluación comparativa de la productividad, lo cual podría traer consecuencias negativas para la organización, ya que si solo se compara de manera interna, tendría una información sesgada de cómo se está comportando el mercado.

Algunos de los modelos como Asmild *et al.* (2006) y Moreira *et al.* (2010) han sido validados en un entorno real, sin embargo, la información utilizada para el diseño del modelo ha sido obtenida de la misma organización, lo cual dificulta su generalización si se quisiera comparar con otras organizaciones.

También, los modelos plantean un modelo cualitativo (Yilmaz & O'Connor, 2011; Machek *et al.*, 2012; Cheikhi *et al.*, 2012), que si bien se podría considerar la precisión del modelo, esto no podría generalizarse debido a que los modelos no presentan una aplicación práctica o no han sido aplicados en casos de estudio.

#### 3.8.4. *¿Qué factores incluyen los modelos?*

Asimismo, los factores que más se han utilizado en los modelos son los siguientes: *Tamaño de Equipo*, *Reutilización* y *Complejidad*, sin embargo, existen muchos factores que no han sido tomados en cuenta en los modelos, tales como *Números totales de lanzamientos*,

*Problemas, Riesgos, Solicitud de cambio, Ocurrencia de espera, Duración de espera, Motivo de espera.* Si tomamos en cuenta los factores que existen en la literatura y los modelos que se han elaborado a la fecha, se puede notar que no existe una homogeneización de los factores a tomar en cuenta en un modelo, y, en muchos casos, no se ha tomado en cuenta algunos de los factores, afectando por ello la precisión del modelo, ya que se considera que cuantos más factores cubra el modelo, se podrá establecer más aspectos de la realidad.

### **3.9. Conclusión**

Si una organización necesita conocer la eficiencia y eficacia con la que se vienen desarrollando las actividades claves en el proceso de producción, es necesario que tengan claro la forma de cómo medir la productividad, la cual es una actividad compleja, porque no hay un consenso sobre los aspectos a tener en cuenta para su medición, ya que incorpora diversos factores que en muchos casos no son tomados en cuenta, siendo los más resaltantes en los modelos encontrados los siguientes: *Tamaño del Equipo, Reutilización, Complejidad, Cohesión del Equipo, Unidad del Equipo, Relaciones Sociales, Vida Social, Reuniones Regulares, Frecuencia de Reuniones*; sin embargo, existen muchos factores que no han sido considerados, lo cual afecta la precisión de los modelos.

No existen modelos para la medición de la productividad en las fábricas de software como un todo, habiéndose encontrado estudios en su mayoría orientados a la unidad de trabajo de Programación y muy pocos a la de Análisis & Diseño y Pruebas.

Estos estudios representan al proceso de desarrollo como una unidad, dificultándose la medición si se quisiera realizar de manera independiente por cada uno de los componentes que participan en el ciclo de vida de desarrollo del software, no habiéndose encontrado evidencias de modelos que midan la productividad en todas las fases del proceso de desarrollo de software.

Los modelos no realizan una medición de la productividad externa, pues solo se han orientado a medir la producción interna de la organización, lo cual conllevaría a tomar malas decisiones, ya que se podría considerar una alta productividad en la interna, sin embargo, podríamos estar debajo del estándar, lo cual nos vuelve ineficientes.



No existe un estándar de la unidad de medida a utilizar en la medición de la productividad, ya que para los factores de entrada y salida se utilizan diversos criterios, lo cual dificulta la medición comparativa entre proyectos y organizaciones.

## Capítulo 4: Nuevos Factores que Influyen en la Productividad de las Fábricas de Software

El presente capítulo identifica tres nuevos factores que influyen en la productividad de la fábrica de software: Uso de Reglas y Eventos, Heterogeneidad del Equipo y Comunicación Osmótica. Para este fin, se han revisado teorías sobre la Perspectiva de la Acción del Lenguaje y la Teoría de la Memoria Transactiva, así como metodologías para sustentar estos factores; y para validarla, se hace un estudio empírico en fábricas de software de Perú.

### 4.1. Introducción

Cusumano (1989) define Software Factory como *“una empresa que tiene características de producción de software en gran escala, estandarización de tareas, estandarización del control, división del trabajo, mecanización, automatización y aplicación sistemática de buenas prácticas de Ingeniería de Software”*. La fábrica de software trae grandes ventajas, tales como la disminución en costos de producción por producto hasta en un 60%; el ahorro del tiempo de puesta en el mercado hasta en un 98%; la reducción del requerimiento en mano de obra hasta en un 60%; el mejoramiento de la productividad en alrededor de 10 veces y la calidad de cada producto derivado en hasta 10 veces menos errores, aumentando así el portafolio de productos y servicios ofrecidos, y la posibilidad de ganar nuevos mercados (Clements & Northrop, 2001).

Al medir la productividad, una fábrica de software cuenta con indicadores que le permiten compararse en el mercado, de tal manera que le ayuda a plantearse acciones que incrementen su eficiencia global, lo que le permitirá utilizar todos los recursos de una manera eficaz y eficiente para obtener los mejores resultados posibles. En otras palabras, se necesita saber cómo la organización se está desempeñando, en relación a períodos anteriores, y a sus competidores, así, surgen las siguientes preguntas: ¿Se está incrementando, disminuyendo, avanzando o se está retrocediendo? ¿Cuál es la magnitud de ese avance o de ese retroceso? ¿Son eficaces las estrategias implementadas?

Todos los modelos que miden la productividad en una fábrica de software toman en cuenta diversos elementos, tales como procesos, recursos, unidad de medida, entre otros, pero no

consideran que afecta el recurso y cuál es su repercusión en la medición de la productividad, por ejemplo, la motivación y la confianza en el equipo son factores que podrían tener un impacto positivo en la productividad, mientras que en un equipo desmotivado podría tener un efecto contrario (Yilmaz & O'Connor, 2011). Esta es la razón por la cual se vienen desarrollando estudios para identificar los factores que afectan la productividad, sin embargo, se evidencia que: (i) los factores identificados se orientan a las unidades de trabajo de Programación, lo que no se podría generalizar a la Fábrica de Software, dado que esta contempla otras unidades de trabajo y cada unidad tiene sus particularidades; (ii) existen factores que influyen en la productividad en otros dominios de conocimiento, pero no han sido analizados en la fábrica de software.

Además, no han considerado que la productividad depende de diversos factores que va más allá del input y output, los mismos que influyen en el proceso, contexto, entre otros (Arcudia-Abad, *et al.*, 2007; Nomura, *et al.*, 2006).

En este capítulo introducimos nuevos factores que afectan la productividad en las fábricas de software, las cuales están sustentadas en las teorías Perspectiva de la Acción del Lenguaje (LAP, por las siglas del inglés *Language Action Perspective*), Teoría de la Memoria Transactiva (TMT, por las siglas del inglés *Transactive Memory Theory*), y las buenas prácticas del agilismo como el Time-boxing. Además, estudiamos cómo afectan dichos factores en las unidades de trabajo como son a) Análisis & Diseño, b) Programación, y c) Pruebas. Para la validación de la propuesta, se realizaron 150 encuestas. El presente trabajo es parte de una investigación sobre modelos de productividad para fábricas de software.

El capítulo está organizado en 6 secciones. La sección 2 detalla el modelo conceptual propuesto y los elementos que la componen. La sección 3 describe la metodología de investigación e incluye la estrategia aplicada para la obtención de la información y análisis de resultados. La sección 4 presenta los resultados del estudio. En la sección 5 se establece una discusión sobre los hallazgos encontrados en la validación. Finalmente, nuestras conclusiones son presentadas en la sección 6.

## **4.2. Modelo Conceptual**

A continuación, se muestran las unidades de trabajo y los factores que la afectan.

### **4.2.1. Unidades de Trabajo**

De la Tabla 4.1, se puede identificar tres grandes componentes en la organización de la fábrica de software: Gestión del Proyecto, Producción de Software y Soporte. A su vez, cada componente puede estar dividido en unidades de trabajo, tal como se muestra en la Tabla 4.2. Estas fueron obtenidas del análisis y emparejamiento realizado de los trabajos realizados sobre estructura de la fábrica de software, ya que en muchos casos los autores traslapan las unidades o en su defecto le han dado otra denominación.

Tabla 4.1.

Componentes de la fábrica de software

<b>Componente</b>	<b>Descripción</b>	<b>Referencias</b>
<i>Gestión de Proyecto</i>	Responsable de la relación con el cliente, negociación, planificación, elaboración y gerenciamiento de contratos de acuerdo con el planeamiento estratégico de la empresa, así como de la gerencia, elaboración, programación y distribución de las órdenes de servicio y asignación de los recursos necesarios para la ejecución del servicio, estructurándose en las siguientes unidades de trabajo: Servicio, Gestión de Proyectos y Planificación y Control de la Producción.	Li <i>et al.</i> , 2001; Kruchten, 2004; Yanosky, 2005; Nomura <i>et al.</i> , 2006
<i>Producción de Software</i>	Responsable de actividades propias del proceso de desarrollo. En esta componente se incluyen las unidades de trabajo de: Modelado de Negocio, Arquitectura de Solución, Proyecto Conceptual, Requisitos, Análisis y Diseño, Implementación y Pruebas.	Basili <i>et al.</i> , 1992; Li <i>et al.</i> , 2001; Rockwell & Gera, 1993; Kruchten, 2004; Fernandes & Teixeira, 2004; Yanosky, 2005; Nomura <i>et al.</i> , 2006; Trujillo, 2007; Fabri <i>et al.</i> , 2004; Fabri <i>et al.</i> , 2004a; Fabri <i>et al.</i> , 2005; Fabri <i>et al.</i> , 2007; Fabri <i>et al.</i> , 2007a; Fabri <i>et al.</i> , 2007b; Fabri <i>et al.</i> , 2007c; Pessoa <i>et al.</i> , 2004

Componente	Descripción	Referencias
<i>Soporte</i>	Proporciona apoyo para la integración y desarrollo de actividades del proceso de desarrollo, incluyéndose entre otros: gestión de procesos, infraestructura y soporte, diseño de interfaces, seguridad, documentación, entre otros.	Basili <i>et al.</i> , 1992; Li <i>et al.</i> , 2001; Kruchten, 2004; Yanosky, 2005; Nomura <i>et al.</i> , 2006; Fabri <i>et al.</i> , 2004; Fabri <i>et al.</i> , 2004a; Fabri <i>et al.</i> , 2005; Fabri <i>et al.</i> , 2007; Fabri <i>et al.</i> , 2007a; Fabri <i>et al.</i> , 2007b; Fabri <i>et al.</i> , 2007c; Pessoa <i>et al.</i> , 2004

Tabla 4.2.

Unidades de trabajo de los componentes de la Fábrica de Software

<b>Componente</b>	<b>Unidad de Trabajo</b>	<b>Descripción</b>	<b>Referencias</b>
<i>Gestión del Proyecto</i>	Servicio	Definición del tipo de servicio que se brindará al cliente.	Nomura <i>et al.</i> , 2006
	Gestores de Proyecto	Asignación de la capa de gestión que estará a cargo de la ejecución del proyecto.	Nomura <i>et al.</i> , 2006
	Planificación y Control de la Producción	Definición del equipo que atenderá los requerimientos, así como la información necesaria para el Plan de Producción.	Nomura <i>et al.</i> , 2006
<i>Producción de Software</i>	Modelo de Negocio	Definición y elaboración del modelado de negocio del producto a desarrollar.	Li <i>et al.</i> , 2001; Kruchten, 2004; Yanosky, 2005; Nomura <i>et al.</i> , 2006
	Análisis & Diseño	Definición de la arquitectura base y la especificación funcional del requerimiento.	Kruchten, 2004, Fernandes & Teixeira, 2004; Nomura <i>et al.</i> , 2006
	Programación	Implementación de la solución requerida por el cliente.	Kruchten, 2004, Fernandes & Teixeira, 2004; Nomura <i>et al.</i> , 2006

Componente	Unidad de Trabajo	Descripción	Referencias
<i>Soporte</i>	Pruebas	Definición y ejecución de casos de prueba, que son corroborados con los criterios de aceptación definidos por el cliente.	Kruchten, 2004, Fernandes & Teixeira, 2004; Nomura <i>et al.</i> , 2006
	Procesos	Establecimiento de los procesos claves para la atención de los requerimientos de acuerdo con el tipo de servicio a brindar al cliente.	Li <i>et al.</i> , 2001
	Gestión de Indicadores	Definición y medición de los KPI que se utilizarán en el proyecto.	CMMI, 2016
	Gestión del Conocimiento	Políticas y activos organizacionales.	CMMI, 2016
	Arquitectura	Soporte de activos de programación o componentes de software.	Basili, <i>et al.</i> , 1992; Li, <i>et al.</i> , 2001; Yanosky, 2005
	Metodologías, Técnicas y Herramientas	Teniendo como referencia el proceso definido, se definen las metodologías, técnicas y herramientas que darán soporte al proyecto.	Li <i>et al.</i> , 2001; Kruchten, 2004



La Figura 4.1 ilustra el funcionamiento de una fábrica de software, en la cual se puede apreciar que los clientes solicitan diferentes tipos de requerimientos, los cuales son atendidos por la componente de Gestión del Proyecto (PM, por las siglas del inglés *Project Management*), en la que, tomando en consideración la capacidad de la fábrica, se acuerda los requerimientos a ser atendidos, se estima y planifica la atención en períodos acordados con el cliente, generándose el Plan de Producción. Luego, los requerimientos aceptados en PM son trasladados a la componente de Producción de Software, en donde se realizan las actividades claves para el desarrollo del producto final, tales como Análisis & Diseño, Programación y Pruebas. La componente de Soporte acompaña y brinda soporte en las diferentes actividades que se realizan a través del ciclo de producción de software.

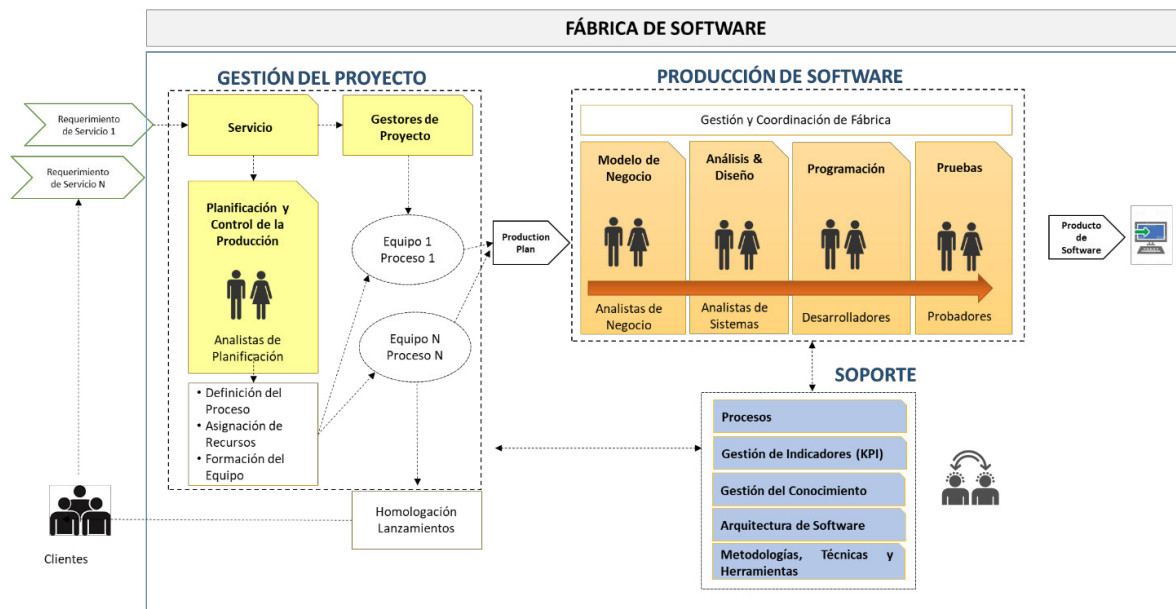


Figura 4.1. Estructura de Fábrica de Software adaptado de Nomura et al. (2006)

El 90% del esfuerzo de la Fábrica de Software es absorbida por la componente de Producción de Software, y de esta las unidades de trabajo Análisis & Diseño, Programación y Pruebas, consumen el 85% del esfuerzo de esta componente (Jacobson *et al.*, 2000), es por ello que el presente trabajo está orientado a la búsqueda de los factores que afectan la productividad en esta componente y las unidades de trabajo de Análisis & Diseño, Programación y Pruebas.

#### 4.2.2. Factores Propuestos

Se revisaron 40 teorías que han sido aplicados en la formación de equipos de trabajo, comportamiento de equipos, comportamiento organizacional y comunicación efectiva para conocer su relación con la productividad, descartándose aquellas que no tenían relación. Luego, se identificaron factores que afectan a la productividad con base en estas teorías y otras fuentes, varias de ellas ya identificadas en la literatura, obteniéndose 4 factores nuevos y 2 factores existentes que solo han sido aplicados a la unidad de trabajo de Programación, que describimos a continuación.

El factor *Uso de Reglas y Eventos (RULE)* se refiere al cumplimiento de normas y reglas de comportamiento que son definidas por el mismo equipo, lo cual permite que puedan cumplir metas y objetivos de manera clara y específica, y estos a su vez puedan ser medidos. Una de las buenas prácticas que se implementan en estas normas es el Time-boxing que “*asegura que los miembros del equipo no ocupen demasiado o muy poco trabajo durante un período de tiempo particular y no gasten su tiempo y energía en trabajos para los que tienen poca claridad*” (Guide to the Scrum Body of Knowledge [SBOK], 2016), siendo algunas de sus ventajas el proceso de desarrollo eficiente, menos gastos generales y la alta velocidad de los equipos, lo cual conlleva a una mejora de la productividad. Por ejemplo, el equipo debería participar de un evento llamado reunión diaria, que les permita ir revisando el avance del producto y esta reunión está determinada en un Time-boxing de 15 minutos, a fin de que sea una reunión efectiva.

El factor *Tipo de Comunicación “Cara a Cara” en el Equipo (FACE)*, se refiere a la forma de cómo las personas interactúan para mejorar el entendimiento de los requerimientos, facilitando el proceso de aprendizaje y colaboración en el equipo. (Simmons, 1991) define que la comunicación tiene una gran influencia en la productividad y, de acuerdo a la teoría Perspectiva de la Acción del Lenguaje (Flores & Ludlow, 1980), considera que el lenguaje es la dimensión principal de la actividad de cooperación humana. Por ejemplo, podría considerarse la importancia que le han dado a este factor en las metodologías ágiles, siendo, uno de los principios fundamentales del Manifiesto Ágil que “*el método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara*” (SBOK, 2016).

El factor ***Tipo de Comunicación “Osmótica” en el Equipo (OSMO)*** está referida a la información que fluye en el fondo auditivo de los miembros del equipo, de tal forma que ellos captan la información relevante, como si fuera por ósmosis, lo que se produce normalmente por estar en el mismo ambiente de trabajo. Así, cuando una persona hace una pregunta, otros en el ambiente pueden sintonizar o desconectarse, de esta forma se contribuye a la discusión o continúan con su trabajo. En la Teoría de la Memoria Transactiva, (Wegner, 1987) define que los miembros son capaces de beneficiarse de los conocimientos y experiencias de otros en caso de desarrollar una buena comprensión, a través del compartir en el grupo, siendo por ello necesario entender qué tan importante es la comunicación osmótica para la adquisición de conocimiento en ambientes de colaboración, lo cual ayuda al incremento de la productividad. También, se podría mencionar que este factor podría tener un efecto negativo, en caso el equipo de trabajo requiera una mayor tranquilidad y concentración.

El factor ***Heterogeneidad del Equipo (HETE)*** se refiere a la mejora del desempeño del equipo a través de la formación de equipos multifuncionales, de tal manera que facilita el aprovechamiento de conocimiento de todos los roles participantes. En Ancona y Caldwell (1992), se demuestra que la diversidad funcional en equipos puede influenciar el desempeño a través de procesos internos y comunicación externa. Los autores mencionan que “*mientras que la diversidad produce procesos que facilitan el desempeño, también impide el desempeño*”. Cohen y Bailey (1997) definen que la diversidad del equipo resulta en un mejor uso del conocimiento, mejor comunicación y cooperación en equipos, lo que conlleva a una mejor productividad del equipo.

El factor ***Tamaño del Equipo (SIZE)*** se refiere a la cantidad de participantes dentro del equipo de trabajo, de tal manera que sea lo necesario para asegurar las habilidades y capacidades requeridas y a su vez facilite la colaboración entre los miembros del equipo, ya que mientras mayor sean los canales de comunicación es más complejo el traslado de información hacia los diferentes niveles. De acuerdo a Rodríguez *et al.* (2011), el tamaño del equipo de software es considerado como uno de los drivers de productividad del proyecto. Asimismo, muchos autores han realizado estudios acerca de este factor y han determinado su impacto en la productividad del software, pero solo ha sido aplicado en la unidad de trabajo de Programación (Verner, Evanco & Cerpa, 2007; Brooks, 1995; Abdel-Hamid & Madnick,

1991; Abdel-Hamid, 1989; Montgomery, 1981; Smith, Hale & Parish, 2001; Trendowicz & Munch, 2009; Norden, 1958; Norden, 1977; Pillai & Sukumaran, 1997; Harman & Jones, 2001; Di Penta, Harman, Antoniol & Qureshi, 2007; Alba & Chicano, 2005; Alba & Chicano, 2007, Gueorguiev, Harman & Antoniol, 2009; Chicano, Luna, Nebro & Alba, 2011; Kremmel, Kubalik & Biffel, 2010; Kremmel, Kubalik & Biffel, 2011; Pendharkar & Rodger, 2007).

El factor *Confianza en el Equipo (TRUS)* se refiere a la capacidad de los miembros del equipo para confiar entre ellos, lo cual contribuye a una mayor cohesión y, en consecuencia, garantiza una mejor productividad. En Sridhar, Paul, Nath y Kapur (2007) encuentran que la confianza entre los miembros del equipo y sus comunicaciones efectivas tienen correlación positiva con el éxito del proyecto, lo cual afecta el desempeño del equipo de desarrollo. Asimismo, diversos autores han demostrado la importancia de la confianza en la productividad (Yilmaz & O'Connor, 2011; Sudhakar, Farooq & Patnaik, 2011).

Una síntesis de los factores introducidos se presenta en la Tabla 4.3.

Tabla 4.3.

Nuevos factores que impactan en la productividad

<b>ID</b>	<b>Factor</b>	<b>Descripción</b>	<b>Soporte</b>
<i>RULE</i>	<i>Uso de Reglas y Eventos.</i>	Cumplimiento de normas y reglas de comportamiento que son definidas por el mismo equipo.	Metodologías Ágiles (SBOK, 2016)
<i>FACE</i>	<i>Tipo de comunicación “cara a cara” en el equipo</i>	Método de comunicación que permite que las personas interactúen para mejorar el entendimiento de los requerimientos.	Metodologías Ágiles (SBOK, 2016); Perspectiva de la Acción del Lenguaje (Flores & Ludlow, 1980; Simmons, 1991)
<i>OSMO</i>	<i>Tipo de comunicación</i>	Aprendizaje a través de información que fluye en el fondo auditivo de los	Teoría de la Memoria Transactiva (Wegner, 1987)

<b>ID</b>	<b>Factor</b>	<b>Descripción</b>	<b>Soporte</b>
	<i>“osmótica” en el equipo.</i>	miembros del equipo, de tal forma que ellos captan la información relevante.	
<i>HETE</i>	<i>Heterogeneidad del equipo</i>	Formación de equipos de trabajo multifuncionales.	Ancona and Caldwell, 1992; Cohen & Bailey, 1997
<i>SIZE</i>	<i>Tamaño del equipo</i>	Número de participantes en el equipo de trabajo.	Rodríguez <i>et al.</i> , 2011; Verner <i>et al.</i> , 2007; Brooks, 1995; Abdel-Hamid & Madnick, 1991; Abdel-Hamid, 1989; Montgomery, 1981; Smith <i>et al.</i> , 2001; Trendowicz & Munch, 2009; Norden, 1958; Norden, 1977; Pillai & Sukumaran, 1997; Harman & Jones, 2001; Di Penta <i>et al.</i> , 2007; Alba & Chicano, 2005; Alba & Chicano, 2007, Gueorguiev <i>et al.</i> , 2009; Chicano <i>et al.</i> , 2011; Kremmel <i>et al.</i> , 2010; Kremmel <i>et al.</i> , 2011; Pendharkar & Rodger, 2007
<i>TRUS</i>	<i>Confianza en el equipo</i>	Capacidad de los miembros del equipo para confiar entre ellos.	Yilmaz & O’Connor, 2011; Sridhar <i>et al.</i> , 2007; Sudhakar <i>et al.</i> , 2011

### 4.2.3. Hipótesis

Después de identificar los factores, se estableció las relaciones con las unidades de trabajo.

#### 4.2.3.1. Influencia del factor *Uso de Reglas y Eventos (RULE)*

Dado que las actividades de análisis involucran una adecuada planificación y constante interacción con el cliente y con el equipo, de tal manera que se realice una buena elicitación

de los requerimientos y estos a su vez se transformen en el producto deseado, siendo por ello necesario que el analista se enfoque en delimitar el alcance del análisis, por lo que deben establecerse actividades prioritarias que ayuden al logro del objetivo y no se distraiga en actividades que pudiesen retrasar; por todo ello es necesario establecer reglas y eventos que den soporte al cumplimiento de estas actividades (SBOK, 2016).

***Hipótesis H1.1:*** *El factor Uso de Reglas y Eventos influye en la unidad de trabajo de Análisis & Diseño*

#### *4.2.3.2. Influencia del factor Heterogeneidad del Equipo (HETE)*

La diversidad en el equipo es una buena manera de incorporar nuevas perspectivas que pueden ayudar a los integrantes a superar problemas de distintos ángulos, es por ello que si en el equipo de Análisis & Diseño se cuenta con diversos perfiles, el resultado conllevará a una mejor comprensión de los requerimientos y a una forma diferente de plantear alternativas de solución. Asimismo, al ser la programación una actividad que incorpora diversas disciplinas, se hace necesario que el equipo sea multifuncional y que tenga habilidades y capacidades que permitan ser más eficientes en el desarrollo del producto.

***Hipótesis H2.1:*** *El factor Heterogeneidad del Equipo influye en la unidad de trabajo de Análisis & Diseño*

***Hipótesis H2.2:*** *El factor Heterogeneidad del Equipo influye en la unidad de trabajo de Programación*

#### *4.2.3.3. Influencia del factor Tipo de comunicación “cara a cara” en el equipo (FACE)*

La comunicación “cara a cara” es esencial para crear un vínculo entre los participantes del proyecto, convirtiéndose en el método más eficiente y eficaz de comunicación con y dentro de un equipo de desarrollo, ya que produce un intercambio de energía que dinamiza la experiencia humana y nos facilita la interacción entre muchas personas al mismo tiempo, lo que propicia mayor fluidez, efectividad y rapidez en los que respecta al diálogo, la discusión y la toma de decisiones.

***Hipótesis H3.1:*** *El factor Tipo de comunicación “cara a cara” influye en la unidad de trabajo de Análisis & Diseño*

#### 4.2.3.4. *Influencia del factor Tipo de comunicación “osmótica” en el equipo (OSMO)*

Los equipos de desarrollo normalmente trabajan en lugares co-localizados, por lo cual este tipo de comunicación se hace necesario en el equipo, y que se requiere que todos conozcan las actividades, a fin de evitar la dependencia y el equipo pueda avanzar.

***Hipótesis H4.2:*** *El factor Tipo de comunicación “osmótica” influye en la unidad de trabajo de Programación*

#### 4.2.3.5. *Influencia del factor Tamaño del Equipo (SIZE)*

El tamaño del equipo es un factor crítico en el desarrollo del software, ya que mientras más integrantes tengan un equipo, más canales de comunicación se generan, por ello es necesario que el equipo sea lo suficientemente grande para asegurar las habilidades adecuadas, pero lo suficientemente pequeño para facilitar la comunicación. Diversos estudios (Verner *et al.*, 2007; Brooks, 1995; Abdel-Hamid & Madnick, 1991; Abdel-Hamid, 1989; Montgomery, 1981; Smith *et al.*, 2001; Trendowicz & Munch, 2009; Norden, 1958; Norden, 1977; Pillai & Sukumara, 1997; Harman & Jones, 2001; Di Penta *et al.*, 2007; Alba & Chicano, 2005; Alba & Chicano, 2007, Gueorguiev *et al.*, 2009; Chicano *et al.*, 2011; Kremmel *et al.*, 2010; Kremmel *et al.*, 2011; Pendharkar & Rodger, 2007) han demostrado la influencia de este factor en la unidad de trabajo de Programación, es por ello que nos planteamos una interrogante acerca de su influencia en la unidad de trabajo de *Análisis & Diseño*.

***Hipótesis H5.1:*** *El factor Tamaño del Equipo influye en la unidad de trabajo de Análisis & Diseño*

***Hipótesis H5.2:*** *El factor Tamaño del Equipo influye en la unidad de trabajo de Programación*

#### 4.2.3.6. *Influencia del factor Confianza en el Equipo (TRUS)*

Un equipo de trabajo exitoso tiene sus cimientos en la base de la confianza. Cada miembro del equipo debe establecer la confianza, cultivándola a través de sus acciones, sus palabras y su trabajo para mantenerla. Cada miembro también necesita ser capaz de confiar en los miembros de su equipo para hacer un compromiso con ellos y sus objetivos.

***Hipótesis H6.1:*** *El factor Confianza en el Equipo influye en la unidad de trabajo de Análisis & Diseño*

**Hipótesis H6.2:** *El factor Confianza en el Equipo influye en la unidad de trabajo de Programación*

**Hipótesis H6.3:** *El factor Confianza en el Equipo influye en la unidad de trabajo de Pruebas*

En la Tabla 4.4, se resume las hipótesis planteadas.

Tabla 4.4.

Hipótesis planteadas para la definición de factores que afectan la productividad en el desarrollo de software

Factor	Unidad de Trabajo		
	Análisis & Diseño (ADI)	Programación (PRO)	Pruebas (TES)
<i>RULE</i>	H1.1		
<i>HETE</i>	H2.1	H2.2	
<i>FACE</i>	H3.1		
<i>OSMO</i>		H4.2	
<i>SIZE</i>	H5.1	H5.2	
<i>TRUS</i>	H6.1	H6.2	H6.3

#### 4.2.4. Modelo

Luego de revisar los diferentes factores y con base en los criterios establecidos, en la Figura 4.2, se puede visualizar el modelo conceptual de los factores propuestos y su incidencia en la productividad en las diferentes unidades de trabajo de la fábrica de software.



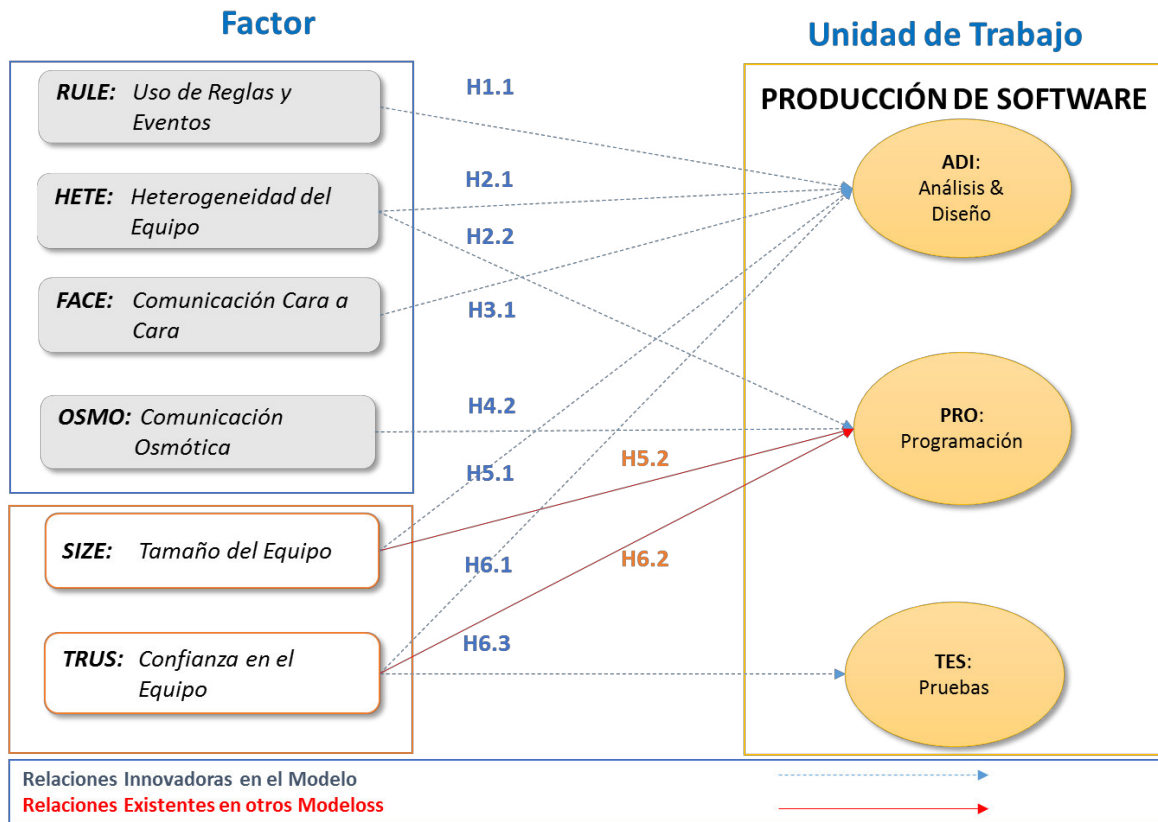


Figura 4.2. Modelo conceptual inicial de los factores que afectan la productividad

### 4.3. Metodología

#### 4.3.1. Recopilación de Datos

En la presente investigación se utilizó como instrumento de estudio el cuestionario, elaborándose una encuesta Online con Google Form. La encuesta fue aplicada en el período junio a octubre de 2017 a diferentes profesionales y expertos que participan en la supervisión y monitoreo en la fábrica de software. Este estudio ha sido desarrollado en organizaciones que han implementado modelos de fábricas de software en el Perú, que tienen un rol importante en la industria del software, y tienen operaciones en América del Sur. El objetivo de la encuesta ha sido determinar la percepción de los diferentes profesionales y expertos que interactúan en la fábrica de software en relación a los factores que influyen en la productividad.

La encuesta fue estructurada en 3 secciones: i) Sección 1 – Información General, consiste de cinco preguntas y está relacionada con la caracterización de la empresa y de las personas que llenan el cuestionario, ii) Sección 2 – Factores de Productividad de la Fábrica de Software, la cual corresponde a preguntas que determinan la influencia de los factores que afectan la productividad en las diferentes unidades de trabajo de la Fábrica de Software, la misma que ha sido establecida con base en una escala de Likert de cinco niveles de respuesta, donde: (1: No tiene Influencia, 2: Baja Influencia, 3: Mediana Influencia, 4: Alta Influencia y 5: Influye Totalmente), y; iii) Sección 3 – Otros Aspectos. Esta sección nos ayudará a tener un mayor entendimiento del impacto de los factores en la productividad de las unidades de trabajo de la fábrica de software. La encuesta se encuentra disponible en <http://tinyurl.com/y5kbpk7p>

Luego de elaborar la encuesta, se realizó una prueba piloto para validar las preguntas, donde participaron gerentes y jefes de proyectos senior, así, se revisó si las preguntas guardaban relación con las hipótesis planteadas, corrigiéndose la redacción de las preguntas y el lenguaje utilizado.

Se enviaron cartas y correos electrónicos invitando a la participación en las encuestas a la Asociación Peruana de Productores de Software (APESOFT), Cámara de Comercio de Lima (CCL), Advent International (GMD S.A.), instituciones públicas y privadas que tienen un área de TI, estudiantes de postgrado de la Universidad Nacional Mayor de San Marcos que tienen roles de supervisor en áreas de TI, entre otros. Se obtuvieron 160 encuestas, de los cuales se descartaron 10 por tener incongruencias en las respuestas.

#### **4.3.2. Análisis de Resultados**

El instrumento para el análisis de los datos fue el cuestionario, luego se realizaron los siguientes análisis estadísticos a los datos obtenidos: i) Análisis descriptivo de la muestra (análisis de aspectos demográficos de la encuesta: tamaño de empresa, sector del negocio y años de experiencia del encuestado, media, varianza, moda y la distribución de las respuestas de los encuestados), ii) Fiabilidad de datos (prueba de fiabilidad usando el Alfa de Cronbach), cuyo valor manifiesta la consistencia interna, es decir, muestra la correlación entre cada una de las preguntas, iii) Regresión Lineal Múltiple para la comprobación de las hipótesis planteadas y explicar el nivel de significancia entre las unidades de trabajo de la fábrica (variable dependiente) y los seis factores propuestos (variables independientes).

## 4.4. Resultados

### 4.4.1. Análisis Descriptivo de la Muestra

Habiéndose obtenido la participación de 150 personas, es necesario considerar cuál es el % de error muestral, de tal manera de establecer la validez del presente estudio. Al no contar con información histórica de estudios similares se hallará el error muestral en función de la varianza máxima posible, la cual asegura la validez (representatividad) de la muestra tomada. Para el cálculo del nivel error, se aplicará la siguiente fórmula de tamaño de muestra:

$$n = \frac{z^2 pq}{\epsilon^2} \quad (1)$$

Donde:

$n= 150$  (tamaño de muestra),  $z= 1.96$  (95% de nivel de confianza),  $p= 0.5$  (varianza máxima)

De donde:  $\epsilon^2 = \frac{z^2 pq}{n} = \frac{1.96^2 \times 0.5 \times 0.5}{150} = 0.0064$ , es decir  $\epsilon = 0.08$  (8% de error muestral)

Por lo tanto, podemos asegurar que para un nivel de confianza del 95%, un  $p$  de varianza máxima igual a 0.5, un  $q = 1-p = 0.5$ , y un error muestral del 8%, se tomó una muestra de 150 personas ( $n=150$ ).

Se realizó la difusión de las encuestas a diferentes organizaciones que brindan el servicio de fábrica de software en el Perú, las mismas que están caracterizadas de la siguiente manera:

- El 88.7% de los participantes laboran en fábricas de software cuyo tamaño oscila entre 10 y 500 personas, siendo solo el 11.3% de los participantes que laboran en empresas que tienen más de 500 personas, tal como se visualiza en la Figure 4.3.
- El 60% de los participantes laboran en fábricas de software que brindan servicios al Sector Gobierno, seguido de los sectores Transporte y Servicios con el 11.3%, respectivamente, y el 17.4% se encuentra dividida entre los demás sectores, tal como se visualiza en la Figure 4.3. Asimismo, el 65.3% de los participantes tienen entre uno a cinco años de experiencia en fábricas de software, y el 34.7% de los participantes cuentan con más de cinco años de experiencia.

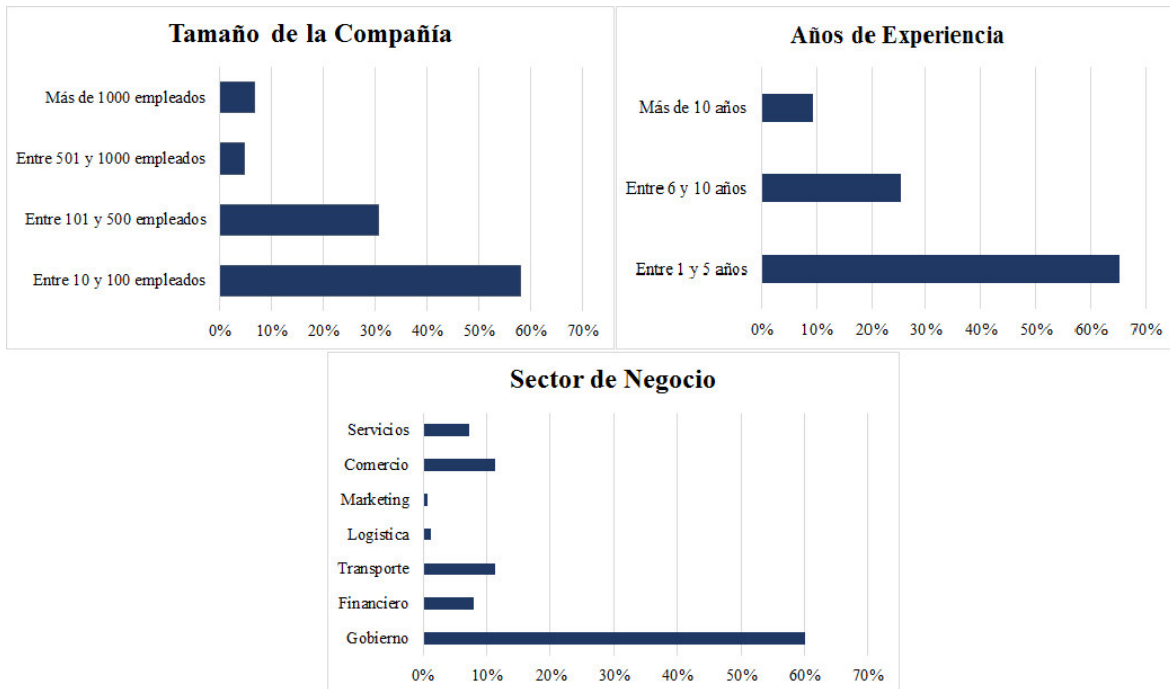


Figura 4.3. Datos de la caracterización de la muestra

#### 4.4.2. Fiabilidad de datos

Celina y Campo (2005) menciona que el valor aceptable para el coeficiente alfa de Cronbach es 0,7. Un valor superior a este revela una fuerte relación entre las preguntas, con lo cual la validez del instrumento es aceptable. Según los resultados obtenidos con la herramienta SPSS, se obtuvo un valor del alfa de Cronbach igual a 0,783 (ver Figura 4.4), que comprueba que la validez del instrumento es aceptable.

→

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,783	10

Figura 4.4. Estadísticas de fiabilidad

### 4.4.3. Regresión Lineal Múltiple

Para el análisis de los resultados, se tendrá en cuenta que si la significación de t-test es menor de 0.05, implica que la variable independiente se relaciona de forma significativa con la variable dependiente, por tanto, influye sobre ella, y es explicativa (Ramanathan, 2002).

#### 4.4.3.1. Componente: ADI-Análisis y Diseño de Software

Mediante la técnica de Regresión Lineal Múltiple explicaremos el nivel de significancia entre la variable dependiente (ADI), y las seis variables independientes (RULE, HETE, FACE, OSMO, SIZE, TRUS), a fin de determinar cuáles son las variables que tienen mayor impacto en la productividad de dicha componente.

Tabla 4.5.

Regresión de la variable dependiente (ADI) con todas las variables independientes

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>RULE</i>	0,154	0,069	2,240	0,027
<i>HETE</i>	-0,141	0,061	-2,312	0,022
<i>FACE</i>	0,068	0,087	0,781	0,436
<i>OSMO</i>	-0,005	0,059	-0,087	0,931
<i>SIZE</i>	-0,056	0,072	-0,778	0,438
<i>TRUS</i>	0,235	0,078	3,009	0,003

Tal como se muestra en la Tabla 4.5, las variables más significantes son las siguientes: RULE, HETE y TRUS.

Para tener un modelo predictivo y teniendo en consideración estas variables significativas, se realiza la regresión lineal, así, se obtiene los resultados de la Tabla 4.6.

Tabla 4.6.

Regresión de la variable dependiente (ADI) con las variables independientes significativas

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>RULE</i>	0,163	0,054	2,991	0,003
<i>HETE</i>	-0,129	0,053	-2,436	0,016
<i>TRUS</i>	0,243	0,049	5,002	0,000

El modelo predictivo se muestra a continuación:

$$ADI = 0,163 \text{ RULE} - 0,129 \text{ HETE} + 0,243 \text{ TRUS}$$

Interpretando el modelo se puede decir:

- Cada vez que el factor *RULE* se incremente en un punto (intensidad), *ADI* se incrementará en 0.163, es decir, el mayor Uso de Reglas y Eventos conllevará a un mejor desempeño de las actividades del equipo.
- Al incrementarse en una unidad el nivel de influencia por efecto del Heterogeneidad del Equipo (*HETE*), entonces el impacto sobre el Componente de Análisis y Diseño decrece en 0.129 unidades, es decir, que a menos heterogeneidad en el equipo, se genera mayor eficiencia en los resultados.
- Cada vez que el factor *TRUS* se incremente en un punto (intensidad), *ADI* se incrementará en 0.243, es decir, que una mayor Confianza en el Equipo conllevará a un mejor desempeño en el logro de resultados.

#### 4.4.3.2. *Componente: PRO-Programación de Software*

Mediante la técnica de Regresión Lineal Múltiple explicaremos el nivel de significancia entre la variable dependiente (*PRO*), y las seis variables independientes (*RULE*, *HETE*, *FACE*, *OSMO*, *SIZE*, *TRUS*), a fin de determinar cuáles son las variables que tienen mayor impacto en la productividad en la Componente de Programación de Software.

Tabla 4.7.

Regresión de la variable dependiente (PRO) con todas las variables independientes

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>RULE</i>	-0,048	0,052	-0,923	0,357
<i>HETE</i>	0,100	0,048	2,062	0,041
<i>FACE</i>	0,061	0,047	1,301	0,195
<i>OSMO</i>	0,120	0,051	2,182	0,032
<i>SIZE</i>	0,125	0,056	2,225	0,028
<i>TRUS</i>	0,170	0,058	2,932	0,004

Tal como se muestra en la Tabla 4.7, las variables más significantes son: HETE, OSMO, SIZE y TRUS.

Para tener un modelo predictivo y teniendo en consideración estas variables significativas, se realiza la regresión lineal, así, se obtiene los resultados de la Tabla 4.8.

Tabla 4.8.

Regresión de la variable dependiente (PRO) con las variables independientes significativas

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>HETE</i>	0,076	0,040	1,894	0,060
<i>OSMO</i>	0,115	0,042	2,282	0,001
<i>SIZE</i>	0,147	0,045	3,282	0,001
<i>TRUS</i>	0,157	0,046	3,418	0,001

El modelo predictivo se muestra a continuación:

$$PRO = 0,076 HETE + 0,115 OSMO + 0,147 SIZE + 0,157 TRUS$$

Tal como se puede visualizar, las cuatro variables influyen positivamente en la productividad de la unidad de trabajo de Programación.

#### 4.4.3.3. Componente: TES-Pruebas de Software

Mediante la técnica de Regresión Lineal Múltiple explicaremos el nivel de significancia entre la variable dependiente (TES), y las seis variables independientes (RULE, HETE, FACE,

OSMO, SIZE, TRUS), a fin de determinar cuáles son las variables que tienen mayor impacto en la productividad en la Componente de Pruebas de Software.

Tabla 4.9.

Regresión de la variable dependiente (TES) con todas las variables independientes

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>RULE</i>	0,039	0,034	1,151	0,252
<i>HETE</i>	0,040	0,035	1,159	0,248
<i>FACE</i>	-0,006	0,034	-0,174	0,862
<i>OSMO</i>	-0,022	0,033	-0,678	0,499
<i>SIZE</i>	-0,004	0,034	-0,125	0,901
<i>TRUS</i>	-0,085	0,037	-2,271	0,025

Tal como se muestra en la Tabla 4.9, la variable más significativa es TRUS.

Para tener un modelo predictivo y teniendo en consideración estas variables significativas, se realiza la regresión lineal, así, se obtiene los resultados de la Tabla 4.10.

Tabla 4.10.

Regresión de la variable dependiente (TES) con las variables independientes significativas

<b>Factor</b>	<b>B</b>	<b>Error estándar</b>	<b>t</b>	<b>Sig.</b>
<i>TRUS</i>	0,044	0,007	6,071	0,000

El modelo predictivo se muestra a continuación:

$$TES = 0,044 TRUS$$

Como se observa:

- Cada vez que el factor TRUS se incrementa en un punto (intensidad), TES se incrementará en 0.044, es decir, que una mayor Confianza en el Equipo conllevará a un mejor desempeño en el logro de resultados.



## 4.5. Discusión

### 4.5.1. Unidades de Trabajo

Tal como se ha podido observar en los diferentes estudios, no existe un estándar acerca de cómo se debe organizar una fábrica de software, lo cual dificulta el análisis de la medición de la productividad. Normalmente, los estudios delimitan la fábrica de software solo a la producción de software, lo cual tiene varias desventajas, ya que no se contemplaría la gestión de proyecto y otras unidades que interactúan en ella, lo cual podría sesgar la interpretación de la productividad.

Se ha podido identificar que una fábrica de software se organiza en tres grandes componentes: *Gestión del Proyecto, Producción de Software y Soporte*, y estos a su vez pueden estar divididos en unidades de trabajo (*Servicio, Gestores de Proyecto, Planificación y Control de la Producción, Modelo de Negocio, Análisis & Diseño, Programación, Pruebas, Procesos, Gestión de Indicadores, Gestión del Conocimiento, Arquitectura, Metodologías, Técnicas y Herramientas*), es decir, las áreas en las cuales se realizan las actividades necesarias para el cumplimiento del objetivo del componente.

Además, se ha podido determinar que en el Perú las fábricas de software se han implementado en su mayor parte en entidades públicas (60%). También, se puede apreciar que el enfoque de fábrica de software en el Perú es muy reciente, tal como lo corrobora el 65.3% de los participantes, quienes cuentan en promedio con cinco años de experiencia.

### 4.5.2. Factores

De los datos obtenidos en nuestro estudio, se ha evidenciado que el factor *Uso de Reglas y Eventos (RULE)* tiene una influencia significativa positiva en la unidad de trabajo de Análisis & Diseño, lo cual puede deberse a que el analista requiere establecer una adecuada planificación de las actividades a realizar y estas tienen que estar alineadas al alcance del requerimiento, por lo cual el analista no debería distraerse en actividades accesorias que impidan el cumplimiento del objetivo, siendo por ello de mucha importancia establecer reglas que den soporte y precisen estas actividades.

El factor *Heterogeneidad del Equipo (HETE)* tiene una influencia negativa en la unidad de trabajo de Análisis & Diseño, sin embargo, tiene una influencia significativa positiva en la

unidad de trabajo de Programación, el cual podría explicarse en que para las tareas de análisis en general se requiere especialistas en tipos de negocios, por lo que involucrar más especialistas reduciría la productividad, sin embargo, en las tareas de programación se requiere diferentes habilidades y capacidades para el desarrollo de un producto.

El factor ***Tipo de comunicación “cara a cara” en equipo (FACE)*** no tiene un impacto significativo en la unidad de trabajo de Análisis & Diseño, así, se descarta la hipótesis inicial que había sido sustentada con base en la teoría Perspectiva de la Acción del Lenguaje (Flores & Ludlow, 1980), lo cual se podría explicar por el incremento del uso de los medios tecnológicos para la interacción. La interacción entre los participantes en el desarrollo de software es muy importante para el logro de los objetivos, sin embargo, la versión presencial está siendo superada por los medios virtuales.

El factor ***Tipo de comunicación osmótica en equipo (OSMO)*** tiene una influencia significativa positiva en la unidad de trabajo de Programación, lo cual se debe a que las actividades de programación requieren diferentes habilidades y capacidades, lo que genera que mientras más personas tengan conocimiento de lo que están haciendo los demás integrantes del equipo, más fácil será la integración del producto y mayor contingencia se tendrá en caso de ausencia de algún miembro del equipo, por lo que ello conllevará a una mayor productividad.

Se corrobora el impacto del factor ***Tamaño del Equipo (SIZE)*** en la unidad de trabajo de Programación (Verner *et al.*, 2007; Brooks, 1995; Abdel-Hamid & Madnick, 1991; Abdel-Hamid, 1989; Montgomery, 1981; Smith *et al.*, 2001; Trendowicz & Munch, 2009; Norden, 1958; Norden, 1977; Pillai & Sukumaran, 1997; Harman & Jones, 2001; Di Penta *et al.*, 2007; Alba & Chicano, 2005; Alba & Chicano, 2007, Gueorguiev *et al.*, 2009; Chicano *et al.*, 2011; Kremmel *et al.*, 2010; Kremmel *et al.*, 2011; Pendharkar & Rodger, 2007), sin embargo, no hay significancia de este factor sobre la unidad de trabajo de Análisis & Diseño.

El factor ***Confianza en el Equipo (TRUS)*** tiene una influencia significativa positiva en las tres unidades de trabajo, lo cual refuerza la necesidad de diseñar estrategias que mejoren la confianza entre los miembros del equipo, que permita el logro del objetivo que es la entrega del producto final, la cual se da a través del trabajo integrado entre todas las unidades de la fábricas.

Si bien los otros factores analizados también tienen un nivel de influencia positiva o negativa en la productividad de la fábrica de software, los valores obtenidos no son lo suficientemente significativos para ser considerados; esto implica que a pesar que estos factores sean gestionados correcta o incorrectamente, su influencia en la productividad no va a generar un alto impacto. Por ejemplo, el uso del tipo de comunicación osmótica en el equipo (OSMO) genera una mejora en la productividad en la unidad de trabajo de Análisis & Diseño, sin embargo, esta influencia no será muy significativa.

Finalmente, en la Figura 4.5, se puede visualizar el modelo conceptual final con los factores que tienen un mayor impacto en cada una de las unidades de trabajo de la fábrica de software.

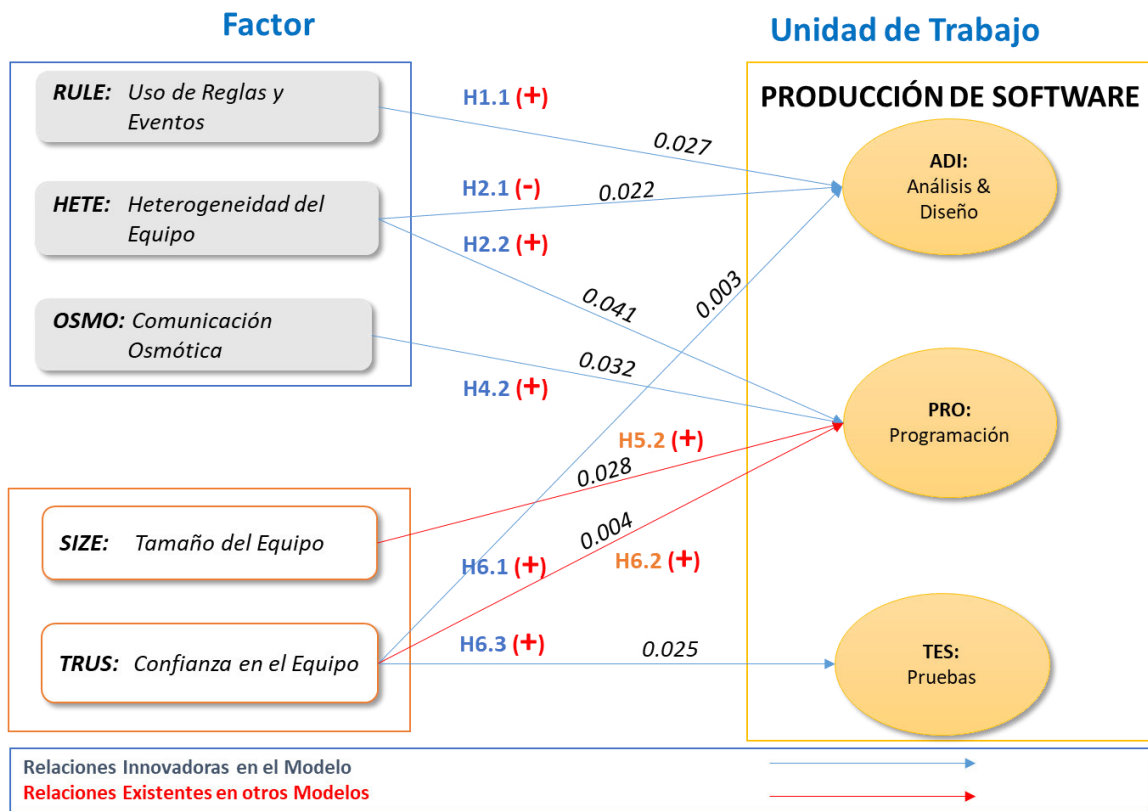


Figura 4.5. Modelo final de factores que afectan la productividad

### 4.5.3. Limitaciones del Estudio

Este estudio ha utilizado la técnica de la encuestas para obtener los resultados mostrados, por lo que está sujeta a algunas limitaciones: a) El estudio experimental ha utilizado la técnica de

la encuesta a los involucrados en el proceso de desarrollo de software del Perú, y b) El estudio ha sido aplicado solo a la componente de Producción de Software.

#### **4.5.4. Trabajos Futuros**

Futuros estudios podrían considerar analizar los factores que afectan a la unidad de trabajo de Modelo de Negocio de la componente Producción de Software, así como a las componentes de Gestión de Proyecto y Soporte, de tal manera que se pueda tener un estudio integral de la fábrica de software. Asimismo, un aspecto a tener en cuenta en futuros trabajos sería realizar un análisis del factor Tipo de comunicación “cara a cara” en equipo (FACE) para determinar su impacto en entornos de trabajo virtuales y presenciales. También, es recomendable que se cuente con una base de datos de la información que se genera en los proyectos de fábricas de software, de tal manera que facilite los estudios de investigación en esta área, para establecer estrategias que permitan potenciar los factores positivos y reducir los factores negativos, a fin de mitigar su impacto en la productividad.

#### **4.6. Conclusiones**

En este trabajo se ha introducido tres nuevos factores que influyen en la productividad de software sustentadas en teorías y se han estudiado la influencia de los factores *Tamaño del equipo (SIZE)* y *Confianza en el equipo (TRUS)* en las diferentes unidades de trabajo de la componente de la fábrica de software.

Los resultados obtenidos de la encuesta administrada a 150 profesionales involucrados en actividades de desarrollo de software en entorno de fábrica en el Perú corroboraron la influencia de los factores propuestos como en la unidad de trabajo de Análisis & Diseño, *Uso de reglas y eventos (RULE)* y *Confianza en el equipo (TRUS)* tienen una influencia significativa positiva, y la *Heterogeneidad del equipo (HETE)* tiene una influencia negativa. Por otro lado, en la unidad de Programación, *Heterogeneidad del equipo (HETE)*, *Tipo de comunicación osmótica en equipo (OSMO)*, *Tamaño del equipo (SIZE)*, y *Confianza en el equipo (TRUS)* tienen una influencia significativa positiva. También, en la unidad de trabajo de Pruebas, se ha demostrado que *Confianza en el equipo (TRUS)* tiene una influencia significativa positiva.

Asimismo, se puede visualizar que el factor *Confianza en el equipo (TRUS)* afecta a las tres unidades de producción. También, se reafirma que el factor *Tamaño del equipo (SIZE)* tiene una influencia significativa positiva en la unidad de trabajo de Programación, sin embargo, no tiene este mismo efecto en las demás unidades.

## **Capítulo 5: Modelo DEA Propuesto para la Medición de la Productividad de la Fábrica de Software**

En el presente capítulo se propone un modelo para la medición de la productividad basado en Análisis Envoltorio de Datos con Orientación de Entrada y Retornos de Escala Variable (DEA VRS-I). El modelo propuesto consta de dos fases: en la fase 1 considera la evaluación de las fábricas de software considerando el Esfuerzo Total y Costo del Proyecto como entradas y Puntos de Función como salida; y la fase 2 evalúa la productividad de los proyectos que realiza la fábrica de software.

### **5.1. Introducción**

La productividad es definida como la razón entre una salida que se produce y la entrada que se usa para producir la salida (Coelli, Rao, O'Donnell, & Battese, 2005).

No hay un estándar en la definición de la productividad, sin embargo, los autores tienen un aspecto común, ya que consideran que la productividad es el ratio del producto obtenido entre el esfuerzo, sin embargo, los estudios realizados sobre la productividad de una fábrica de software se orientan a medir la productividad en la unidad de trabajo de programación de software, tal como se puede apreciar en las entradas y salidas que han sido evaluadas (SLOC, FP, PH, PM), lo cual ha conllevado a que se hayan desarrollado muchos modelos de medición del desempeño que han llegado a ser inexactos. Sin embargo, los estudios no han revisado de manera integral a la fábrica de software, obviando otras unidades de trabajo también importantes como Análisis & Diseño y Pruebas.

Es en este contexto que definimos la productividad de las fábricas de software *como “un indicador de la eficiencia de los recursos utilizados en diferentes unidades de trabajo de la fábrica de software para la consecución del producto final”*. Se ha planteado esta definición, ya que considera todas las unidades de trabajo que participan en el desarrollo de software, y, además, permite evaluar los diferentes factores que influyen en todo el ciclo de la producción del software.

Cardoso *et al.* (2010) afirman que un modelo es una representación simplificada de la realidad, definiéndose a través de los diferentes componentes y procesos que forman parte del sistema en estudio. Se han identificado trece modelos que miden la productividad en el desarrollo de software, los cuales están basados en los siguientes enfoques: DEA, CMMI & Six Sigma, Reutilización de Software, SEM, Factor Total de Productividad, IEEE Std. 1045 & ISO 9216-4.

## **5.2. Modelos de Productividad DEA**

DEA es una técnica de programación matemática ampliamente utilizada que fue desarrollada originalmente por Charnes, Cooper y Rhodes (1978) y fue extendido por Banker, Charnes y Cooper (1984) para incluir retornos de escala variable. DEA generaliza la medida de eficiencia técnica de Farrell (1957) para evaluar la eficiencia relativa de unidades pares con respecto a múltiples medidas de rendimiento (Charnes, Cooper, Lewin & Seiford, 1994; Cooper, Seiford & Tone, 2006). Las unidades bajo evaluación en DEA se llaman unidades de toma de decisiones (DMU). Una DMU se considera eficiente cuando ninguna otra unidad DMU puede producir más salidas usando una cantidad igual o menor de entradas. La DEA es una técnica de medición de la eficiencia basada en la obtención de una frontera de eficiencia a partir del conjunto de observaciones que se considere sin la estimación de ninguna función de producción (Charnes *et al.*, 1997). A diferencia de los métodos paramétricos que requieren un conocimiento detallado del proceso, DEA no es paramétrico y no requieren una forma funcional explícita relacionada con las entradas y salidas (Cooper *et al.*, 2006; Cook & Seiford, 2009).

Desde su introducción, la investigación en DEA ha ido evolucionando, tanto en el ámbito teórico como en la práctica, encontrándose trabajos en diversas áreas, tales como medición del desempeño de instituciones educativas, benchmarking de procesos logísticos, comparación de sucursales de oficinas regionales de bancos, regulación de servicios públicos, medición de productividad investigativa y docente en departamentos académicos, así como en la industria del software, tal como se puede apreciar en el resumen de la Tabla 5.1, en el cual se actualiza la revisión de literatura elaborada por (Pai *et al.*, 2015).

Tabla 5.1.

Estudios de Proyectos de Software aplicando DEA

<b>Id</b>	<b>Entrada</b>	<b>Salida</b>	<b>Modelo</b>	<b>Unidad de Trabajo</b>	<b>Tamaño de Muestra</b>	<b>Referencia</b>
<i>1</i>	Esfuerzo	Puntos Función	de DEA Constant Returns to Scale	Programación	8 conjunto de datos, cada uno con 15 – 63 proyectos	Banker & Kemerer, 1989
<i>2</i>	Esfuerzo	Puntos Función, Calidad	de Stochastic DEA	Programación	65 proyectos	Banker, Datar & Kemerer, 1991
<i>3</i>	Esfuerzo	Puntos Función	de DEA-based F-tests	Programación	11 conjunto de datos, cada uno con 15 – 63 proyectos	Banker, Chang & Kemerer, 1994
<i>4</i>	Esfuerzo	Comment Change, Code Change, Routines Change	Additive DEA Constant Returns to Scale	Programación	78 proyectos	Mahmood, Pettingell & Shaskevich, 1996
<i>5</i>	Costo del Proyecto	Tamaño (Puntos de Función), Calidad (Defectos),		(Programación – Pruebas)	11 proyectos	Paradi, Reese & Rosen, 1997



<b>Id</b>	<b>Entrada</b>	<b>Salida</b>	<b>Modelo</b>	<b>Unidad de Trabajo</b>	<b>Tamaño de Muestra</b>	<b>Referencia</b>
		Duración (Tiempo de Comercialización en Días)				
<b>6</b>	Esfuerzo	Usuarios, EDI, Conversiones	DEA Returns to Scale	Variable Soporte	30 proyectos COTS (por las siglas del inglés Commercial-off-the-shelf)	Myrtveit & Stensrud, 1999
<b>7</b>	Esfuerzo	Puntos de Función, Ubicaciones, Unidades de Negocio, Usuarios Concurrentes	DEA Returns to Scale	Variable (Programación – Soporte)	114 proyectos	Flitman, 2003
<b>8</b>	Esfuerzo	Usuarios, EDI, Conversiones	DEA Returns to Scale	Variable Soporte	30 proyectos Enterprise Resource Planning (ERP)	Stensrud & Myrtveit, 2003
<b>9</b>	Esfuerzo, Costos del	Líneas de Código Fuente		Programación	70 proyectos	Yang & Paradi, 2004

<b>Id</b>	<b>Entrada</b>	<b>Salida</b>	<b>Modelo</b>	<b>Unidad de Trabajo</b>	<b>Tamaño de Muestra</b>	<b>Referencia</b>
	Vendedor, Duración					
<b>10</b>	Esfuerzo	Puntos de Función		Programación	144 proyectos	Asmild <i>et al.</i> , 2006
<b>11</b>	Esfuerzo	Puntos de Función, Líneas de Código Fuente	DEA & Regression Analysis	Soporte	10 proyectos ERP (por las siglas del inglés Enterprise Resource Planning)	Parthasarathy & Anbazhagan, 2008
<b>12</b>	Desarrolladores, Incidencias	Kilobytes por Descarga, Descargas, Rango del Esfuerzo del Proyecto, Puntos de Función como una Variable Contextual		(Programación)	34 proyectos	Wray & Mathieu, 2008

<b>Id</b>	<b>Entrada</b>	<b>Salida</b>	<b>Modelo</b>	<b>Unidad de Trabajo</b>	<b>Tamaño de Muestra</b>	<b>Referencia</b>
<b>13</b>	Esfuerzo, Tamaño del Equipo	Tamaño (Puntos de Función)	DEA Returns to Scale	Variable Programación	540 proyectos	Pendharkar & Rodger, 2009
<b>14</b>	Horas de Labor Variable	Métricas de Complejidad UML	DEA Returns to Scale	Variable (Análisis & Diseño)	25 proyectos	Cao <i>et al.</i> , 2012
<b>15</b>	Esfuerzo de Desarrollo (DE), Esfuerzo de Revisión y Pruebas (EoC), y Esfuerzo de Retrabajo (EoNC)	Puntos de Función, Calidad (f{Defectos})	DEA Returns to Scale	Variable (Programación – Pruebas)	79 proyectos	Pai <i>et al.</i> , 2015

Por lo que se puede apreciar en la Tabla 5.1, la mayoría de los trabajos que evalúan la productividad en proyectos de desarrollo de software usando DEA consideran el modelo VRS y solo consideran la unidad de trabajo de Programación, lo cual sesga la medición de la productividad, ya que para el desarrollo de software se involucran otras unidades de trabajo muy importantes como el Análisis & Diseño y Pruebas. Consideramos utilizar la metodología DEA por los motivos sustentados porque:

- Desarrolla un proceso de optimización para cada observación individual con el objetivo de calcular una frontera discreta determinada por las DMU's eficientes.
- No requiere supuesto alguno sobre la forma funcional de la frontera.
- Las DMU's ineficientes se proyectan sobre la frontera mediante una combinación convexa de las DMU's eficientes más próximas.
- Permite identificar las fuentes y el nivel de ineficiencia de los inputs y outputs.

### **5.3. Metodología**

El propósito de la DEA es construir una frontera no paramétrica por tramos sobre el conjunto de datos de tal manera que todos los puntos observados se encuentren en la frontera de producción o debajo de ella, ya que permite calcular las eficiencias relativas de todas las DMU con respecto a la frontera. Cada DMU, no posicionada en la frontera, se reduce a una combinación convexa de DMU en la faceta fronteriza más cercana (Charnes et al., 1978; Sahoo, 2016). Los diferentes aspectos que se han tenido en cuenta para la definición del modelo de la DEA y los datos que se deben considerar para su evaluación se analizan con mayor detalle en la subsección a continuación.

#### **5.3.1. Elección del modelo de la DEA**

Los siguientes párrafos describen las consideraciones al definir el modelo DEA.

##### **5.3.1.1. Isotonicidad**

Una regla fundamental para los parámetros DEA es que un aumento en una variable de entrada debe mejorar cada uno de las salidas (Chung, Lee, Kang & Lai, 2008), a esto se le llama propiedad de isotonicidad de parámetros DEA. Se debe realizar análisis de correlación para asegurar relaciones positivas entre entradas y salidas. En caso de correlaciones

negativas, se pueden aplicar técnicas de transformaciones. Sin embargo, varios problemas relacionados con la transformación de variables para satisfacer la propiedad de isotonicidad se han descrito en (Liu, Meng, Li & Zhang, 2010). Por lo tanto, en lugar de conducir a la transformación de parámetros, correlaciones negativas pueden ser una indicación de que uno o más de los parámetros deben ser excluidos del modelo. Para el presente caso de estudio se han considerado dos variables de entrada (Esfuerzo, Costo del Proyecto) y una variable de salida (Puntos de Función) que ya han sido utilizadas en estudios anteriores (Banker & Kemerer, 1989; Banker *et al.*, 1991; Banker *et al.*, 1994; Mahmood *et al.*, 1996; Myrtveit & Stensrud, 1999; Flitman, 2003; Stensrud & Myrtveit, 2003; Yang & Paradi, 2004; Asmild *et al.*, 2006; Parthasarathy & Anbazhagan, 2008; Pendharkar & Rodger, 2009; Pai *et al.*, 2015; Paradi *et al.*, 1997), por lo cual no es necesario establecer la isotonicidad de los parámetros.

#### **5.3.1.2. Orientación del Modelo DEA**

DEA proporciona dos básicas orientaciones del modelo: maximización de la salida y minimización de la entrada. Un modelo orientado a maximizar la salida determina el máximo aumento proporcional de las salidas en relación con los valores de entrada, que es adecuado para establecer un conjunto de valores de salida objetivo. Un modelo orientado a minimizar la entrada determina la cantidad por la cual los valores de entrada se pueden disminuir mientras se sigue produciendo las mismas salidas. Esta investigación adopta un modelo orientado a las entradas para determinar la eficiencia basada en el uso de parámetros de entrada para producir los valores de calidad y productividad resultantes.

#### **5.3.1.3. Selección del Modelo DEA**

Ahora, mediante DEA buscaremos determinar si podemos crear una fábrica de software ideal (virtual) que sea mejor que una o más de las fábricas de software reales. Cualquier fábrica dominada será una fábrica ineficiente.

Para el caso del presente estudio, se tomará en cuenta el modelo DEA Retornos de Escala Variable (DEA VRS), ya que se considera que algunas fábricas de software de tamaño diferente al de las eficientes, pueden no ser capaces de conseguir la productividad de estas. Es así que el estudio se realizará mediante la eficiencia técnica, que toma en cuenta que cada DMU debe tomar como referencia a la productividad mayor de entre las de su tamaño.

Debido a que los contratos establecidos con las fábricas de software implican la entrega de productos definidos al inicio del servicio, se ha considerado la Orientación de Entrada, por el hecho de que las fábricas de software deben alcanzar la productividad de la unidad de referencia a costa de reducir las horas de esfuerzo y el costo del proyecto real, y adicionalmente consideramos que las variables de entrada pueden ser controladas.

### 5.4. Modelo DEA

#### 5.4.1. Parámetros para el Modelo DEA

La Figura 5.1 muestra los parámetros que componen el modelo DEA: *DMU*, *Entrada*, *Modelo*, *Salida*, *Fases*. La fábrica será evaluada en dos fases: la fase 1 busca determinar la productividad de las fábricas de software, con el fin de conocer la más eficiente, lo cual permitirá conocer la brecha existente y las buenas prácticas que se vienen aplicando, de tal manera que sean replicadas a las otras fábricas de software, que conlleven a una mejora de su eficiencia. La fase 2 determinará la productividad de los proyectos de fábrica de software, con el fin de obtener buenas prácticas o lecciones aprendidas de los proyectos más eficientes, que se conviertan en modelos a seguir por los proyectos que están en su categoría y que podrían lograr la eficiencia de estos.

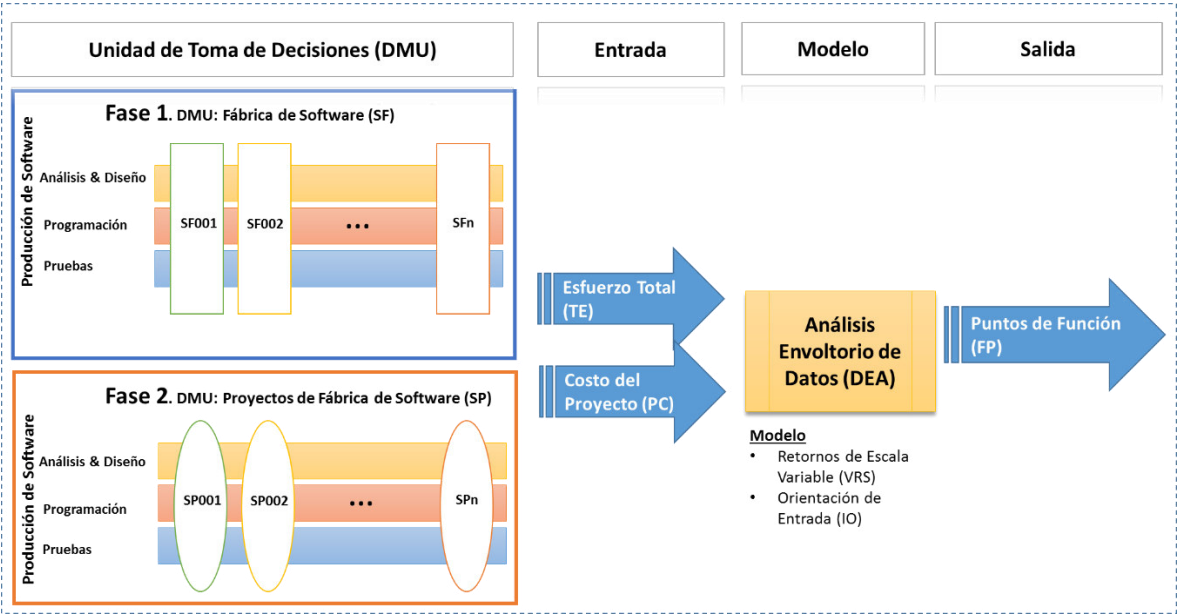


Figura 5.1. Parámetros para el Modelo DEA

A continuación, se describen los parámetros del modelo DEA:

a. Unidad de Toma de Decisiones (DMU). Se define como cualquier organización que genera productos o servicios consumiendo ciertos recursos, con la capacidad de poder modificar tanto los recursos consumidos (entradas) como la producción creada (salidas). Para el caso planteado, se ha considerado la evaluación en dos fases:

- Fase 1: DMU – Fábrica de Software. Considera la información de las diferentes organizaciones que vienen trabajando en el modelo de Fábrica de Software.
- Fase 2: DMU – Proyectos de Fábrica de Software. Considera la información de los proyectos desarrollados dentro de cada Fábrica de Software.

Para las dos fases, se ha recolectado la información de las actividades que se ejecutan dentro de la componente de Producción de Software, así, se tiene en cuenta solo las siguientes unidades de trabajo: Análisis & Diseño, Programación y Pruebas.

b. Entrada. Es la variable de entrada que será evaluada por el modelo DEA. En el caso planteado se ha considerado dos variables para las dos fases, y que han sido ampliamente estudiados para la evaluación de la eficiencia en proyectos de desarrollo de software, siendo estas:

- Esfuerzo Total (TE). Es el esfuerzo de trabajo medido en horas (hrs.), que ha sido utilizado en las fábricas de software y en sus respectivos proyectos, donde se ha considerado el esfuerzo de las unidades de trabajo de Análisis & Diseño, Programación y Pruebas. Esta variable ha sido ampliamente utilizada en los estudios de (Banker & Kemerer, 1989; Banker *et al.*, 1991; Banker *et al.*, 1994; Mahmood *et al.*, 1996; Myrtveit & Stensrud, 1999; Flitman, 2003; Stensrud & Myrtveit, 2003; Yang & Paradi, 2004; Asmild *et al.*, 2006; Parthasarathy & Anbazhagan, 2008; Pendharkar & Rodger, 2009; Pai *et al.*, 2015)
- Costo del Proyecto (PC). Es el costo del proyecto expresado en dólares americanos (US\$), que incluye los costos incurridos en las fábricas de software y sus proyectos, a través de las unidades de trabajo de Producción de Software. Esta variable ha sido analizado y utilizado en (Paradi *et al.*, 1997).

c. Salida. Es la variable de salida que será evaluada por el modelo DEA. En el caso planteado se ha considerado la variable Puntos de Función (FP) que representa al tamaño del software, por ser la unidad de medida que se utiliza en las diferentes fábricas de

software y sus proyectos, el cual ha sido obtenido a través de los equipos de estimación (responsables de contabilizar FP's al final del proyecto). El proceso de desarrollo especifica que los FP's son estimados al inicio y fin del proyecto, para poder evidenciar los cambios a través del proyecto. Esta variable ha sido ampliamente utilizada en los estudios de (Banker & Kemerer, 1989; Banker *et al.*, 1991; Banker *et al.*, 1994; Flitman, 2003; Asmild *et al.*, 2006; Parthasarathy & Anbazhagan, 2008; Pendharkar & Rodger, 2009; Pai *et al.*, 2015)

- d. Modelo. Para el estudio se utilizará el modelo DEA VRS-I (Banker *et al.* 1984), ya que se busca minimizar las entradas (Esfuerzo Total y Costo del Proyecto) y mantener las salidas actuales (número de Puntos de Función) de las fábricas de software y sus respectivos proyectos.
- e. Fases. Se ha considerado identificar eficiencias en dos fases: Fase 1: Fábrica de Software y Fase 2: Proyectos de Fábrica de Software.

**Fase 1.** En esta fase se busca identificar las fábricas de software más eficientes, a través de la minimización del *Esfuerzo Total* y *Costo del Proyecto* consumidos en un período determinado por las fábricas de software, pero se debe mantener la cantidad de *Puntos de Función*, por lo que utilizamos DEA VRS-I, donde las entradas son minimizadas y las salidas se mantienen en su nivel actual. Este modelo asume 2 entradas, 1 salida, y  $n$  DMU's, tal como se muestra a continuación:

$$\text{Minimizar } \theta \tag{1}$$

*s.t.*

$$\sum_{j=1}^6 \lambda_j \cdot x_{ij} \leq \theta x_{io} \quad i = 1,2; \tag{2}$$

$$\sum_{j=1}^6 \lambda_j \cdot y_j \geq y_o; \tag{3}$$

$$\sum_{j=1}^6 \lambda_j = 1; \tag{4}$$

$$\lambda_j \geq 0 \quad j = 1,2,\dots,n; \tag{5}$$

Donde,  $\theta$  representa la medida de productividad o eficiencia de la fábrica de software,  $x_{ij}$  es la  $i$ th entrada ( $x_{1j}$  es el *Esfuerzo Total* y  $x_{2j}$  es el *Costo del Proyecto*) de la fábrica  $j$  (DMU $_j$ ),  $y_j$  y  $\lambda_j$  es la salida (*Puntos de Función*) y el coeficiente o peso asignado de la



fábrica  $j$ , respectivamente;  $x_{i0}$  y  $y_0$  son, respectivamente, la  $i$ th entrada, y la salida de la fábrica bajo evaluación ( $DMU_0$ ).

El modelo busca determinar la eficiencia de la Fábrica de Software a través de la minimización del *Esfuerzo Total* y *Costo del Proyecto* (1) en el DMU virtual. Es por ello que, si  $\theta=1$ , el DMU virtual requiere tantas entradas como lo hace una DMU real bajo evaluación, por lo cual no se evidencia ineficiencia. Si  $\theta<1$ , el DMU virtual requiere menos entradas para obtener el nivel de salida logrado por el DMU real bajo evaluación, es decir, es ineficiente. Se considera restricciones que buscan encontrar la combinación de las entradas y los pesos ponderados ( $\lambda$ ) que determinan la fábrica de software más eficiente (2), y que las salidas sean mayores o se mantengan constantes (3). Asimismo, el total de pesos ponderados que afectan a las entradas y salidas deberían ser igual a 1 (4), y a su vez cada peso ponderado debe tener un valor positivo (5).

A continuación, en la Tabla 5.2 se muestra información para describir un ejemplo.

Tabla 5.2.

Información acerca de Fábricas de Software

DMU	Sector	Esfuerzo Total (TE)	Costo del Proyecto (PC)	Puntos de Función (FP)
SF001	Banca	54,788.36	\$ 2,161,797.26	1,700.00
SF002	Banca	17,268.00	\$ 663,962.77	594.00
SF003	Gobierno	79,646.99	\$ 3,187,837.24	2,736.00
SF004	Gobierno	62,700.98	\$ 1,779,787.46	1,500.00
SF005	Gobierno	63,595.12	\$ 2,588,293.33	2,183.00
SF006	Gobierno	109,362.63	\$ 4,232,423.03	3,760.00
<b>Total</b>		<b>387,362.08</b>	<b>\$ 14,614,101.08</b>	<b>12,473.00</b>

El modelo DEA de este ejemplo para la DMU *SF001*, es como sigue:

*Minimizar*  $\theta$

$$\lambda_1(54,788.36) + \lambda_2(17,268.00) \dots + \lambda_6(109,362.63) \leq \theta(54,788.36)$$

$$\lambda_1(2,161,797.26) + \lambda_2(663,962.77) \dots + \lambda_6(4,232,423.03) \leq \theta(2,161,797.26)$$

$$\lambda_1(1,700) + \lambda_2(594) \dots \lambda_6(3,760) \geq 1,700$$

$$\lambda_1 + \lambda_2 \dots + \lambda_6 = 1$$

$$\lambda_1, \lambda_2, \dots, \lambda_6, \theta \geq 0$$

**Fase 2.** En esta fase se busca identificar los proyectos de fábricas de software más eficientes, a través de la minimización del *Esfuerzo Total* y *Costo del Proyecto* consumidos en un período determinado por los proyectos, pero se debe mantener la cantidad de *Puntos de Función*. Este modelo asume 2 entradas, 1 salida, y  $m$  DMU's, tal como se muestra a continuación:

$$\text{Minimizar } \theta \tag{6}$$

s.t.

$$\sum_{j=1}^{160} \lambda_j \cdot x_{ij} \leq \theta x_{i0} \quad i = 1, 2; \tag{7}$$

$$\sum_{j=1}^{160} \lambda_j \cdot y_j \geq y_0; \tag{8}$$

$$\sum_{j=1}^{160} \lambda_j = 1; \tag{9}$$

$$\lambda_j \geq 0 \quad j = 1, 2, \dots, m; \tag{10}$$

Donde,  $\theta$  representa la medida de productividad o eficiencia de los proyectos de fábrica de software,  $x_{ij}$  es la  $i$ th entrada ( $x_{1j}$  es el *Esfuerzo Total* y  $x_{2j}$  es el *Costo del Proyecto*) del proyecto  $j$  (DMU $_j$ ),  $y_j$  y  $\lambda_j$  son la salida (*Puntos de Función*) y el coeficiente o peso asignado del proyecto  $j$ , respectivamente;  $x_{i0}$  y  $y_0$  son, respectivamente, la  $i$ th entrada, y la salida del proyecto bajo evaluación (DMU $_0$ ).

## 5.5. Fuente de Datos

### 5.5.1. Recopilación de Datos

Para el presente estudio, se ha considerado la información obtenida de los diferentes proyectos de la Fábrica de Software de Palo Alto II (ex GMD S.A.), empresa de Outsourcing de Procesos de Negocios, Tecnología de la Información (TI) y Transformación Digital con mayor confiabilidad y 33 años de experiencia en el Perú, cuenta con un staff de 3000

profesionales e infraestructura adecuada que lo convierte en la Fábrica de Software más grande en el país. Además, cuenta con certificaciones de calidad como ISO 9001, ISO 27001, OHSAS 18001, ISO 20000, ISO 22301, NTP 392-030 y metodologías CMMI-5, ITIL y PMI (GMD, 2018), y, de acuerdo a la consultora internacional IDC (IDC, 2017), es la empresa Líder de Outsourcing en el Perú.

Esta empresa se enfoca en desarrollar soluciones de software para el sector público y privado bajo el modelo de fábrica de software, así, implanta este modelo de manera local en cada cliente, y cuenta para ello con su propio personal, pero sigue los lineamientos de la fábrica de software matriz, habiéndose seleccionado para el presente estudio solo 6 fábricas de software de las 18 fábricas existentes, por ser estas las que cumplen los criterios de realizar actividades de la componente Producción de Software. Asimismo, se ha considerado como parte de la muestra 160 proyectos que están distribuidos en las 6 fábricas de software, los cuales han sido desarrollados en el período de noviembre 2015 a octubre 2017, y que han tenido una duración aproximada de 5 a 7 meses. Inicialmente, se recolectó la información de 200 proyectos, sin embargo, debido a que se presentaban inconsistencias en la data, se procedió a realizar una depuración, así, se obtuvo finalmente solo 160 proyectos que han sido utilizados para el análisis respectivo. La información obtenida de los proyectos está referida al tamaño del proyecto, tipo de sector, esfuerzo y costo del proyecto.

### 5.5.2. Análisis Descriptivo de la Muestra

**Fase 1.** Se ha considerado información de las fábricas de software del sector Banca y Gobierno. La información se encuentra disponible en: <https://tinyurl.com/ycfyp5po>

Tabla 5.3.

Fábricas de software considerados en la muestra

DMU	Sector	Número de Proyectos	Esfuerzo Total (TE)	Costo del Proyecto (PC)	Puntos de Función (FP)
SF001	Banca	23	54,788.36	\$ 2,161,797.26	1,700.00
SF002	Banca	7	17,268.00	\$ 663,962.77	594.00
SF003	Gobierno	33	79,646.99	\$ 3,187,837.24	2,736.00
SF004	Gobierno	22	62,700.98	\$ 1,779,787.46	1,500.00

<b>DMU</b>	<b>Sector</b>	<b>Número de Proyectos</b>	<b>Esfuerzo Total (TE)</b>	<b>Costo del Proyecto (PC)</b>	<b>Puntos de Función (FP)</b>
<i>SF005</i>	Gobierno	21	63,595.12	\$ 2,588,293.33	2,183.00
<i>SF006</i>	Gobierno	54	109,362.63	\$ 4,232,423.03	3,760.00
<b>Total</b>		<b>160</b>	<b>387,362.08</b>	<b>\$ 14,614,101.08</b>	<b>12,473.00</b>

Las estadísticas descriptivas para entradas y salidas de las fábricas de software se muestran en la Tabla 5.4.

Tabla 5.4.

Estadísticas descriptivas para variables de entrada y salida – Fábricas de Software

<b>Estadísticas</b>	<b>Entradas</b>		<b>Salidas</b>
	<b>Esfuerzo Total (TE)</b>	<b>Costo del Proyecto (PC)</b>	<b>Puntos de Función (FP)</b>
<i>Media</i>	64,560.35	\$ 2,435,683.51	2,078.83
<i>Mediana</i>	63,148.05	\$ 2,375,045.29	1,941.50
<i>Desviación Estándar</i>	30,236.73	\$ 1,221,597.77	1,091.09
<i>Mínimo</i>	17,268.00	\$ 663,962.77	594.00
<i>Máximo</i>	109,362.63	\$ 4,232,423.03	3,760.00

A partir de la información de las fábricas de software, se puede notar que en la Figura 5.2, si solo consideramos la cantidad de FP producidos, las fábricas SF006 y SF003 son los que presentan mayor productividad, sin embargo, son también las que presentan esfuerzos similares a las fábricas SF004 y SF005, que su vez presentan menores costos, a fin de determinar si dichos productos han sido implementados con eficiencia. Entonces un análisis de la productividad considerando solo los FP no es suficiente.



Figura 5.2. Fábricas de software considerados en la muestra

Asimismo, podemos notar, en la Figura 5.3, que si consideráramos la técnica estadística de regresión lineal enfocada en promedios, podríamos asumir que la fábrica SF004 es la más productiva, ya que están sobre la línea de tendencia, y las demás (SF001, SF002, SF003, SF005, SF006) deberían mejorar su productividad, por encontrarse por debajo del promedio. Entonces un análisis de la productividad, considerando solo los promedios de productividad, no es suficiente.

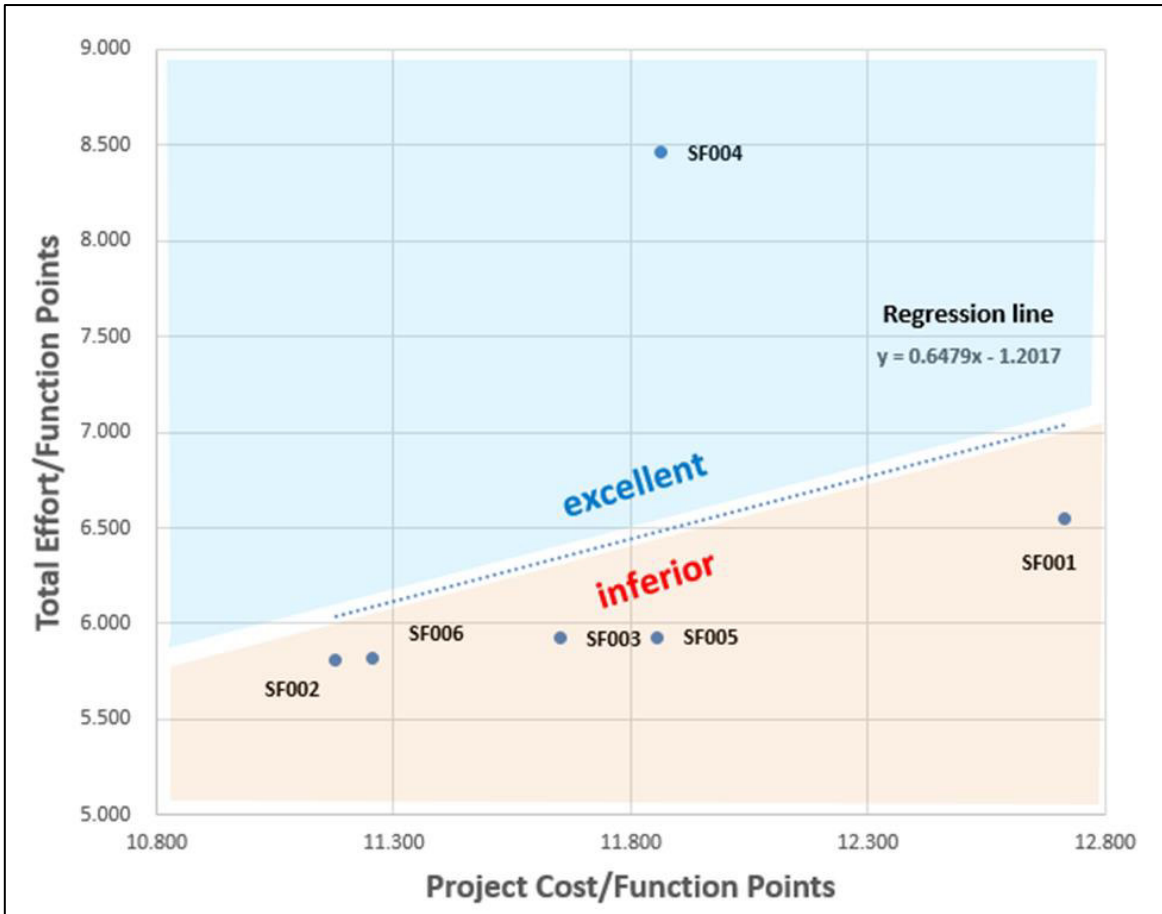


Figura 5.3. Línea de regresión estadística para fábricas de software

**Fase 2.** Considera 160 proyectos desarrollados por las 6 fábricas seleccionadas de la Fase 1.

En la Tabla 5.5 se muestra 10 de los 160 proyectos de fábrica de software que se han utilizado para el análisis. La información se encuentra disponible en la ruta:

<https://tinyurl.com/ycfyp5po>

Tabla 5.5.

Proyectos de fábricas de software considerados en la muestra

<b>DMU</b>	<b>Esfuerzo Total (TE)</b>	<b>Costo del Proyecto (PC)</b>	<b>Puntos de Función (FP)</b>
<i>SP001</i>	1,427.84	\$ 45,267.01	49.00
<i>SP002</i>	1,279.90	\$ 41,195.87	28.00
<i>SP003</i>	1,063.74	\$ 41,162.56	25.00
<i>SP004</i>	1,043.02	\$ 40,567.90	25.00
<i>SP005</i>	1,358.80	\$ 52,708.51	36.00
<i>SP006</i>	1,385.02	\$ 52,679.84	10.00
<i>SP007</i>	1,991.80	\$ 77,004.22	69.00
<i>SP008</i>	3,888.00	\$ 178,606.37	120.00
<i>SP009</i>	1,530.50	\$ 69,905.58	50.00
<i>SP010</i>	2,304.00	\$ 81,217.63	62.00

Tabla 5.6.

Estadísticas descriptivas para variables de entrada y salida – Proyectos de Fábricas de Software

<b>Estadísticas</b>	<b>Entradas</b>		<b>Salidas</b>
	<b>Esfuerzo Total (TE)</b>	<b>Costo del Proyecto (PC)</b>	<b>Puntos de Función (FP)</b>
<i>Media</i>	2,421.01	\$ 91,338.13	77.96
<i>Mediana</i>	1,781.07	\$ 73,292.85	61.00
<i>Desviación</i>	1,672.68	\$ 54,284.23	47.47
<i>Estándar</i>			
<i>Mínimo</i>	1,012.00	\$ 34,196.03	10.00
<i>Máximo</i>	13,524.05	\$ 328,033.66	282.00

## 5.6. Resultados Empíricos e Interpretaciones

Para evaluar la eficiencia relativa de las fábricas de software y sus respectivos proyectos, con múltiples entradas (*Esfuerzo Total, Costo del Proyecto*) y una salida (*Puntos de Función*) se ha utilizado el modelo DEA VRS-I. Los resultados han sido computados con DEAP v2.1 software, así, en la Tabla 5.7 se muestra una lista de abreviaturas que se utilizarán en la interpretación de los resultados.

Tabla 5.7.

Tabla de abreviaturas

Acrónimo	Descripción
<i>DEA</i>	Análisis Envoltorio de Datos
<i>CRS</i>	Retornos de Escala Constante
<i>VRS</i>	Retornos de Escala Variable
<i>TE</i>	Eficiencia Técnica
<i>CRSTE</i>	Eficiencia Técnica con Retornos de Escala Constante
<i>VRSTE</i>	Eficiencia Técnica con Retornos de Escala Variable
<i>SE</i>	Escala de Eficiencia
<i>IRS</i>	Retornos de Escala Creciente
<i>DRS</i>	Retornos de Escala Decreciente
<i>VRS<sub>EFF</sub></i>	Proyectos Eficientes VRS

A continuación se describen los resultados obtenidos para las dos fases en evaluación.

### 5.6.1. Fase 1

De las 6 fábricas de software analizadas, hay 2 DMU's eficientes ( $VRSTE=1$ ), lo que satisface la "regla de un tercio", que indica que el tamaño de la muestra es aceptable si el número de DMU's eficientes no es más de un tercio del número total de DMU's en la muestra (Cooper, Seiford & Zhu, 2004), por lo que el tamaño de la muestra es aceptable. La Tabla 5.8 muestra la eficiencia ( $VRSTE$ ) de las fábricas de software para el modelo DEA VRS-I.



Tabla 5.8.

Eficiencia DEA VRS-I – Fábrica de Software

DMU	VRSTE
SF001	0.893
<b>SF002</b>	<b>1.000</b>
SF003	0.977
SF004	0.937
SF005	0.996
<b>SF006</b>	<b>1.000</b>

Un resumen de las eficiencias se muestra en la Figura 5.4

```

EFFICIENCY SUMMARY:

  firm  crste  vrste  scale
SF001  0.893  0.893  1.000  -
SF002  1.000  1.000  1.000  -
SF003  0.977  0.977  1.000  -
SF004  0.933  0.937  0.995  drs
SF005  0.996  0.996  1.000  -
SF006  1.000  1.000  1.000  -

mean  0.966  0.967  0.999

Note: crste = technical efficiency from CRS DEA
      vrste = technical efficiency from VRS DEA
      scale = scale efficiency = crste/vrste

Note also that all subsequent tables refer to VRS results
    
```

Figura 5.4. Resumen de eficiencias

De acuerdo a la Figura 5.4, en promedio, los puntajes de eficiencia de las fábricas de software son:

- 96.6% para CRSTE; en general, las fábricas de software podrías reducir sus entradas en un 3.4% para obtener el mismo número de FP.
- 96.7% para VRSTE; una mejor fábrica de software podría ser capaz de reducir el consumo de entradas en un 3.3%.

- 99.9% para SE; al ajustar la escala, fábricas de software podrían reducir sus entradas en un 0.1%

A continuación, se analiza los resultados de la SF004.

Results for SF004:				
Technical efficiency = 0.937				
Scale efficiency = 0.995 (drs)				
PROJECTION SUMMARY:				
variable	original value	radial movement	slack movement	projected value
output Points	1.500	0.000	0.000	1.500
input Effort	62.700	-3.959	-15.538	43.203
input Cost	1780.000	-112.388	0.000	1667.612
LISTING OF PEERS:				
peer	lambda	weight		
SF006	0.281			
SF002	0.719			

Figura 5.5. Resultados para SF004

En la Figura 5.5, notamos que SF004 tiene una eficiencia técnica de 93.7% y una escala de eficiencia de 99.5%, con una tendencia a retornos de escala decreciente (DRS). Al mejorar la operación de la fábrica de software, se podría reducir el uso de entradas en un 6.3% (100 – 93.7). Al ajustar la fábrica de software a su tamaño óptimo, se podría reducir el uso de entradas en 0.5% (100 – 99.5).

La columna ‘*original value*’ contiene los valores originales de las variables de estudio de la fábrica de software: SF004 produce 1.5 FP con 62.7 horas de esfuerzo y con un costo total de \$ 1,780. Sin embargo, SF004 podría producir el mismo número de FP con menos entradas: 43.20 horas en vez de 62.7; \$ 1667.612 en vez de 1,780 (ver la columna ‘*projected value*’). La reducción en las entradas TE y PC es 6.31% de los valores originales:  $((- 3.959 / 62.7) \times 100)$ , para la entrada TE y  $((- 112.388 / 1780) \times 100)$ , para la entrada PC. Sin embargo, en el caso de la entrada TE, para que llegue a ser eficiente, no solo debe reducirse en 6.31% (menos 3.959 de la columna ‘*radial movement*’), sino también en un 15.538 adicional (columna ‘*slack movement*’), es decir que para que la entrada TE llegue a ser eficiente debe reducirse en un 31.1%. Para mejorar su eficiencia, SF004 tiene que analizar las prácticas de la SF006 y SF002, los cuales son identificados como sus pares. Para ser un par (o un benchmark), una fábrica de software debe tener una eficiencia técnica de 100%. El peso ponderado ( $\lambda$ ) asociado con cada par corresponde a su importancia relativa entre el grupo de pares. Idealmente, SF004 debería analizar las mejores prácticas desde una fábrica de software

compuesta por 28.1% de SF006 y 71.9% de SF002. Como una fábrica de software ‘virtual’ no existe, SF004 debería concentrarse en analizar las mejores prácticas del par asociado con el mayor valor de  $\lambda$ , por lo cual debería compararse con la SF002.

La Tabla 5.9 muestra las fábricas de software de referencia VRSTE para cada fábrica de software considerado en la muestra. Una fábrica de software de referencia siempre es seleccionada desde un conjunto de fábricas establecidas en la frontera eficiente. En el caso del modelo VRSTE, una fábrica de referencia siempre será una fábrica eficiente de similar tamaño a las fábricas que la consideran como referencia (Stensrud & Myrtveit, 2003).

Tabla 5.9.

Eficiencia DEA VRS-I – Fábrica de Software

DMU	VRSTE de Entrada	Orientación	Benchmark ( $\lambda$ óptimo)	
			SF006	SF002
<i>SF001</i>		0.893	0.344	<b>0.656</b>
<b>SF002</b>		<b>1.000</b>		1.000
<i>SF003</i>		0.977	<b>0.656</b>	0.344
<i>SF004</i>		0.937	0.281	<b>0.719</b>
<i>SF005</i>		0.996	<b>0.500</b>	<b>0.500</b>
<b>SF006</b>		<b>1.000</b>	1.000	

La tabla 5.9 muestra, en la primera columna, la lista de todas las fábricas y, en la segunda, la eficiencia de las mismas, la tercera y cuarta columna muestran los pesos asociados a las fábricas de referencia SF006 y SF002, respectivamente. Por ejemplo, si revisamos la fábrica SF004, podemos notar que para mejorar su eficiencia, ella debería acercarse a la frontera eficiente, y el Pareto más cercano es dado por la combinación ponderada (pesos) de SF002 y SF006 (Virtual Software Factory A). También, observamos que la SF002 es más adecuado como fábrica de referencia, porque permite generar mayor productividad con menor esfuerzo y costo, que SF006 (71.9% vs. 28.1%), tal como se puede visualizar en la Figura 5.6, en el cual las fábricas SF002 y SF006 se encuentran en la frontera de eficiencia VRS.

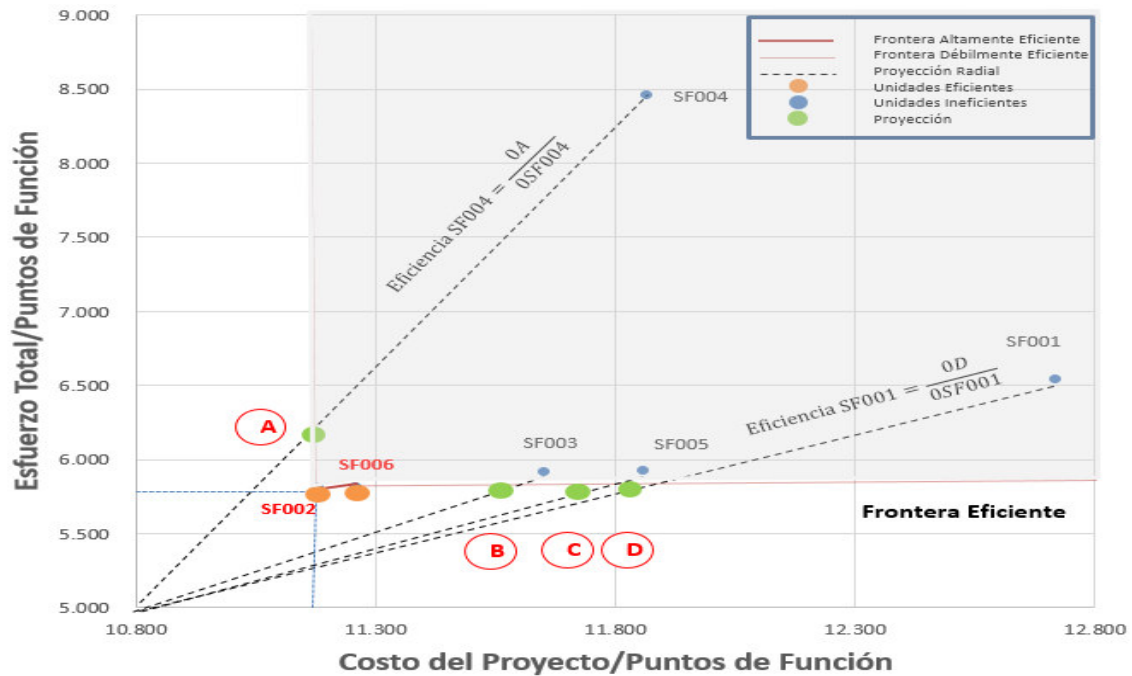


Figura 5.6. Conjunto de referencias VRS para la fábrica de software

### 5.6.2. Fase 2

Se identificaron 10 proyectos eficientes de 160 proyectos evaluados (ver Tabla 5.10).

Tabla 5.10.

Eficiencia DEA VRS-I – Proyectos de Fábrica de Software

DMU	VRSTE	Fábrica de Software (SF)	Frecuencia
SP023	1	SF004	0
SP030	1	SF002	<b>68</b>
SP036	1	SF004	131
SP046	1	SF004	4
SP065	1	SF006	<b>5</b>
SP079	1	SF005	52
SP095	1	SF005	7
SP122	1	SF005	41
SP136	1	SF003	49
SP146	1	SF001	40

Tal como se puede visualizar en la Figura 5.7, los proyectos eficientes que más son tomados en cuenta como proyectos que aplican buenas prácticas y reflejan una mayor productividad son el SP036 (SF004) y SP030 (SF002), notándose que el proyecto más eficiente no está dentro de las fábricas de software más eficientes, por lo cual se podría mencionar que si los proyectos de fábrica de software eficientes replicarán las buenas prácticas o estrategias que realiza el proyecto SP036, se podría tener una mayor productividad. Asimismo, se puede notar que las fábricas de software que tienen la mayor cantidad de proyectos eficientes son la SF005 y SF004.

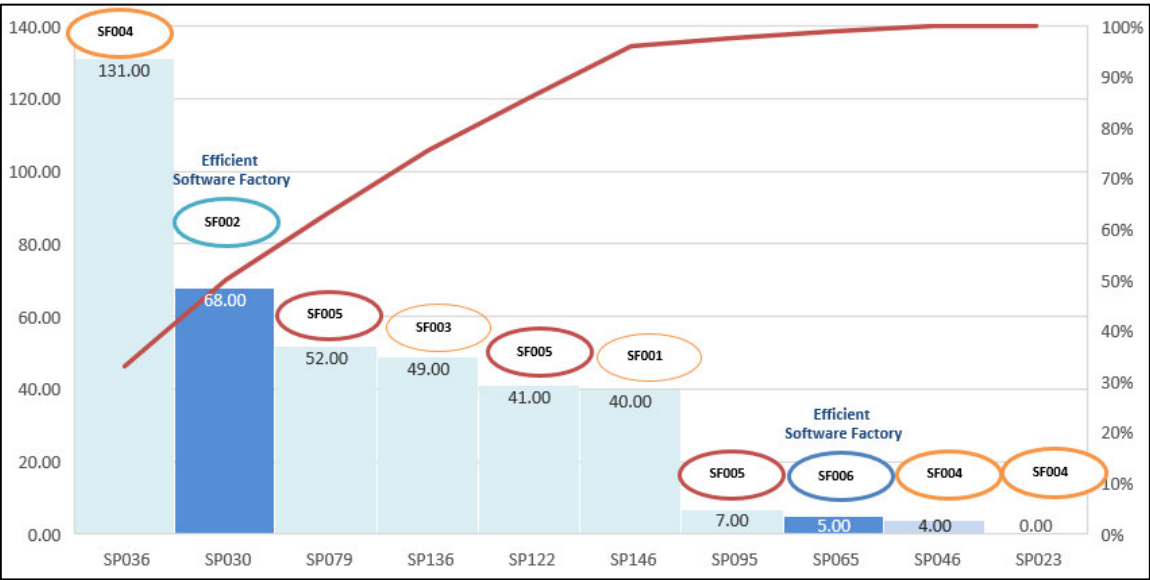


Figura 5.7. Cantidad de veces que un proyecto es referencia de otro proyecto

Las estadísticas son mostradas en Tabla 5.11.

Tabla 5.11.

Eficiencia promedio DEA VRS

DMU	N	CRSTE	VRSTE	SE	VRS <sub>EFF</sub>
Proyecto Fábrica de Software	160	0.906	0.940	0.963	10

De acuerdo a la Tabla 5.11, en promedio, los puntajes de eficiencia de las fábricas de software son:

- 90.6% para CRSTE; en general, los proyectos de fábrica de software podrían reducir sus entradas en un 9.4% para obtener el mismo número de FP.

- 94.0% para VRSTE; un mejor proyecto de fábrica de software podría ser capaz de reducir el consumo de entradas en un 6%.
- 96.3% para SE; al ajustar la escala, proyectos de fábrica de software podrían reducir sus entradas en un 3.7%

En la Figura 5.8 se puede identificar qué factores han influenciado para que los proyectos de fábrica de software sean eficientes.

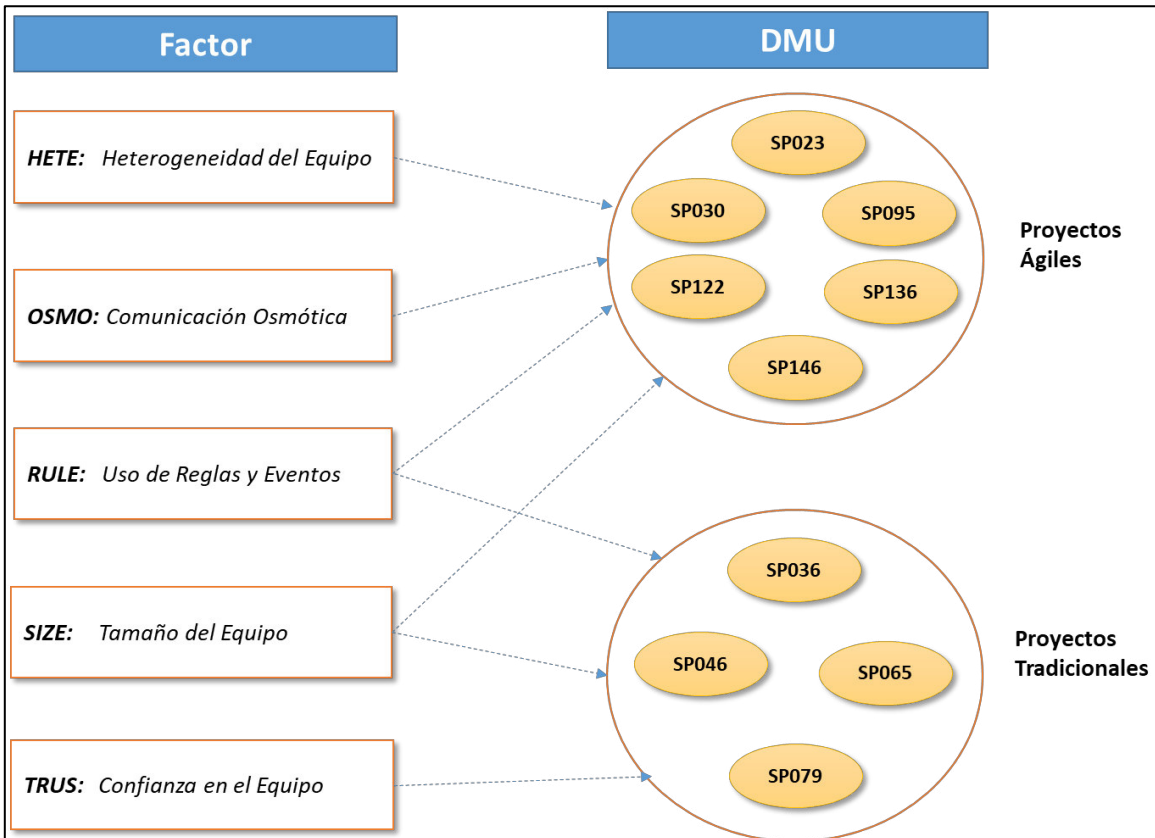


Figura 5.8. Factores y su influencia en los proyectos de fábrica de software eficientes

En la Tabla 5.12 se detallan algunas características de los proyectos de fábrica de software eficientes y cómo los factores identificados en el capítulo anterior han influenciado para el logro del objetivo.

Tabla 5.12.

Caracterización de los proyectos de fábrica de software

<b>Factor</b>	<b>Tipo de Metodología</b>	<b>Justificación</b>
<i>Heterogeneidad del Equipo</i>	Ágil	En estos proyectos se ha identificado la participación de múltiples roles con diferentes habilidades y capacidades para el desarrollo de un producto, pudiéndose resaltar que al ser proyectos de innovación donde los requerimientos son cambiantes implica la necesidad de incorporar nuevas perspectivas que ayudan a los miembros del equipo a superar problemas desde diferentes ángulos.
<i>Comunicación Osmótica</i>	Ágil	Los miembros de los proyectos se encuentran localizados en ambientes de trabajo llamados “Cuartos de Guerra”, lo cual permite que la información fluya en el fondo auditivo de los miembros del equipo, de tal forma que ellos captan la información relevante, facilitando el aprendizaje y reduciendo los tiempos de curva de aprendizaje, ya que todo el equipo conoce las actividades que vienen realizando cada uno de sus miembros.
<i>Uso de Reglas y Eventos</i>	Ágil Tradicional	En los proyectos eficientes, se ha identificado que en los equipos de trabajo se han establecido reglas para facilitar la colaboración. Por ejemplo, el uso del Log de

<b>Factor</b>	<b>Tipo de Metodología</b>	<b>Justificación</b>
		<p>Impedimentos para reportar cualquier problema que esté impidiendo la culminación de actividades comprometidas, haciendo que ésta sea visible a todo el equipo.</p> <p>También se ha identificado la implementación de timebox en los eventos, lo cual evita que el equipo se distraiga en actividades accesorias que impidan el cumplimiento del objetivo.</p>
<i>Tamaño del Equipo</i>	<p>Ágil</p> <p>Tradicional</p>	<p>Los proyectos utilizan equipos de trabajo pequeños, oscilando la cantidad de participantes entre 6 a 10 personas, lo cual ha facilitado los niveles de comunicación.</p>
<i>Confianza en el Equipo</i>	Ágil	<p>En los proyectos se han implementado prácticas ágiles, tal como el proceso de estimación, la cual es realizada por los miembros del equipo que van a ejecutar las tareas, y éstas son propuestas en los planes de trabajo, lo que ha permitido que el equipo se sienta comprometido y genere niveles de confianza entre los diferentes niveles de la organización.</p>



De los 160 proyectos analizados, hay 10 DMU's eficientes ( $VRTSE=1$ ), lo que corresponde aproximadamente al 6.3% de las DMU's. Además, la "regla de un tercio" indica que el tamaño de la muestra es aceptable, si el número de DMU's eficientes no es más de un tercio del número total de DMU's en la muestra (Cooper *et al.*, 2004). Como el número de DMU's es 160 y la cantidad de proyectos eficientes es 10, entonces se satisface la regla. Una DMU con una eficiencia menor a 1 es relativamente ineficiente con respecto a sus puntos de referencia.

Tabla 5.13.

Conjunto de referencias VRS para los proyectos de fábrica de software

DMU	Eficiencia	Benchmark ( $\lambda$ óptimo)										
		VRS-I	SP023	SP030	SP036	SP046	SP065	SP079	SP095	SP122	SP136	SP146
SP001	0.98687		0.029	0.123			0.848					
SP002	0.98617						0.274					0.726
SP003	0.99381					0.396		0.604				
SP004	0.98791					0.473		0.527				
SP005	0.99168									0.545		0.455
SP006	0.99404					0.139		0.861				
SP007	0.99997		0.050	0.743			0.207					
SP008	0.99455			0.757					0.243			
SP009	0.9952			0.438							0.563	
SP010	0.99207		0.066	0.389			0.545					
...												
SP151	0.97853										1.000	
SP152	0.99198			0.879					0.121			
SP153	0.99173			0.771					0.229			
SP154	0.99791			0.219						0.103		0.678
SP155	0.99504		0.142	0.757			0.101					
SP156	0.9899		0.005	0.562			0.433					
SP157	0.99979		0.003	0.389			0.608					
SP158	0.99489		0.025	0.485			0.490					
SP159	0.98488									0.220		0.780
SP160	.99771		0.532	0.212					0.256			

La Tabla 5.13 muestra los proyectos de referencia VRS para cada proyecto de fábrica de software considerado en la muestra. Un proyecto de referencia siempre es seleccionado desde un conjunto de proyectos establecidos en la frontera eficiente. En el caso del modelo DEA VRS, un proyecto de referencia siempre será un proyecto eficiente de similar tamaño a los proyectos que la consideran como referencia (Stensrud & Myrtveit, 2003). La tabla muestra, en la primera columna, la lista de todos los proyectos, y en la segunda, los proyectos de referencia con sus respectivos  $\lambda$ . Por ejemplo, si revisamos el proyecto SP001, podemos notar que tiene tres proyectos en su conjunto de referencia: proyectos SP030, SP036, y SP079. También, observamos que el proyecto SP079 es más importante referencia que los proyectos SP036 y SP030 (84.8% vs. 12.3% y 2.9%, respectivamente).

## **5.7. Procedimiento de Medición**

A continuación se detallará el procedimiento a seguir para la aplicación del modelo de medición de la productividad basado en DEA.

### **5.7.1. Alcance**

El alcance del procedimiento está orientado a:

- Fábricas de software, y;
- Proyectos implementados en las fábricas de software.

### **5.7.2. Responsabilidades**

En la Tabla 5.14, se describen las responsabilidades de los diferentes roles que participan en el proceso de medición de la productividad:

Tabla 5.14.

Responsabilidades de los participantes en el procedimiento

<b>Rol</b>	<b>Componente</b>	<b>Responsabilidades</b>
<i>Gestor del Proyecto</i>	Gestión del Proyecto	<ul style="list-style-type: none"> <li>• Responsable de la Gestión del Proyecto</li> <li>• Asignación de los recursos necesarios para la ejecución del proyecto.</li> <li>• Seguimiento de las actividades ejecutadas por el equipo del proyecto.</li> </ul>
<i>Equipo de Fábrica</i>	Producción de Software	<ul style="list-style-type: none"> <li>• Ejecución de las actividades del proyecto.</li> <li>• Reporte de actividades diarias.</li> </ul>
<i>Equipo de Arquitectura</i>	Soporte	<ul style="list-style-type: none"> <li>• Validación de la información reportada por los diferentes equipos de trabajos que vienen ejecutando proyectos.</li> </ul>
<i>Gestor de Indicadores</i>	Soporte	<ul style="list-style-type: none"> <li>• Integración de la información de los proyectos de fábrica de software.</li> <li>• Aplicación del modelo de medición de la productividad.</li> <li>• Consolidación y difusión de los resultados de la aplicación del modelo.</li> <li>• Implementación de estrategias de mejora continua.</li> <li>• Mejoras y ajustes en el modelo DEA</li> </ul>

### 5.7.3. Procedimiento

#### 5.7.3.1. Entradas y Salidas

##### Entradas

Insumo	Origen
<i>Esfuerzo</i>	Equipo de Fábrica
<i>Costos del Proyecto</i>	Gestor del Proyecto

##### Salidas

Servicio o Producto	Destino
<i>Puntos de Función</i>	Cliente

#### 5.7.4. Proceso

En la Figura 5.8. se visualizan las actividades para la medición de la productividad.

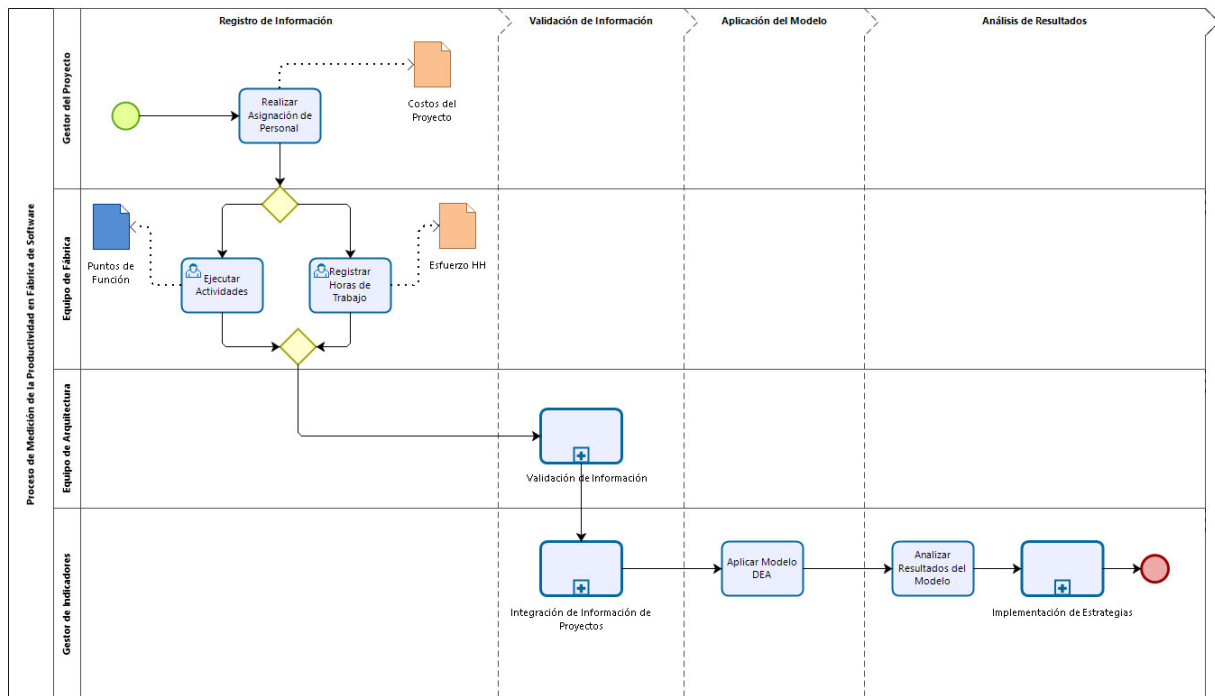


Figura 5.9. Proceso para la medición de la productividad.

En la Tabla 5.15 se detallan las actividades a realizarse en el proceso.

Tabla 5.15.

Actividades del proceso de medición de la productividad

<b>Rol</b>	<b>Actividad</b>
<i>Gestor del Proyecto</i>	<ul style="list-style-type: none"> <li>• Realizar la asignación del personal que será parte del Equipo de Fábrica.</li> </ul>
<i>Equipo de Fábrica</i>	<ul style="list-style-type: none"> <li>• Ejecutar actividades del proyecto.</li> <li>• Registrar horas de trabajo que se han utilizado para la ejecución de las tareas y/o actividades</li> </ul>
<i>Equipo de Arquitectura</i>	<ul style="list-style-type: none"> <li>• Validar información reportada por los diferentes equipos de trabajo. En caso exista alguna incidencia u observación, ésta será reportada al Gestor de Proyecto para su respectiva validación.</li> </ul>
<i>Gestor de Indicadores</i>	<ul style="list-style-type: none"> <li>• Integrar la información de los diferentes proyectos para su respectiva evaluación.</li> <li>• Depurar información inconsistente del proyecto y reportar al Gestor de Proyecto respectivo.</li> <li>• Introducir valores (entradas y salidas) en las herramientas del Modelo DEA.</li> <li>• Evaluar los resultados obtenidos con los diferentes Gestores de Proyectos, a fin de definir las buenas prácticas utilizadas en los proyectos que han obtenido mejores resultados.</li> <li>• Definir estrategias para la mejora continua.</li> <li>• Difundir información a los proyectos.</li> </ul>

### **5.7.5. Cambios**

Para las modificaciones y revisiones que se realicen al presente documento, se deberá tener en cuenta el siguiente formato:

## Revisión No.

Cláusula (*)	Cambio
(N° de cláusula)	(Breve descripción del cambio).
	(Agregue las filas que sean necesarias. Eliminar la tabla cuando es Revisión 0.).

(\*) Cuando en una misma revisión se realiza un cambio que afecte varias cláusulas, ésta se menciona una sola vez, anotando las cláusulas afectadas. Sólo se registran los últimos tres cambios.

### 5.7.6. Referencias

Se deberá tener en cuenta los instructivos de trabajo y procedimientos de cada organización.

## 5.8. Conclusiones

Este trabajo presenta un modelo para la medición de la productividad basado en DEA, así, se toma en consideración información de fábricas de software implementadas y ejecutadas en el Perú. Para el estudio, se ha utilizado el modelo DEA con Retornos de Escala Variable y Orientación de Entrada (DEA VRS-I), ya que se busca minimizar las entradas (*Esfuerzo Total* y *Costo del Proyecto*) y mantener la salida actual (*Puntos de Función*) de las fábricas de software y sus respectivos proyectos.

Para ello se ha considerado el análisis en dos fases, la fase 1 considera la evaluación de las fábricas de software considerando el *Esfuerzo Total* y *Costo del Proyecto* como entradas y *Puntos de Función* como salida; y la fase 2 evalúa la productividad de los proyectos que realiza la fábrica de software.

Los resultados muestran 6 fábricas de software y 160 proyectos implementados por las respectivas fábricas, el modelo identifica la fábrica de software más eficiente, las mismas que sirven como modelo de fábricas de su mismo tamaño o complejidad, con lo cual facilita la tarea de la gestión de proyectos, ya que la orientación estará a determinar qué prácticas

eficientes se vienen realizando en esta fábrica que le ha conllevado a convertirse en el modelo a seguir. El modelo también permite identificar en la fase 2 los proyectos más eficientes.

Asimismo, se ha podido comprobar que es necesario la medición no solo a nivel de fábrica de software, sino a nivel de sus proyectos, ya que, como muestran los resultados, no necesariamente una fábrica de software modelo tiene los proyectos más eficientes. Realizar un análisis en dos fases permite descubrir qué factores se encuentran inmersos en estos proyectos, que podrían ser implementados en los proyectos de la fábrica más eficiente, lo cual conllevaría a mejorar su eficiencia. Tal como se ha podido comprobar con los resultados, las fábricas de software más eficientes podrían mejorar su productividad, si tomaran en cuenta las buenas prácticas que se vienen implementando en los proyectos.

Este modelo permite quitar el sesgo de medir la productividad en comparación a otras técnicas estadísticas, tal como se puede notar en el análisis de regresión lineal, donde muchas veces podemos considerar proyectos eficientes cuando no necesariamente lo son.

#### **5.8.1. Limitaciones del Estudio**

Las pruebas realizadas han sido limitadas a fábricas de software en el Perú, y sus respectivos proyectos solo se ha orientado a las unidades de trabajo Análisis & Diseño, Programación y Pruebas, las cuales pertenecen al componente de Producción de Software.

#### **5.8.2. Trabajos Futuros**

Futuros estudios podrían considerar la variante de DEA (DEA Network) para poder medir la eficiencia en las unidades de trabajo del componente de Producción de Software, de tal manera que se pueda tener un estudio integral de la fábrica de software, y poder establecer oportunidades de mejoras internas que podrían generarnos mejores resultados a nivel de fábrica de software.

También, es necesario considerar el análisis de las fábricas de software en las diferentes variantes de DEA, a fin de poder conocer el comportamiento de la productividad, ya sea tomando en cuenta la maximización de los outputs o la minimización de los inputs.

Asimismo, se debería considerar el diseño de estrategias que permitan que fábricas de software o proyectos puedan lograr una mejor productividad, tomando como base las fábricas

o proyectos que se encuentren en la frontera eficiente, y no necesariamente podría ocurrir que esta entidad eficiente y homogénea con la evaluada deba tener su reflejo en la realidad, ya que puede darse el hecho de que la entidad eficiente no sea real, sino una combinación lineal de otras existentes (Fábrica de Software Virtual).



## Capítulo 6: Conclusiones y trabajos futuros

### 6.1. Conclusiones

En la presente tesis se ha realizado una revisión sistemática de la literatura sobre factores que afectan la productividad y modelos para medirla. Asimismo, se han introducido tres nuevos factores que influyen en la productividad de una fábrica de software y un modelo basado en DEA Orientado a Entrada con Retorno de Escala Variable para medir la productividad en fábricas de software.

Se ha identificado 74 factores, los cuales han sido agrupados de acuerdo a las categorías propuestas por Wagner *et al.* (2008), ya que contempla categorías que en muchos casos no han sido cubiertos por Trendowicz *et al.* (2009), tales como *Entorno de Desarrollo, Cultura Corporativa, Cultura del Equipo y Entorno*, sin embargo, estas categorías son necesarias, ya que agrupan a factores que son transversales a toda la organización y no son exclusivas del producto o proyecto.

Se ha identificado diez modelos de medición de la productividad de software que están basados en Análisis Envoltorio de Datos (Asmild *et al.*, 2006; Cao *et al.*, 2012; Pai *et al.*, 2015), CMMI & Six Sigma (Moreira *et al.*, 2010), Lógica Difusa (López, *et al.*, 2010), Modelo de Ecuaciones Estructurales (Yilmaz & O'Connor, 2011), Reutilización de Software (Nwelih & Amadin, 2008), Factor de Productividad Total (Machek, *et al.*, 2012), ISO 9126-4 IEEE Std.1045 (Cheikhi, *et al.*, 2012), y otras técnicas estadísticas (Unluturk & Kurtel, 2015). Si bien es cierto se definen como modelos de medición de la productividad, muchos de ellos plantean una función, sin embargo, no derivan a una fórmula (Yilmaz & O'Connor, 2011; Machek, *et al.*, 2012; Cheikhi, *et al.*, 2012); solo se mencionan diversos factores a tener en cuenta en la medición.

Los modelos solo han sido diseñados para realizar una medición de la productividad interna, pero a pesar de ello son incompletos, ya que han sido orientados en su mayoría para la unidad de trabajo de Programación (Asmild, *et al.*, 2006; Nwelih & Amadin, 2008; Moreira, *et al.*, 2010; Yilmaz & O'Connor, 2011; Machek, *et al.*, 2012; Cheikhi, *et al.*, 2012) y muy pocos para Pruebas y Análisis & Diseño (Moreira, *et al.*, 2010; Pai, *et al.*, 2015; Unluturk & Kurtel, 2015; Cao, *et al.*, 2012), lo cual genera un sesgo en la medición, ya que no permitiría

establecer la productividad en los otros componentes de la fábrica de software. No se ha encontrado modelos que midan la productividad en toda la fábrica de software. Se han identificado 74 factores que influyen en la productividad, entre ellos los más utilizados son los siguientes: *Tamaño de Equipo*, *Reutilización* y *Complejidad*, sin embargo, existen muchos factores que no han sido tomados en cuenta en los modelos, tales como *Números totales de lanzamientos*, *Problemas*, *Riesgos*, *Solicitud de cambio*, *Ocurrencia de espera*, *Duración de espera*, *Motivo de espera*. Si tomamos en cuenta los factores que existen en la literatura y los modelos que se han elaborado a la fecha, se puede notar que no existe una homogeneización de los factores a tomar en cuenta en un modelo, y en muchos casos no se ha tomado en cuenta algunos de los factores, lo cual afecta la precisión del modelo, ya que se considera que cuantos más factores cubra el modelo, se podrá establecer más aspectos de la realidad.

En esta tesis se ha introducido tres nuevos factores que influyen en la productividad de software sustentadas en teorías y se han estudiado la influencia de los factores Tamaño del equipo (SIZE) y Confianza en el equipo (TRUS) en las diferentes unidades de trabajo de la componente de la fábrica de software. Los resultados obtenidos de la encuesta administrada a 150 profesionales involucrados en actividades de desarrollo de software en entorno de fábrica en el Perú corroboraron la influencia significativa positiva de los factores propuestos Uso de reglas y eventos (RULE) y Confianza en el equipo (TRUS), y la influencia significativa negativa del factor Heterogeneidad del equipo (HETE) en la unidad de trabajo de Análisis & Diseño. Por otro lado, en la unidad de Programación, la Heterogeneidad del equipo (HETE), Tipo de comunicación osmótica en equipo (OSMO), Tamaño del equipo (SIZE), y Confianza en el equipo (TRUS) tienen una influencia significativa positiva. También, en la unidad de trabajo de Pruebas, se ha demostrado que Confianza en el equipo (TRUS) tiene una influencia significativa positiva. Asimismo, se puede visualizar que el factor Confianza en el equipo (TRUS) afecta a las tres unidades de producción. Además, se reafirma que el factor Tamaño del equipo (SIZE) tiene una influencia significativa positiva en la unidad de trabajo de Programación, sin embargo, no tiene este mismo efecto en las demás unidades.

La tesis también propone un modelo para la medición de la productividad basado en Modelo DEA Orientado a Entrada y con Retornos de Escala Variable, que contempla dos fases: la

fase 1 considera la evaluación de las fábricas de software considerando el Esfuerzo Total y Costo del Proyecto como entradas y Puntos de Función como salida; y la fase 2 evalúa la productividad de los proyectos que realiza la fábrica de software. Los resultados sobre 6 fábricas de software y 160 proyectos implementados por las respectivas fábricas muestran que el modelo identifica la fábrica de software más eficiente, la misma que sirve como modelo de fábricas de su mismo tamaño o complejidad, con lo cual facilita la tarea de la gestión de proyectos, ya que la orientación estará dada a determinar qué prácticas eficientes se vienen realizando en esta fábrica que le ha conllevado a convertirse en un modelo a seguir. El modelo también permite identificar en la fase 2 los proyectos más eficientes. Asimismo, se ha podido comprobar que es necesario la medición no solo a nivel de fábrica de software, sino a nivel de sus proyectos, ya que como muestran los resultados, no necesariamente una fábrica de software modelo tiene los proyectos más eficientes. Realizar un análisis en dos fases permitirá descubrir qué factores se encuentran inmersos en estos proyectos, que podrían ser implementados en los proyectos de la fábrica más eficiente, lo cual conllevaría a mejorar su eficiencia. Las fábricas de software (donde se incluye las más eficientes) podrían mejorar su productividad si tomaran en cuenta las buenas prácticas que se vienen implementando en los proyectos.

Este modelo permite quitar el sesgo de medir la productividad en comparación a otras técnicas estadísticas, tal como se puede notar en el análisis de regresión lineal, donde muchas veces podemos considerar proyectos eficientes cuando no necesariamente lo son.

## **6.2. Limitaciones del Trabajo**

La revisión de la literatura se ha limitado a artículos de Journal en Scimago Journal Rank y en inglés.

Los factores propuestos han sido bien sustentados basados en teorías de la Perspectiva de la Acción del Lenguaje y la Teoría de la Memoria Transactiva, sin embargo, su validación experimental se ha realizado mediante técnicas de encuesta, aplicadas en fábricas de software implementadas y ejecutadas en el Perú, debido a que los dataset disponibles no contemplan la información de factores y su aplicación en fábricas de software, solo limitándose a la unidad de trabajo de Programación, por lo que se requiere hacer encuestas en otros países para verificar la influencia de los factores propuestos.

El estudio ha sido orientado a las unidades de trabajo de Análisis & Diseño, Programación y Pruebas, las cuales pertenecen al componente de Producción de Software.

### **6.3. Trabajos Futuros**

Futuros estudios podrían considerar analizar los factores que afectan a la unidad de trabajo de Modelamiento de Negocio de la componente Producción de Software, así como a las componentes de Gestión de Proyecto y Soporte, de tal manera que se pueda tener un estudio integral de la fábrica de software y en base a ello determinar su impacto en el modelo DEA. También, un aspecto a tener en cuenta en futuros trabajos sería realizar un análisis del factor Tipo de comunicación “cara a cara” en equipo (FACE) para determinar su impacto en entornos de trabajo virtuales y presenciales. También, es recomendable que se cuente con una base de datos de la información que se genera en los proyectos de fábricas de software, de tal manera que facilite los estudios de investigación en esta área, para establecer estrategias que permitan potenciar los factores positivos y reducir los factores negativos, a fin de mitigar su impacto en la productividad.

Asimismo, se podrían considerar el análisis de las fábricas de software en las diferentes variantes de DEA, así, puede ser una de ellas la variante DEA Network para poder medir la eficiencia en las unidades de trabajo del componente de Producción de Software, de tal manera que se pueda tener un estudio integral de la fábrica de software, y de esta manera conocer el comportamiento de la productividad, ya sea tomando en cuenta la maximización de salidas o la minimización de las entradas, con la finalidad de establecer oportunidades de mejoras internas que podrían generarnos mejores resultados a nivel de fábrica de software.

En la presente tesis se ha establecido un modelo e implícitamente un procedimiento de cómo se puede medir la productividad, tal como se puede ver en el Capítulo 5, sin embargo en trabajos futuros se debe establecer un procedimiento explícito que permita a las empresas adoptar el modelo y tener una guía de implementación.

## Referencias Bibliográficas

Abdel-Hamid, T. K. (1989). The dynamics of software project staffing: a system dynamics based simulation approach. *IEEE Transactions on Software Engineering* 15, 109–119.

Abdel-Hamid, T. K., & Madnick, S.E. (1991). *Software Project Dynamics: An Integrated Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Alba, E., & Chicano, J. (2005). Management of software projects with gas. In: *Proceedings of the 6th Metaheuristics International Conference (MIC'05)*. Elsevier Science Inc., Vienna, Austria, pp. 13–18.

Alba, E., & Chicano, J. (2007). Software project management with GAs. *Information Sciences* 177, 2380–2401.

Ancona, D. G., & Caldwell, D. F. (1992), “Demography and design: predictors of new product team performance”, *Organization Science*, Vol. 3 No. 3, pp. 321-41.

Aoyama, M. (1996). Beyond software factories: concurrent-development process and an evolution of software process technology in Japan. *Information and Software Technology*, 38(3), 133-143.

Arcudia-Abad, E., Solís-Carcaño, R. G., & Cuesta-Santos, A. R. (2007). Propuesta tecnológica para incrementar la productividad en la construcción masiva de vivienda. *Ingeniería, Investigación y Tecnología*, 8(2), 11.

Asmild, M., Paradi, J. C., & Kulkarni, J. C. (2006). Using data envelopment analysis in software development productivity measurement. *Software Process Improvement and Practice*, 11(6), 561-572 13, DOI: 10.1002/spip.298.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiency in data envelopment analysis. *Manag Sci* 30(9):1078–1092.

Banker, R. D., & Kemerer, C. F. (1989). Scale economies in new software development. *IEEE Trans Softw Eng* 15(10):1199–1205.

Banker, R. D., Datar, S.M., & Kemerer, C.F. (1991). A model to evaluate variables impacting the productivity of software maintenance projects. *Manag Sci* 37(1):1–18.

- Banker, R. D., Chang, H., & Kemerer, C.F. (1994). Evidence on economies of scale in software development. *Inf Softw Technol* 36(5):275–282.
- Banker, R. D., & Kauffman, R. J. (1991). Reuse and productivity in integrated computer-aided software engineering: An empirical study. *MIS Quarterly*, 14(3), 374-401.
- Basili, V. R. (1989). The Experience Factory: Packaging Software Experience. Paper presented at the *14th Annual Software Engineering Workshop NASA Goddard Space Flight Center*.
- Basili, V. R. (1993). The Experience Factory and its Relationship to Other Improvement Paradigms. Paper presented at the *4th European Software Engineering Conference – ESEC '93*.
- Basili, V. R., Briand, L. C., & Melo, W. L. (1996). How reuse influences productivity in object-oriented systems. *Communications of the ACM*, 39(10), 104-116.
- Basili, V. R., Caldiera, G., & Cantone, G. (1992). A reference architecture for the component factory. *ACM Transactions on Software Engineering and Methodology*, 1(1), 53-80.
- Behrens, C. A. (1983). Measuring the Productivity of Computer Systems Development Activities with Function Points, *IEEE Trans. on Softw. Eng.*, vol. 9, pp. 649-658.
- BID (2010). *Banco Interamericano de Desarrollo: La era de la productividad: como transformar las economías desde sus cimientos*.
- Blackburn, J. D., Scudder, G. D., & Van Wassenhove, L. N. (1996). Improving speed and productivity of software development: A global survey of software developers. *IEEE Transactions on Software Engineering*, 22(12), 875-885.
- Bonilla, E. (2011). La importancia de la productividad como componente de la competitividad, pp. 1–10.
- Briand, L. C., El-Emam, K., & Bomarius, F. (1998). COBRA: A hybrid method for software cost estimation, benchmarking, and risk assessment. In *IEEE Computer Society Press (Ed.), Proceedings of The 20th International Conference on Software Engineering* (pp. 390-399). Kyoto, Japan, April 19-25.

Brooks, F. P. (1995). *The Mythical Man-Month*. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, anniversary ed.

Brunekreef, J., & Diertens, B. (2002). Towards a user-controlled software renovation factory. *Science of Computer Programming*, 45(2-3), 175-191.

Card, D. N. (2006). The Challenge of Productivity Measurement, *Proceeding of Pacific Northwest Software Quality Conference*, Portland.

Cao, Q., Ching Gu, V., & Thompson, M. A. (2012). Using complexity measures to evaluate software development projects: A nonparametric approach. *The Engineering Economist: A Journal Devoted to the Problems of Capital Investment*, 57(4), 274-283.

Card, D. N. (2006). The challenge of productivity measurement. *Proceedings of Pacific Northwest Software Quality Conference*, Portland, January.

Cardoso, C., Bert, F., & Podestá, G. (2010). Modelos Basados en Agentes (MBA): definición, alcances y limitaciones. *Landuse, biofuels and rural development in the La Plata Basin, IDRC Grant ID 104783-001*.

Castañeda P., & Mauricio D. (2018). A Review of Literature About Models and Factors of Productivity in the Software Factory, *International Journal of Information Technologies and Systems Approach*, Volume 11 No 1, pp. 48-71.

Castañeda, P., & Mauricio, D. (in press). New Factors Affecting Productivity Of The Software Factory, *International Journal of Information Technologies and Systems Approach*.

Castañeda, P., & Mauricio, D. (in press). Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Factory, *International Journal of Information Technologies and Systems Approach*.

Cao Q., Ching Gu V., & Thompson M. A. (2012). Using Complexity Measures to Evaluate Software Development Projects: A Nonparametric Approach, *The Engineering Economist: A Journal Devoted to the Problems of Capital Investment*, 57:4, 274-283, DOI: 10.1080/0013791X.2012.729878.

Celina, H., & Campo A. (2005). Aproximación al uso del coeficiente alfa de Cronbach, *Revista colombiana de psiquiatría*, vol. XXXIV, número 004, *Asociación Colombiana de*

*Psiquiatría*, Bogotá, Colombia, pp. 572 – 580, disponible en: <http://redalyc.uaemex.mx/pdf/806/80634409.pdf>

Çetin, F., & Alabaş-Uslu, Ç. (2015). Performance evaluation of projects in software development. *Journal of Aeronautics and Space Technologies*, 8(2), 1-6.

Charnes, A. et al (1997): *Data Envelopment Analysis: Theory, Methodology and Applications*, New York, Kluwer Academic Publishers, Second edition.®

Charnes A., Cooper W. W., & Rhodes E. (1978). Measuring the efficiency of decision making units. *Eur J Oper Res* 2:429–444.

Charnes, A., Cooper, W. W., Lewin, A. Y., & Seiford, L. M. (1994). *Data envelopment analysis: theory, methodology, and application*.

Cheikhi, L., Al-Qutaish, R. E., & Idri, A. (2012). Software productivity: Harmonization in ISO/IEEE software engineering standards. *Journal of Software*, 7(2), 462-470.

Chicano, J., Luna, F., Nebro, A. J., & Alba, E. (2011). Using multi-objective metaheuristics to solve the software project scheduling problem. In: *13th Annual Conference on Genetic and evolutionary computation (GECCO'11)*, GECCO'11. ACM, New York, NY, USA, pp. 1915–1922.

Chung, S. H., Lee, A. H. I., Kang, H. Y., & Lai, C. W. (2008). A DEA window analysis on the product family mix selection for a semiconductor fabricator, *Expert Syst. with Appl.*, vol. 35, pp. 379–388.

Clements, P. C., & Northrop, L. M. (2001). *Software product lines: Practices and patterns*. Boston: Addison-Wesley Professional.

CMMI (2016). Capability Maturity Model Integration. CMMI Institute. Recuperado de: <http://cmmiinstitute.com/>

CMMI Institute (2018). Published Appraisal Results, <https://sas.cmmiinstitute.com/pars/>

Coelli, T. J., Rao, D. P., O'Donnell, C. J., & Battese, G. E. (2005). *An introduction to efficiency and productivity analysis*. New York, USA: Springer Science + Business Media.



Cohen, S. G., & Bailey, D. E. (1997), “What makes teams work: group effectiveness research from the shop floor to the executive suite”, *Journal of Management*, Vol. 23 No. 3, pp. 239-90.

Cook, W. D. & Seiford, L. M. (2009) Data envelopment analysis (DEA)—thirty years on. *Eur J Oper Res* 192:1–17.

Cooper, W. W., Seiford, L. M., & Zhu, J. (2004) Handbook on data envelopment analysis. Springer, New York.

Cooper, W. W., Seiford, L. M., & Tone, K. (2006) *Introduction to data envelopment analysis and its uses with DEA solver software and references*. Springer Science and Business Media, New York.

Cusumano, M. A. (1989). The software factory: A historical interpretation. *IEEE Software*, 6(2), 23-30.

Cusumano, M., Mac Cormack, M. A., Kemerer, C. F., & Crandall, B. (2003). Software development worldwide: The state of the practice, *IEEE Software*, vol. 20, pp. 28-34, Nov. /Dec.

Dale, C. J., & Van der Zee, H. (1992). Software productivity metrics: who needs them? *Info. & Softw. Technol.*, vol. 34, pp. 731-738.

Di Penta, M., Harman, M., Antoniol, G., & Qureshi, F. (2007). The effect of communication overhead on software maintenance project staffing: a search-based approach. In: *IEEE International Conference on Software Maintenance (ICSM 2007)*, pp. 315–324.

Echeverry, J. C., y Hernández, M. (2005). Posibilidades y limitantes de un cambio en la productividad de los sectores colombianos: textiles – confecciones, avícola – porcícola, siderurgia – metalmecánica y galletería – confitería – chocolatería. C.E.D.E. Universidad de los Andes, (40), 1 – 127.

Fabri, J. A., Trindade, A. L. P., Begosso, L. R., Lerario, I., Silveira, F. L. F., & Pessoa, M. S. P. (2004). Techniques for the development of a software factory: Case CEPEIN-FEMA. In: *17th International Conference Software & Systems Engineering and their Applications*, Paris.

- Fabri, J. A., Trindade, A. L. P., L'erário, A., Pessoa, M. S. P., & Spinola, M. (2004a). Desenvolvimento e Replicação de uma Fábrica de uma Software. In: *VI Simpósio Internacional de Melhoria de Processo de Software – SIMPROS*, São Paulo.
- Fabri, J. A., Trindade, A. L. P., Durscki, R., Spinola, M. M., & Pessoa, M. S. P. (2005). Proposta de um Mecanismo de Desenvolvimento e Customização de uma Fábrica de Software Orientada a Domínios. In: *XXXI Latin American Computing Conference*, Cali.
- Fabri, J. A., Trindade, A. L. P., Begosso, L. R., Lerario, I., & Pessoa, M. S. P. (2007). A Organização de uma Máquina de Processo e a Melhoria do Processo de Produção de Software em um Ambiente de Fábrica. In: *VI Jornadas Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento*, Lima, Perú.
- Fabri, J. A., Trindade, A. L. P., Silveira, M., & Pessoa, M. S. P. (2007a). O Papel do CMMI na Configuração de um Meta-Processo de Produção de Software com Características Fabris: Um Estudo de Caso. In: *VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento* (pp. 375-383), Lima, Perú.
- Fabri, J. A., Scheible, A. C. F., Moreira, P. M. L., Trindade, A. L. P., Begosso, L. R., Braun A. P., & Pessoa, M. S. P. (2007b). Meta-process used for production process modeling of a software factory: The Unitech case. In: *Managing Worldwide Operations and Communications with Information Technology–IRMA 2007*, Vancouver, Canadá.
- Fabri, J. A., Trindade, A. L. P., Begosso, L. R., Pessoa, M. S. P., & L'erário, A. (2007c). The use of the idef-0 to model the process in a software factory. In: *Managing Worldwide Operations and Communications with Information Technology–IRMA 2007*, Vancouver, Canadá.
- Farrell, M. J. (1957). The measurement of productive efficiency. *J R Stat Soc* 120:253–290
- Fenton, N. E., & Pfleeger, S. L. (1997). Measuring productivity. In *Software metrics: A Rigorous & Practical Approach*, 412-425.
- Fernandes, A. A., & Teixeira, D. S. (2004). *Fábrica de software: Implantação e gestão de operações*. São Paulo: Editora Atlas.
- Flitman, A. (2003). Towards meaningful benchmarking of software development team productivity. *Benchmarking Int J* 10(4):382–399.

- Flores, F., & Ludlow, J. (1980). Doing and Speaking in the Office, in: *Decision Support Systems: Issues and Challenges*, G. Fick and R.H. Sprague (eds.), Pergamon Press, New York, pp. 95-118.
- Foss, W. B. (1993). Fast, faster, fastest development. *Computerworld*, 27(22), 81-83.
- GMD (2018). GMD. Lima, Perú. Recuperado de <http://www.gmd.com.pe/>
- Greenfield, J., Short, K., Cook, S., Kent, S., & Crupi, J. (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* (1st ed.): Wiley.
- Griss, M. (1993). Software reuse: from library to factory. *IBM System Journal*, 32(4), 548-566.
- Gueorguiev, S., Harman, M., & Antoniol, G. (2009). Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*, ACM, Montral, Canada (Best Paper Award), pp. 1673–1680.
- Harman, M., & Jones, B.F. (2001). Search-based software engineering. *Information and Software Technology* 43, 833–839.
- Hernández, A., Colomo R., & García, A. (2012). Productivity in software engineering: A study of its meanings for practitioners: Understanding the concept under their standpoint. In *Information Systems and Technologies (CISTI)*, Madrid, Spain.
- IDC (2017). Latin America Semiannual IT Services Tracker. Recuperado de <http://www.gmd.com.pe/novedades/noticias/news-15>
- IEEE (1992). *IEEE Standard for Software Productivity Metrics 1045*, <http://ieeexplore.ieee.org/servlet/opac?punumber=2858>
- Jacobson I., Booch G. & Rumbaugh J. (2000). *El Proceso Unificado de Desarrollo de Software*, Pearson Educación S.A., Madrid, pp. 321-322.
- Jeffery, D. R., & Low, G. C. (1990). Calibrating estimation tools for software development, *Software Engineering Journal*, vol. 5, pp. 215-221.

- Jeffery, R., Ruhe, M., & Wieczorek, I. (2001). Using public domain metrics to estimate software development effort. In: *Proceedings of METRICS'01*, pp 16-27.
- Jiang, Z., Naudé, P., & Comstock, C. (2007). An investigation on the variation of software development productivity. *International Journal of Computer and Information Science and Engineering*, 1(2), 72-81.
- Jorgensen, M., Indahl, U., & Sjøberg, D. I. K. (2003). Software effort estimation by analogy and 'regression toward the mean', *J. of Systems & Software*, vol. 68, pp. 253-262.
- Khan, R., Ahmed, I., & Faisal, M. (2014). An industrial investigation of human factors effect on software productivity: Analyzed by SEM model. *International Journal of Computer Science and Mobile Computing*, 3(5), 16-24.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, *Software Engineering Group*, School of Computer Science and Mathematics, Keele University.
- Kremmel, T., Kubalik, J., & Biffel, S. (2010). Multiobjective evolutionary algorithm for software project portfolio optimization. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO'10)*. ACM, Portland, Oregon, USA, pp. 1389–1390.
- Kremmel, T., Kubalik, J., & Biffel, S. (2011). Software project portfolio optimization with advanced multiobjective evolutionary algorithms. *Applied Soft Computing* 11, 1416–1426.
- Kruchten, P. (2004). *The rational unified process: An introduction* (3rd ed.). Boston: Addison-Wesley.
- Krugman, P. (1997). *El Internacionalismo Moderno*. México. Ed. Crítica.
- Li, C., Li, H., & Li, M. (2001). A software factory model based on ISO 9000 and CMM for Chinese small organizations. *Second Asia-Pacific Conference on Quality Software (APAQS'01)*, Hong Kong, China.
- Liu, W., Meng, W., Li, X., & Zhang, D. (2010). DEA models with undesirable inputs and outputs, *Ann. Operations Res.*, vol. 173, pp. 177–194.

- López, C., Kalichanin, I., Meda, M. E., & Chavoya, A. (2010) Software development productivity prediction of small programs using fuzzy logic. *Seventh International Conference on Information Technology*, Las Vegas, Nevada, USA, April 12-14.
- Machek, O., Hnilica, J., & Hejda, J. (2012). Estimating productivity of software development using the total factor productivity approach. *International Journal of Engineering Business Management*, 4, 4–34. doi:10.5772/52797
- Mahmood M.A., Pettingell K.J., & Shaskevich A.I. (1996). Measuring productivity of software projects: a data envelope analysis approach. *Decis Sci* 27(1):57–80.
- Maxwell, K. D. (2001). Collecting data for comparability: Benchmarking software development productivity. *IEEE Software*, 18(5), 22-25.
- Montgomery Jr., P. (1981). A model of the software development process. *Journal of Systems and Software* 2, 237–255.
- Moreira, C. I., Carneiro, C., Pires, C. S., & Bessa, A. (2010). A practical application of performance models to predict the productivity of projects. In T. Sobh (Ed.), *Innovations and advances in computer sciences and engineering*, 273-277.
- Myrtveit I., & Stensrud E. (1999). Benchmarking COTS projects using data envelopment analysis. In: *Proceedings of METRICS'99*, Washington, D.C., pp 269–278.
- Nyhan, R. C., & Martin, L. L. (1999). Comparative Performance Measurement, a Primer on Data Envelopment Analysis. *Public Productivity & Management Review*, 22(3), 348-364.
- Nomura, L., Spinola, M., Hikage, O., & Tonini, A. C. (2006). FS-MDP: Um Modelo de Definição de Processos de Fábrica de Software, *XXVI Encontro Nacional de Engenharia de Produção*, Fortaleza, Brazil, October 09-11.
- Norden, P. (1977). Project life cycle modeling: background and application of the life cycle curves. In: U.S. Army Computer Systems Command.
- Norden, P. V. (1958). Curve fitting for a model of applied research and development scheduling. *IBM Journal of Research and Development* 2, 232–248.
- Nwelih E., & Amadin I. F. (2008). Modeling software reuse in traditional productivity model, *Asian Journal of Information Technology* 7, 484-488.

- Pai, D. R., Subramanian, G. H., & Pendharkar, P. C. (2015). Benchmarking software development productivity of CMMI level 5 projects. *Information Technology and Management*, 16(3), 235-251.
- Paiva, E., Barbosa, D., Lima, R. & Albuquerque, A. (2010). Factors that influence the productivity of software developers in a developer view. In: Sobh T., Elleithy K. (Ed.), *Innovations in Computing Sciences and Software Engineering*, pp. 99-104.
- Paradi, J.C., Reese, D.N., & Rosen, D. (1997) Applications of DEA to measure the efficiency of software production at two large Canadian banks. *Ann Oper Res* 73:91–115.
- Parthasarathy, S., & Anbazhagan, N. (2008). Evaluating ERP projects using DEA and regression analysis. *Int J Bus Inf Syst* 3(2):140–157.
- Pendharkar, P.C., Rodger, J.A. (2007). An empirical study of the impact of team size on software development effort. *Information Technology and Management* 8, 253–262.
- Pendharkar, P.C., & Rodger, J. (2009) The Relationship between software development team size and software development cost. *Commun ACM* 52(1):141–144.
- Pessoa, M. S. P., Fabri, J. A., L'erário, A., Spinola, M., & Begosso, A. (2004). Desenvolvimento e Replicação de uma Fábrica de uma Software. In: *IV Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento*, Madrid, España.
- Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, 53(4), 317-343.
- Piattini, M., & Garzás, J. (2010). *Fábricas de software: Experiencias, Tecnologías y Organización*. 2nd ed. Paracuellos del Jarama (Madrid): Ra-ma.
- Piattini, M., Caballero, I., & García, F. (2006). *Calidad de los Sistemas Informáticos*. Madrid: Ra-ma.
- Piattini, M., & Daryanani, S. (1995). *Elementos y Herramientas en el Desarrollo de los Sistemas de Información*. Una visión actual de la tecnología CASE: Ra-MA.
- Pillai, K., & Sukumaran, V. S. (1997). A model for software development effort and cost estimation. *IEEE Transactions on Software Engineering* 23, 485–497.

- Polo, M., Piattini, M., & Ruiz, F. (2003). *Advances in Software Maintenance Management: Technologies and Solutions*. EEUU: Idea Group Publishing.
- Premraj, R., Shepperd, M., Kitchenham, B. & Forselius, P. (2005). An empirical analysis of software productivity over time. Presented at *11th IEEE International Software Metrics Symposium (Metrics05)*, Como, Italy, September 19-22.
- Ramanathan, R. (2002). *Introductory Econometrics with Applications*, 5a edn., South-Western.
- Reyes, J. (2010). El residuo de Solow [versión electrónica]. *Revista de Economía Institucional*, vol. 12 (23), segundo semestre de 2010.
- Rockwell, R., & Gera, M. H. (1993). The Eureka Software Factory CoRe: A conceptual reference model for software factories. *Software Engineering Environments Conference*, UK, United Kingdom, July 7-9.
- Rodger, J. A., Pankaj, P., & Nahouraii, A. (2011). Knowledge management of software productivity and development time. *Journal of Software Engineering and Applications*, 4(11), 609-618.
- Rodríguez, D., Sicilia, M. A., García, E., & Harrison, R. (2011). Empirical findings on team size and productivity in software development. *Journal of Systems and Software*, 85(3), 562-570.
- Romo, D., & Andel, G. (2005). Sobre el concepto de competitividad. *Revista Comercio Exterior*, 55 (3).
- SBOK (2016). A Guide to the Scrum Body of Knowledge. USA: SCRUMStudy. Recuperado de: <https://www.scrumstudy.com/>
- Scacchi, W. (1995). Understanding software productivity. In D.Hurley (Ed.), *Software Engineering and Knowledge Engineering: Trends for the next decade Vol. 4*, 37-70, Los Angeles, World Scientific Press.
- Sentas, P., Angelis, L., Stamelos, I. & Bleris, G. (2005). Software productivity and effort prediction with ordinal regression, *Info. & Softw. Technol.*, vol. 47, pp. 17-29.

Simmons, D. B. (1991). Communications: a software group productivity dominator, in *Software Engineering Journal*, vol. 6, no. 6, pp. 454-462. doi: 10.1049/sej.1991.0044

Smith, R. K., Hale, J.E., & Parish, A.S. (2001). An empirical study using task assignment patterns to improve the accuracy of software effort estimation. *IEEE Transactions on Software Engineering* 27, 264–271.

Sridhar, V., Paul, R., Nath, D., & Kapur, K. (2007), “Analyzing factors that affect performance of global virtual teams”, *Proceedings of the 2nd International Conference on Management of Soft factors affecting performance 203 Globally Distributed Work 2007*, Indian Institute of Management (IIM), Bangalore, India, July 25-27, pp. 159-69.

Stensrud E., & Myrtveit I. (2003). Identifying high performance ERP projects. *IEEE Trans Comput* 29(5):398–416.

Swanson, K., McComb, D., Smith, J., & McCubbrey, D. (1991). The application software factory: Applying total quality techniques to systems development. *MIS Quarterly*, 15(4), 567-579.

Subramanian, G. H., & Zarnich, G. E. (1996). An examination of some software development effort and productivity determinants in ICASE tool projects. *Journal of Management Information Systems*, 12(14), 143-160.

Sudhakar G., Farooq A., & Patnaik S. (2011). Soft factors affecting the performance of software development teams, *Team Performance Management Vol. 17* No. 3/4, 2011 pp. 187-205.

Sudhakar, G., Farooq, A., & Patnaik, S. (2012). Measuring productivity of software development teams. *Serbian Journal of Management*, 7(1), 65-75.

Swanson, K., Kent, D., McComb, J., & Dave, D. (1991). The Application Software Factory: Applying Total Quality Techniques to Systems Development. *MIS Quarterly*, 15(4), 567-579.

TechAmerica (1999). *ANSI-EIA-632 Standard: Processes for Engineering a System*.

Tonkay, G.L. (2008). *Productivity in Encyclopedia of Science & Technology*. McGraw-Hill.



Trendowicz, A. (2007). Factors influencing software development productivity—state of the art and industrial experiences. *Technical Report 08.07/E*, Fraunhofer IESE, Kaiserslautern, Germany.

Trendowicz, A., & Munch, J. (2009). Factors Influencing Software Development Productivity – State-of-the-art and Industrial Experiences, *vol. 77 of Advances in Computers*. Elsevier, pp. 185–241.

Trujillo, Y. T. (2007). Modelo de factoría de software aplicando inteligencia. *Serie Científica de la Universidad de las Ciencias Informáticas*, 1(1).

Unluturk, M. S., & Kurtel, K. (2015). Quantifying productivity of individual software programmers: Practical approach. *Computing and Informatics*, 34(4), 959-972.

Van den Brand, M., Sellink, A., & Verhoef, C. (2000). Generation of components for software renovation factories from context-free grammars. *Science of Computer Programming*, 36(2-3), 209-266.

Verner, J., Evanco, W., & Cerpa, N. (2007). State of the practice: an exploratory analysis of schedule estimation and software project success prediction. *Information and Software Technology* 49, 181–193.

Villa, G. (2003). *Análisis por Envoltura de Datos (DEA): Nuevos Modelos y Aplicaciones* (tesis de doctorado). Universidad de Sevilla, Sevilla, España.

Wagner, S., & Ruhe, M. (2008). *A structural review of productivity factors in software development*. Technical Report, Technische Universität München.

Wagner, S., & Ruhe, M. (2008a). A systematic review of productivity factors in software development. *Proceedings of 2nd International Workshop on Software Productivity Analysis and Cost Estimation*, State Key Laboratory of Computer Science, Institute of Software.

Wegner, D. M. (1987). Transactive memory: a contemporary analysis of the group mind, in *Theories of Group Behavior*, eds Mullen B., Goethals G. R., editors. (New York, NY: Springer-Verlag), 185–208.

Wray, B., & Mathieu, R. (2008). Evaluating the performance of open source software projects using data envelopment analysis. *Inf Manag Comput Secur* 16(5):449–462.

Yang, Z., & Paradi, J. C. (2004). DEA evaluation of a Y2K software retrofit program. *IEEE Trans Eng Manag* 51(3):279–287.

Yanosky, M. M. V. (2005). Modelo funcional de la Factoría de Software de la UCI para la línea Carrefour.

Yilmaz, M., & O'Connor, R. V. (2011). An empirical investigation into social productivity of a software process: An approach by using the structural equation modeling. *European Conference on Software Process Improvement 2011: Communications in Computer and Information Science*, 172, 155-166.

Zhan, J., Zhou, X., & Zhao, J. (2012). Impact of software complexity on development productivity. *International Journal of Software Engineering and Knowledge Engineering*, 22(8), 1103-1122.

## Anexo A: Factores de la Revisión de la Literatura

ID	Unidad de Trabajo	Categoría	Factor	Descripción	Referencia
<i>Categoría: Producto</i>					
F1	Programación	Producto	Volatibilidad	Volatibilidad en tipo de desarrollo de software (Nuevo, Mantenimiento)	Rodger <i>et al.</i> , 2011
F2	Programación	Producto	Plataforma de Desarrollo	Plataforma de desarrollo utilizada en el proyecto	Rodger <i>et al.</i> , 2011
F3	Programación	Producto	Lenguaje de Programación	Tipo de lenguaje de programación	Rodger <i>et al.</i> , 2011
F4	Programación	Producto	Reutilización	Porcentaje de objetos reutilizados	Machek <i>et al.</i> , 2012; Yilmaz & O'Connor, 2011; Nwelih & Amadin, 2008
F5	Programación	Producto	Funcionalidad	Cantidad de puntos de función	Machek <i>et al.</i> , 2012; Nwelih & Amadin, 2008
F6	Programación	Producto	Longitud	LOC corregido con respect a la densidad de comentarios	Machek <i>et al.</i> , 2012; Nwelih & Amadin, 2008
F7	Programación	Producto	Datos	Líneas de datos útiles producidos	Machek <i>et al.</i> , 2012

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F8</i>	Programación	Producto	Documentación	Líneas de documentación producida	Machek <i>et al.</i> , 2012
<i>F9</i>	Programación	Producto	Sentencias fuente (SS)	Entregas SS nuevas Entregas SS reutilizadas	Cheikhi <i>et al.</i> , 2012; Unluturk & Kurtel, 2015
<i>F10</i>	Programación	Producto	Puntos de Función	Número de puntos de función	Cheikhi <i>et al.</i> , 2012; Pai <i>et al.</i> , 2015; López <i>et al.</i> , 2010
<i>F11</i>	Programación	Producto	Lanzamientos	Número total de lanzamientos	Çetin & Alabaş-Uslu, 2015
<i>F12</i>	Programación	Producto	Incidencias	Número total de errores y problemas observados en todo el proyecto	Çetin & Alabaş-Uslu, 2015
<i>F13</i>	Programación	Producto	Riesgos	Número total de eventos que pueden terminar con un impacto negativo	Çetin & Alabaş-Uslu, 2015
<i>F14</i>	Programación	Producto	Solicitudes de Cambio	Número de solicitudes de cambio	Çetin & Alabaş-Uslu, 2015
<i>F15</i>	Programación	Producto	Tipo de Conteo	Físico/Lógico	López <i>et al.</i> , 2010
<i>F16</i>	Programación	Producto	Sentencia	Tipo de sentencia (Ejecutable, No ejecutable)	López <i>et al.</i> , 2010

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F17</i>	Programación	Producto	Complejidad	Nivel de complejidad del proyecto (puede ser complejo, moderadamente complejo o no complejo)	Machek <i>et al.</i> , 2012; Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014; Çetin & Alabaş-Uslu, 2015; Nwelih & Amadin, 2008
<i>F18</i>	Pruebas	Producto	Densidad de defectos en pruebas sistémicas	Número de defectos confirmados detectados en la prueba durante un período definido	Moreira <i>et al.</i> , 2010; Pai <i>et al.</i> , 2015; Unluturk & Kurtel, 2015
<i>F19</i>	Pruebas	Producto	Defectos en las revisiones técnicas	Porcentaje de defectos en las revisiones técnicas	Moreira <i>et al.</i> , 2010; Pai <i>et al.</i> , 2015; Unluturk & Kurtel, 2015
<i>F20</i>	Pruebas	Producto	Cobertura de prueba unitaria	Porcentaje de cobertura de prueba unitaria	Moreira <i>et al.</i> , 2010
<i>F21</i>	Análisis & Diseño	Producto	Tipos de objetos por técnica	Cantidad de tipos de objetos por técnica	Cao <i>et al.</i> , 2012
<i>F22</i>	Análisis & Diseño	Producto	Tipos de relación por técnica	Cantidad de tipos de relación por técnica	Cao <i>et al.</i> , 2012

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F23</i>	Análisis & Diseño	Producto	Tipos de propiedad técnica	de Cantidad de tipos de propiedad por técnica	Cao <i>et al.</i> , 2012
<i>F24</i>	Análisis & Diseño	Producto	Número de propiedades un tipo de objeto dado	de Cantidad de número de propiedades para un tipo de objeto dado	Cao <i>et al.</i> , 2012
<i>F25</i>	Análisis & Diseño	Producto	Promedio de propiedades un tipo de objeto dado	de Promedio de propiedades para un tipo de objeto dado	Cao <i>et al.</i> , 2012
<i>F26</i>	Análisis & Diseño	Producto	Número de propiedades de un tipo de relación y los tipos de roles que lo acompañan	de Número de propiedades de un tipo de relación y los tipos de roles que lo acompañan	Cao <i>et al.</i> , 2012
<i>F27</i>	Análisis & Diseño	Producto	Promedio de propiedades tipo de relación	de Promedio de propiedades por tipo de relación	Cao <i>et al.</i> , 2012

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F28</i>	Análisis & Diseño	Producto	Cantidad de tipos de relaciones que se pueden conectar a un determinado tipo de objeto	Cantidad de tipos de relaciones que se pueden conectar a un determinado tipo de objeto	Cao <i>et al.</i> , 2012
<i>F29</i>	Análisis & Diseño	Producto	Número promedio de tipos de relaciones que se pueden conectar a un determinado tipo de objeto	Número promedio de tipos de relaciones que se pueden conectar a un determinado tipo de objeto	Cao <i>et al.</i> , 2012
<i>Categoría: Proceso</i>					
<i>F30</i>	Programación	Proceso	Entrenamiento	Número de personas entrenadas	Machek <i>et al.</i> , 2012
<i>F31</i>	Programación	Proceso	Proceso	No definido	Yilmaz & O'Connor, 2011
<i>F32</i>	Programación	Proceso	Reuniones	Frecuencia de reuniones	Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F33</i>	Programación	Proceso	Documentación (Tipo, Origen, Uso)	Recuento de páginas de documentos, cantidad de pantalla, número de palabras, número de ideogramas, número de gráficos	Cheikhi <i>et al.</i> , 2012
<i>F34</i>	Programación	Proceso	Metodología	Metodologías de desarrollo de software	Çetin & Alabaş-Uslu, 2015
<i>F35</i>	Programación	Proceso	Líneas Base	Número de líneas base (línea base es un punto de referencia)	Çetin & Alabaş-Uslu, 2015
<i>F36</i>	Programación	Proceso	Severidad	Nivel de severidad del proyecto	Çetin & Alabaş-Uslu, 2015
<i>F37</i>	Programación	Proceso	Duración de Análisis	Tiempo de análisis	Çetin & Alabaş-Uslu, 2015
<i>F38</i>	Programación	Proceso	Duración de Desarrollo	Tiempo de desarrollo	Çetin & Alabaş-Uslu, 2015
<i>F39</i>	Programación	Proceso	Duración de Pruebas	Tiempo de pruebas	Çetin & Alabaş-Uslu, 2015
<i>F40</i>	Programación	Proceso	Ocurrencia de Espera	Número de ocurrencias de espera	Çetin & Alabaş-Uslu, 2015



<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F41</i>	Programación	Proceso	Duración de Espera	Tiempo transcurrido hasta que el proyecto comience nuevamente	Çetin & Alabaş-Uslu, 2015
<i>F42</i>	Programación	Proceso	Motivo de Espera	Razón de espera	Çetin & Alabaş-Uslu, 2015
<i>F43</i>	Pruebas	Proceso	Inestabilidad de Requerimientos	Nivel de Inestabilidad de Requerimientos	Moreira <i>et al.</i> , 2010
<i>F44</i>	Pruebas	Proceso	Utilización de Integración Continua	Nivel de Utilización de Integración Continua	Moreira <i>et al.</i> , 2010
<i>Categoría: Entorno de Desarrollo</i>					
<i>F45</i>	Programación	Entorno de Desarrollo	de Entorno de Desarrollo	No definido	Moreira <i>et al.</i> , 2010
<i>Categoría: Cultura Corporativa</i>					
<i>F46</i>	Programación	Cultura Corporativa	Sector del Negocio	No definido	Premraj <i>et al.</i> , 2010
<i>F47</i>	Programación	Cultura Corporativa	Tecnología	No definido	Khan <i>et al.</i> , 2014
<i>F48</i>	Programación	Cultura Corporativa	Proyecto	Tipos de proyectos tales como infraestructura, servicio, producto,	Çetin & Alabaş-Uslu, 2015

ID	Unidad de Trabajo	Categoría	Factor	Descripción	Referencia
				viabilidad. Cada proyecto debe clasificar en uno de estos tipos	
<i>Categoría: Cultura del Equipo</i>					
F49	Programación	Cultura del Equipo	del Cohesión	Cohesión del equipo, Unidad del equipo	Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014
F50	Programación	Cultura del Equipo	del Liderazgo del Equipo	No definido	Yilmaz & O'Connor, 2011
F51	Programación	Cultura del Equipo	del Resultados Colectivos	No definido	Yilmaz & O'Connor, 2011
F52	Programación	Cultura del Equipo	del Conciencia de la Información	No definido	Yilmaz & O'Connor, 2011
F53	Programación	Cultura del Equipo	del Transparencia	No definido	Yilmaz & O'Connor, 2011
F54	Programación	Cultura del Equipo	del Relaciones sociales, vida social	No definido	Yilmaz & O'Connor, 2011; Khan <i>et al.</i> , 2014
F55	Programación	Cultura del Equipo	del Cultura Trabajo	de No definido	Khan <i>et al.</i> , 2014

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
<i>F56</i>	Programación	Cultura Equipo	del Trabajo Individual	Interés en el No definido	Khan <i>et al.</i> , 2014
<i>Categoría: Capacidades y Experiencia</i>					
<i>F57</i>	Programación	Capacidades Experiencia	y Motivación	No definido	Yilmaz & O'Connor, 2011
<i>F58</i>	Programación	Capacidades Experiencia	y Experiencia	No definido	Moreira <i>et al.</i> , 2010
<i>F59</i>	Programación	Capacidades Experiencia	y Liderazgo	No definido	Yilmaz & O'Connor, 2011
<i>F60</i>	Programación	Capacidades Experiencia	y Confianza	No definido	Yilmaz & O'Connor, 2011
<i>F61</i>	Programación	Capacidades Experiencia	y Comunicación	No definido	Yilmaz & O'Connor, 2011
<i>F62</i>	Programación	Capacidades Experiencia	y Habilidades Gerente	del No definido	Khan <i>et al.</i> , 2014
<i>Categoría: Proyecto</i>					
<i>F63</i>	Programación	Proyecto	Tamaño Equipo	de Tamaño del equipo del proyecto	Rodger <i>et al.</i> , 2011; Yilmaz & O'Connor, 2011; Khan <i>et</i>

<b>ID</b>	<b>Unidad de Trabajo</b>	<b>Categoría</b>	<b>Factor</b>	<b>Descripción</b>	<b>Referencia</b>
					<i>al.</i> , 2014; Çetin & Alabaş-Uslu, 2015
<i>F64</i>	Programación	Proyecto	Trabajo de Ingeniería	de Horas-hombre	Machek <i>et al.</i> , 2012
<i>F65</i>	Programación	Proyecto	Trabajo de Pruebas	de Horas-hombre	Machek <i>et al.</i> , 2012
<i>F66</i>	Programación	Proyecto	Trabajo Administrativo	Horas-hombre, personal	Horas- Machek <i>et al.</i> , 2012; Cheikhi <i>et al.</i> , 2012
<i>F67</i>	Programación	Proyecto	Soporte Laboral	Horas-hombre	Machek <i>et al.</i> , 2012
<i>F68</i>	Programación	Proyecto	Materiales y servicios	Valor deflactado	Machek <i>et al.</i> , 2012
<i>F69</i>	Programación	Proyecto	Stock de Capital Tangible	Valor deflactado	Machek <i>et al.</i> , 2012
<i>F70</i>	Programación	Proyecto	Stock de Capital Intangible	Valor deflactado	Machek <i>et al.</i> , 2012
<i>F71</i>	Programación	Proyecto	Otro Capital	Valor deflactado	Machek <i>et al.</i> , 2012
<i>F72</i>	Programación	Proyecto	Costo Personal	del Costo de gestión	Cheikhi <i>et al.</i> , 2012
<i>F73</i>	Programación	Proyecto	Duración del proyecto	del Tiempo total transcurrido del proyecto	Çetin & Alabaş-Uslu, 2015

## Anexo B: Encuesta de investigación para determinar nuevos factores que influyen en productividad

Ruta de la encuesta: <http://tinyurl.com/y5kbpk7p>

### Estructura de la encuesta:

#### Encuesta para determinar los factores que afectan a la productividad en la Fábrica de Software

La Fábrica de Software es una organización altamente organizada que produce partes de software en una línea de producción utilizando elementos estandarizados, herramientas y procesos de producción. En este contexto, la fábrica se divide en tres grandes componentes: Análisis y Diseño de Software, Programación de Software y Pruebas de Software.

Esta encuesta está dirigida a los equipos de desarrollo de software que participan dentro de un entorno de Fábrica de Software. Sus respuestas son anónimas y serán utilizadas estrictamente con fines de investigación.

La Encuesta está dividida en 3 secciones: la Sección 1 está relacionada con la caracterización de la empresa y de las personas que llenan el cuestionario. La Sección 2 corresponde a preguntas que determinen la influencia de los factores que afectan la productividad en los diferentes componentes de la Fábrica de Software y la Sección 3 contiene preguntas que complementan el estudio.

¡Gracias por su colaboración!

1. <i>¿La empresa donde trabajas actualmente viene ejecutando proyectos de Fábrica de Software? Si su respuesta es NO, no es necesario que complete el siguiente cuestionario.</i>	
*	
<input type="radio"/>	Si
<input type="radio"/>	No

**Sección 1: Información General**

<i>1. ¿Cuántos empleados en la empresa vienen ejecutando proyectos de Fábrica de Software?</i>	
<input type="radio"/>	Entre 10 y 100 empleados
<input type="radio"/>	Entre 101 y 500 empleados
<input type="radio"/>	Entre 501 y 1000 empleados
<input type="radio"/>	Más de 1000 empleados
<i>2. ¿Cuál es tu cargo dentro de la empresa?</i>	
<input type="radio"/>	Gerente de Proyecto
<input type="radio"/>	Jefe de Proyecto
<input type="radio"/>	Gestor de Métricas
<input type="radio"/>	Coordinador de Análisis/Programación/Calidad
<input type="radio"/>	Analista de Sistemas y/o Software
<input type="radio"/>	Otro: .....
<i>3. ¿Para qué sector de negocio se encuentra la empresa desarrollando software?</i>	
<input type="radio"/>	Gobierno
<input type="radio"/>	Financiero
<input type="radio"/>	Transporte

<input type="radio"/>	Logística
<input type="radio"/>	Marketing
<input type="radio"/>	Comercio
<input type="radio"/>	Servicios
<input type="radio"/>	Otro: .....
4. <i>¿En qué componente de la Fábrica de Software vienes laborando?</i>	
<input type="radio"/>	Componente de Análisis y Diseño de Software
<input type="radio"/>	Componente de Programación de Software
<input type="radio"/>	Componente de Pruebas de Software
<input type="radio"/>	Otro: .....
5. <i>¿Cuántos años vienes trabajando en proyectos de Fábrica de Software?</i>	
<input type="radio"/>	Entre 1 y 5 años
<input type="radio"/>	Entre 6 y 10 años
<input type="radio"/>	Más de 10 años

## **Sección 2: Factores de Productividad de la Fábrica de Software**

En esta sección se plantean preguntas orientadas a la influencia de diversos factores en cada uno de los componentes de la fábrica. Para dar respuesta a las siguientes preguntas, considere las siguientes escalas:

1. No tiene Influencia
2. Baja Influencia
3. Mediana Influencia
4. Alta Influencia
5. Influye Totalmente

En cada una de las siguientes preguntas, tendrá que utilizar la siguiente tabla para marcar la influencia del factor en los diferentes componentes:

	No tiene Influencia	Baja Influencia	Mediana Influencia	Alta Influencia	Influye Totalmente
En Análisis y Diseño de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
En Programación de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
En Pruebas de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1. *¿Considera Ud. que el USO DE REGLAS Y EVENTOS tiene influencia en un mejor desempeño de las actividades del equipo? \**

Entiéndase por "uso de reglas y eventos" al establecimiento de políticas y/o normas que faciliten el desarrollo de actividades del equipo. Por ejemplo, puede mencionarse el uso de tiempos máximos (Timebox) para las reuniones de trabajo.

2. *¿Considera Ud. que el TIPO DE COMUNICACIÓN "cara a cara" tiene influencia para un mejor desempeño de actividades del equipo? \**



<p>Entiéndase por comunicación "cara a cara" cuando dos o más personas están discutiendo o debatiendo algún tema, sin necesidad de medios de comunicación alternativos, tales como teléfono, internet (chat), entre otros.</p>
<p>3. <i>¿Considera Ud. que el TIPO DE COMUNICACIÓN "osmótica" tiene influencia para un mejor desempeño de actividades del equipo? *</i></p> <p>Entiéndase por comunicación "osmótica" a la forma como la información fluye en el fondo auditivo de los miembros del equipo, de tal forma que ellos captan la información relevante, como si fuera por ósmosis. Esto se realiza normalmente por estar en la misma habitación. Así, cuando una persona hace una pregunta, otros en la habitación pueden sintonizar o desconectarse, contribuyendo a la discusión o continuando con su trabajo.</p>
<p>4. <i>¿Considera Ud. que la PARTICIPACIÓN DEL ROL Programador/Desarrollador/Analista de Sistemas/Analista de Software/Diseñador de Software/Analista de Calidad/Testeador en la toma de decisiones genera un mayor compromiso y ambiente propicio para el mejor desempeño de sus actividades? *</i></p> <p>Entiéndase por "participación del rol" cuando se les hace partícipe en los diferentes procesos del desarrollo de software, tales como estimación de tiempos, definición de arquitectura, alternativas de solución, entre otros.</p>
<p>5. <i>¿Considera Ud. que la HETEROGENEIDAD DEL EQUIPO ayuda a logro de objetivos del equipo de una manera más eficiente? *</i></p> <p>Entiéndase por "heterogeneidad del equipo" a la participación de roles con diferentes habilidades y capacidades en la ejecución de actividades. Ejemplo: Si se está realizando actividades de definición de requerimientos y elaboración del análisis, podría incluirse la participación del Programador y Analista de Calidad y no sólo del Analista Funcional.</p>
<p>6. <i>¿Considera Ud. que la definición del TAMAÑO DEL EQUIPO influye en la comunicación del equipo y en la consecución de un mejor desempeño de sus actividades? *</i></p> <p>Entiéndase por "tamaño del equipo" a la cantidad de personas que participan en un proyecto. Es necesario indicar que mientras más personas participen en un equipo, mayor es la cantidad de canales de comunicación.</p>
<p>7. <i>¿Considera Ud. que la COMPLEJIDAD DEL PRODUCTO influye en el desempeño y logro de objetivos del equipo? *</i></p>
<p>8. <i>¿Considera Ud. que el NIVEL DE EXPERIENCIA de los participantes ayuda a que el equipo sea más eficiente? *</i></p>

9. ¿Considera Ud. que el PROCESO de desarrollo utilizado en el proyecto, facilita e influye en las actividades del equipo? \*

Entiéndase por "proceso de desarrollo" al uso de metodologías ágiles (SCRUM, XP, TDD, entre otros) o metodologías tradicionales (RUP, entre otros)

10. ¿Considera Ud. que la CONFIANZA EN EL EQUIPO es un aspecto positivo que permite que los integrantes del equipo puedan desarrollar sus actividades de manera eficiente? \*

11. ¿Considera Ud. que la FRECUENCIA DE LAS REUNIONES de equipo mejora el entendimiento del proyecto y facilita el desarrollo de actividades del equipo? \*

Entiéndase como la aplicación de buenas prácticas en la ejecución de reuniones frecuentes para recibir retroalimentación.

### ***Sección 3: Otros Aspectos***

En esta sección se plantean preguntas generales que nos ayudarán a tener un mayor entendimiento del impacto de los factores en la productividad de los componentes de la fábrica de software.

1. *¿En qué componente de la Fábrica de Software considera Ud. que tiene mayor impacto cada uno de los siguientes factores? \**

	Análisis y Diseño de Software	Programación de Software	En Pruebas de Software
Uso de Reglas y Eventos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Uso de comunicación "cara a cara" en el equipo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Uso de comunicación osmótica en el equipo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Participación del rol (Programador/Desarrollador/Analista de	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Sistemas/Analista de Software/Diseñador de Software/Analista de Calidad/Testeador) en la toma de decisiones			
Heterogeneidad del Equipo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tamaño del Equipo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Complejidad del Producto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nivel de Experiencia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Proceso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confianza	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reuniones regulares, reuniones frecuentes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*De acuerdo a su experiencia, ¿qué otros factores relevantes considera Ud. que podrían afectar la productividad?*

# Anexo C: Resultados del Análisis Envoltorio de Datos (Software DEAP Versión 2.1)

## *Resultados Fase 1: Fábricas de Software*

Results from DEAP Version 2.1

\*\*\*\*\*

by Tim Coelli, CEPA

<http://www.uq.edu.au/economics/cepa>

Project: Nouveau projet

Model 1: First model

Input orientated DEA

Scale assumption: VRS

Single-stage DEA - residual slacks presented

### EFFICIENCY SUMMARY:

firm crste vrste scale

SF001	0.893	0.893	1.000	-
SF002	1.000	1.000	1.000	-
SF003	0.977	0.977	1.000	-
SF004	0.933	0.937	0.995	drs
SF005	0.996	0.996	1.000	-
SF006	1.000	1.000	1.000	-

mean 0.966 0.967 0.999

Note: crste = technical efficiency from CRS DEA

vrste = technical efficiency from VRS DEA

scale = scale efficiency = crste/vrste

Note also that all subsequent tables refer to VRS results

### SUMMARY OF OUTPUT SLACKS:

firm output:	Points
SF001	0.000
SF002	0.000

SF003	0.000
SF004	0.000
SF005	0.000
SF006	0.000

mean	0.000
------	-------

SUMMARY OF INPUT SLACKS:

firm input:	Effort	Cost
SF001	0.000	40.756
SF002	0.000	0.000
SF003	0.000	107.574
SF004	15.538	0.000
SF005	0.000	129.926
SF006	0.000	0.000

mean	2.590	46.376
------	-------	--------

SUMMARY OF PEERS:

firm peers:

SF001	SF006	SF002
SF002	SF002	
SF003	SF006	SF002
SF004	SF006	SF002
SF005	SF006	SF002
SF006	SF006	

SUMMARY OF PEER WEIGHTS:

(in same order as above)

firm peer weights:

SF001	0.344	0.656
SF002	1.000	
SF003	0.656	0.344
SF004	0.281	0.719
SF005	0.500	0.500
SF006	1.000	

PEER COUNT SUMMARY:

(i.e., no. times each firm is a peer for another)

firm peer count:

SF001	0
SF002	4
SF003	0

SF004 0  
 SF005 0  
 SF006 4

SUMMARY OF OUTPUT TARGETS:

firm output:	Points
SF001	1.700
SF002	0.600
SF003	2.700
SF004	1.500
SF005	2.200
SF006	3.800

SUMMARY OF INPUT TARGETS:

firm input:	Effort	Cost
SF001	48.959	1890.638
SF002	17.300	664.000
SF003	77.741	3005.762
SF004	43.203	1667.612
SF005	63.350	2448.200
SF006	109.400	4232.400

FIRM BY FIRM RESULTS:

Results for SF001:

Technical efficiency = 0.893

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	1.700	0.000	0.000	1.700
input Effort	54.800	-5.841	0.000	48.959
input Cost	2161.800	-230.406	-40.756	1890.638

LISTING OF PEERS:

peer	lambda	weight
SF006	0.344	
SF002	0.656	

Results for SF002:

Technical efficiency = 1.000

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	0.600	0.000	0.000	0.600

input Effort	17.300	0.000	0.000	17.300
input Cost	664.000	0.000	0.000	664.000

LISTING OF PEERS:

peer	lambda	weight
SF002		1.000

Results for SF003:

Technical efficiency = 0.977

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	2.700	0.000	0.000	2.700
input Effort	79.600	-1.859	0.000	77.741
input Cost	3187.800	-74.464	-107.574	3005.762

LISTING OF PEERS:

peer	lambda	weight
SF006		0.656
SF002		0.344

Results for SF004:

Technical efficiency = 0.937

Scale efficiency = 0.995 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	1.500	0.000	0.000	1.500
input Effort	62.700	-3.959	-15.538	43.203
input Cost	1780.000	-112.388	0.000	1667.612

LISTING OF PEERS:

peer	lambda	weight
SF006		0.281
SF002		0.719

Results for SF005:

Technical efficiency = 0.996

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	2.200	0.000	0.000	2.200
input Effort	63.600	-0.250	0.000	63.350
input Cost	2588.300	-10.174	-129.926	2448.200

LISTING OF PEERS:

peer	lambda	weight
SF006		0.500
SF002		0.500

Results for SF006:

Technical efficiency = 1.000

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output Points	3.800	0.000	0.000	3.800
input Effort	109.400	0.000	0.000	109.400
input Cost	4232.400	0.000	0.000	4232.400

LISTING OF PEERS:

peer	lambda	weight
SF006	1.000	

### ***Resultados Fase 2: Proyectos de Fábricas de Software***

Results from DEAP Version 2.1

\*\*\*\*\*

by Tim Coelli, CEPA

<http://www.uq.edu.au/economics/cepa>

Project: Software Factory Projects

Model 1: DEA VRS I

Input orientated DEA

Scale assumption: VRS

Slacks calculated using multi-stage method

EFFICIENCY SUMMARY:

firm	crste	vrste	scale
SP001	0.972	0.977	0.995 drs
SP002	0.618	0.840	0.736 irs
SP003	0.645	0.953	0.677 irs
SP004	0.657	0.971	0.676 irs
SP005	0.726	0.758	0.958 irs
SP006	0.199	0.732	0.271 irs
SP007	0.950	0.956	0.994 drs
SP008	0.837	0.874	0.957 drs
SP009	0.886	0.904	0.980 irs
SP010	0.750	0.757	0.991 drs
SP011	0.940	0.958	0.981 irs
SP012	0.931	0.979	0.952 irs
SP013	0.934	0.996	0.938 irs
SP014	0.800	0.809	0.989 drs



SP015 0.879 0.891 0.987 irs  
SP016 0.939 0.968 0.970 irs  
SP017 0.936 0.950 0.985 irs  
SP018 0.949 0.994 0.954 irs  
SP019 0.952 0.971 0.980 drs  
SP020 0.761 0.784 0.971 irs  
SP021 0.756 0.785 0.963 irs  
SP022 0.798 0.807 0.988 drs  
SP023 0.786 1.000 0.786 drs  
SP024 0.608 0.624 0.975 drs  
SP025 0.953 0.974 0.979 drs  
SP026 0.962 0.987 0.974 drs  
SP027 0.932 0.979 0.952 drs  
SP028 0.969 0.973 0.996 drs  
SP029 0.955 0.975 0.979 drs  
SP030 0.959 1.000 0.959 drs  
SP031 0.961 0.993 0.968 drs  
SP032 0.941 0.954 0.987 irs  
SP033 0.745 0.784 0.950 irs  
SP034 0.626 0.683 0.916 drs  
SP035 0.715 0.716 0.999 drs  
SP036 1.000 1.000 1.000 -  
SP037 0.939 0.958 0.980 drs  
SP038 0.932 0.985 0.947 drs  
SP039 0.950 0.989 0.961 drs  
SP040 0.955 0.979 0.975 irs  
SP041 0.959 0.963 0.996 irs  
SP042 0.762 0.936 0.815 irs  
SP043 0.822 0.865 0.951 irs  
SP044 0.788 0.802 0.983 drs  
SP045 0.748 0.784 0.955 drs  
SP046 1.000 1.000 1.000 -  
SP047 0.736 0.748 0.984 irs  
SP048 0.669 0.689 0.972 irs  
SP049 0.830 0.881 0.942 drs  
SP050 0.941 0.943 0.997 drs  
SP051 0.936 0.950 0.985 irs  
SP052 0.950 0.957 0.993 irs  
SP053 0.963 0.968 0.995 irs  
SP054 0.951 0.958 0.993 drs  
SP055 0.966 0.984 0.982 drs  
SP056 0.940 0.971 0.968 irs  
SP057 0.748 0.780 0.959 drs  
SP058 0.771 0.807 0.956 drs  
SP059 0.945 0.952 0.994 drs  
SP060 0.961 0.967 0.994 drs  
SP061 0.963 0.970 0.993 drs

SP062 0.950 0.955 0.995 drs  
SP063 0.786 0.801 0.982 drs  
SP064 0.942 0.975 0.966 drs  
SP065 0.942 1.000 0.942 irs  
SP066 0.947 0.954 0.992 drs  
SP067 0.961 0.998 0.963 irs  
SP068 0.964 0.988 0.975 irs  
SP069 0.934 0.950 0.984 drs  
SP070 0.949 0.975 0.973 irs  
SP071 0.928 0.966 0.962 irs  
SP072 0.939 0.974 0.964 irs  
SP073 0.939 0.951 0.987 drs  
SP074 0.937 0.945 0.991 irs  
SP075 0.943 0.988 0.954 irs  
SP076 0.928 0.940 0.987 irs  
SP077 0.928 0.993 0.935 drs  
SP078 0.931 0.954 0.975 drs  
SP079 1.000 1.000 1.000 -  
SP080 0.941 0.983 0.957 irs  
SP081 0.932 0.933 0.999 irs  
SP082 0.930 0.968 0.960 drs  
SP083 0.939 0.945 0.994 irs  
SP084 0.927 0.942 0.984 irs  
SP085 0.931 0.938 0.993 irs  
SP086 0.932 0.980 0.951 drs  
SP087 0.943 0.947 0.996 drs  
SP088 0.940 0.969 0.970 irs  
SP089 0.939 0.945 0.994 drs  
SP090 0.928 0.940 0.987 drs  
SP091 0.939 0.943 0.996 drs  
SP092 0.940 0.953 0.987 drs  
SP093 0.929 0.968 0.959 irs  
SP094 0.956 0.984 0.971 irs  
SP095 0.951 1.000 0.951 irs  
SP096 0.929 0.971 0.957 drs  
SP097 0.928 0.986 0.941 drs  
SP098 0.939 0.971 0.967 drs  
SP099 0.943 0.964 0.979 drs  
SP100 0.939 0.946 0.993 drs  
SP101 0.956 0.985 0.971 drs  
SP102 0.945 0.946 1.000 -  
SP103 0.946 0.950 0.996 drs  
SP104 0.952 0.956 0.996 irs  
SP105 0.942 0.994 0.948 irs  
SP106 0.945 0.980 0.965 drs  
SP107 0.950 0.951 0.999 irs  
SP108 0.939 0.992 0.946 drs

SP109 0.927 0.985 0.941 drs  
SP110 0.951 0.954 0.996 drs  
SP111 0.943 0.994 0.949 drs  
SP112 0.948 0.973 0.975 irs  
SP113 0.928 0.947 0.980 irs  
SP114 0.943 0.972 0.970 drs  
SP115 0.953 0.981 0.972 irs  
SP116 0.951 0.973 0.977 irs  
SP117 0.960 0.971 0.989 drs  
SP118 0.957 0.998 0.959 irs  
SP119 0.952 0.969 0.982 drs  
SP120 0.942 0.985 0.956 irs  
SP121 0.939 0.975 0.963 drs  
SP122 0.933 1.000 0.933 drs  
SP123 0.928 0.985 0.942 drs  
SP124 0.945 0.989 0.955 drs  
SP125 0.937 0.963 0.972 drs  
SP126 0.943 0.957 0.986 irs  
SP127 0.937 0.976 0.960 irs  
SP128 0.950 0.977 0.972 drs  
SP129 0.940 0.966 0.974 irs  
SP130 0.934 0.979 0.955 drs  
SP131 0.949 0.970 0.978 drs  
SP132 0.958 0.963 0.995 drs  
SP133 0.837 0.844 0.991 drs  
SP134 0.953 0.982 0.970 drs  
SP135 0.927 0.945 0.981 irs  
SP136 0.952 1.000 0.952 irs  
SP137 0.931 0.957 0.973 drs  
SP138 0.938 0.944 0.993 irs  
SP139 0.936 0.961 0.973 irs  
SP140 0.974 0.985 0.989 irs  
SP141 0.959 0.964 0.995 drs  
SP142 0.956 0.963 0.994 irs  
SP143 0.824 0.853 0.966 drs  
SP144 0.769 0.792 0.970 drs  
SP145 0.940 0.953 0.986 drs  
SP146 0.975 1.000 0.975 irs  
SP147 0.945 0.960 0.985 irs  
SP148 0.938 0.956 0.981 drs  
SP149 0.955 0.980 0.974 drs  
SP150 0.937 0.949 0.987 drs  
SP151 0.931 0.979 0.952 irs  
SP152 0.928 0.955 0.972 drs  
SP153 0.926 0.966 0.958 drs  
SP154 0.964 0.983 0.981 irs  
SP155 0.945 0.959 0.986 drs

SP156 0.950 0.951 0.999 drs  
 SP157 0.969 0.970 0.999 drs  
 SP158 0.958 0.962 0.996 drs  
 SP159 0.954 0.985 0.968 irs  
 SP160 0.946 0.994 0.952 drs

mean 0.906 0.940 0.963

Note: crste = technical efficiency from CRS DEA  
 vrste = technical efficiency from VRS DEA  
 scale = scale efficiency = crste/vrste

Note also that all subsequent tables refer to VRS results

#### SUMMARY OF OUTPUT SLACKS:

firm output:	FP
SP001	0.000
SP002	9.371
SP003	10.000
SP004	10.000
SP005	0.000
SP006	25.000
SP007	0.000
SP008	0.000
SP009	0.000
SP010	0.000
SP011	0.000
SP012	0.000
SP013	0.000
SP014	0.000
SP015	0.000
SP016	0.000
SP017	0.000
SP018	0.000
SP019	0.000
SP020	0.000
SP021	0.000
SP022	0.000
SP023	0.000
SP024	0.000
SP025	0.000
SP026	0.000
SP027	0.000
SP028	0.000
SP029	0.000
SP030	0.000

SP031	0.000
SP032	0.000
SP033	0.347
SP034	0.000
SP035	0.000
SP036	0.000
SP037	0.000
SP038	0.000
SP039	0.000
SP040	0.000
SP041	0.000
SP042	5.151
SP043	0.108
SP044	0.000
SP045	0.000
SP046	0.000
SP047	0.000
SP048	0.000
SP049	0.000
SP050	0.000
SP051	0.000
SP052	0.000
SP053	0.000
SP054	0.000
SP055	0.000
SP056	0.000
SP057	0.000
SP058	0.000
SP059	0.000
SP060	0.000
SP061	0.000
SP062	0.000
SP063	0.000
SP064	0.000
SP065	0.000
SP066	0.000
SP067	0.000
SP068	0.000
SP069	0.000
SP070	0.000
SP071	0.000
SP072	0.000
SP073	0.000
SP074	0.000
SP075	0.000
SP076	0.000
SP077	0.000

SP078	0.000
SP079	0.000
SP080	0.000
SP081	0.000
SP082	0.000
SP083	0.000
SP084	0.000
SP085	0.000
SP086	0.000
SP087	0.000
SP088	0.000
SP089	0.000
SP090	0.000
SP091	0.000
SP092	0.000
SP093	0.000
SP094	0.000
SP095	0.000
SP096	0.000
SP097	0.000
SP098	0.000
SP099	0.000
SP100	0.000
SP101	0.000
SP102	0.000
SP103	0.000
SP104	0.000
SP105	0.000
SP106	0.000
SP107	0.000
SP108	0.000
SP109	0.000
SP110	0.000
SP111	0.000
SP112	0.000
SP113	0.000
SP114	0.000
SP115	0.000
SP116	0.000
SP117	0.000
SP118	0.000
SP119	0.000
SP120	0.000
SP121	0.000
SP122	0.000
SP123	0.000
SP124	0.000

SP125	0.000
SP126	0.000
SP127	0.000
SP128	0.000
SP129	0.000
SP130	0.000
SP131	0.000
SP132	0.000
SP133	0.000
SP134	0.000
SP135	0.000
SP136	0.000
SP137	0.000
SP138	0.000
SP139	0.000
SP140	0.000
SP141	0.000
SP142	0.000
SP143	0.000
SP144	0.000
SP145	0.000
SP146	0.000
SP147	0.000
SP148	0.000
SP149	0.000
SP150	0.000
SP151	0.000
SP152	0.000
SP153	0.000
SP154	0.000
SP155	0.000
SP156	0.000
SP157	0.000
SP158	0.000
SP159	0.000
SP160	0.000

mean            0.375

SUMMARY OF INPUT SLACKS:

firm input:	TE	PC
SP001	0.000	0.000
SP002	0.000	0.000
SP003	0.000	0.000
SP004	0.000	0.000
SP005	0.000	0.000

SP006	0.000	0.000
SP007	0.000	0.000
SP008	0.000	18840.449
SP009	0.000	4940.574
SP010	0.000	0.000
SP011	0.000	8156.043
SP012	0.000	2135.126
SP013	0.000	3896.474
SP014	0.000	0.000
SP015	0.000	3656.977
SP016	0.000	8933.327
SP017	0.000	11723.174
SP018	0.000	6980.434
SP019	0.000	0.000
SP020	0.000	0.000
SP021	0.000	0.000
SP022	0.000	0.000
SP023	0.000	0.000
SP024	0.000	0.000
SP025	0.000	0.000
SP026	0.000	0.000
SP027	0.000	26547.466
SP028	0.000	0.000
SP029	0.000	0.000
SP030	0.000	0.000
SP031	0.000	0.000
SP032	0.000	6535.021
SP033	0.000	0.000
SP034	0.000	0.000
SP035	0.000	0.000
SP036	0.000	0.000
SP037	0.000	0.000
SP038	0.000	1356.737
SP039	0.000	0.000
SP040	0.000	0.000
SP041	0.000	0.000
SP042	0.000	0.000
SP043	0.000	0.000
SP044	0.000	0.000
SP045	0.000	0.000
SP046	0.000	0.000
SP047	0.000	0.000
SP048	0.000	0.000
SP049	0.000	0.000
SP050	0.000	0.000
SP051	0.000	0.000
SP052	0.000	0.000



SP053	0.000	0.000
SP054	0.000	0.000
SP055	0.000	0.000
SP056	0.000	0.000
SP057	0.000	0.000
SP058	0.000	2460.387
SP059	0.000	0.000
SP060	0.000	0.000
SP061	0.000	0.000
SP062	0.000	0.000
SP063	0.000	0.000
SP064	0.000	0.000
SP065	0.000	0.000
SP066	0.000	0.000
SP067	0.000	0.000
SP068	0.000	0.000
SP069	0.000	0.000
SP070	0.000	0.000
SP071	0.000	1624.013
SP072	0.000	8096.369
SP073	0.000	0.000
SP074	0.000	0.000
SP075	0.000	2060.604
SP076	0.000	2327.127
SP077	0.000	33203.359
SP078	0.000	5276.320
SP079	0.000	0.000
SP080	0.000	98.902
SP081	0.000	3256.336
SP082	0.000	10589.544
SP083	0.000	0.000
SP084	0.000	2064.798
SP085	0.000	0.000
SP086	0.000	9602.739
SP087	0.000	0.000
SP088	0.000	0.000
SP089	0.000	0.000
SP090	0.000	8263.235
SP091	0.000	0.000
SP092	0.000	0.000
SP093	0.000	3435.189
SP094	0.000	0.000
SP095	0.000	0.000
SP096	0.000	17886.448
SP097	0.000	23640.601
SP098	0.000	0.000
SP099	0.000	0.000

SP100	0.000	0.000
SP101	0.000	0.000
SP102	0.000	0.000
SP103	0.000	0.000
SP104	0.000	0.000
SP105	0.000	0.000
SP106	0.000	0.000
SP107	0.000	0.000
SP108	0.000	0.000
SP109	0.000	8528.159
SP110	0.000	0.000
SP111	0.000	0.000
SP112	0.000	0.000
SP113	0.000	0.000
SP114	0.000	0.000
SP115	0.000	0.000
SP116	0.000	0.000
SP117	0.000	0.000
SP118	0.000	0.000
SP119	0.000	0.000
SP120	0.000	0.000
SP121	0.000	0.000
SP122	0.000	0.000
SP123	0.000	6360.008
SP124	0.000	0.000
SP125	0.000	0.000
SP126	0.000	0.000
SP127	0.000	0.000
SP128	0.000	0.000
SP129	0.000	0.000
SP130	0.000	0.000
SP131	0.000	0.000
SP132	0.000	0.000
SP133	0.000	0.000
SP134	0.000	0.000
SP135	0.000	2585.111
SP136	0.000	0.000
SP137	0.000	7599.755
SP138	0.000	7587.770
SP139	0.000	4724.389
SP140	0.000	0.000
SP141	0.000	0.000
SP142	0.000	0.000
SP143	0.000	0.000
SP144	0.000	0.000
SP145	0.000	0.000
SP146	0.000	0.000

SP147	0.000	0.000
SP148	0.000	0.000
SP149	0.000	0.000
SP150	0.000	15392.451
SP151	0.000	2908.204
SP152	0.000	7820.877
SP153	0.000	5126.714
SP154	0.000	0.000
SP155	0.000	0.000
SP156	0.000	0.000
SP157	0.000	0.000
SP158	0.000	0.000
SP159	0.000	0.000
SP160	0.000	0.000
mean	0.000	1913.883

SUMMARY OF PEERS:

firm peers:

SP001 SP079 SP030 SP036  
 SP002 SP146 SP079  
 SP003 SP095 SP065  
 SP004 SP065 SP095  
 SP005 SP146 SP136  
 SP006 SP095 SP065  
 SP007 SP079 SP030 SP036  
 SP008 SP122 SP036  
 SP009 SP036 SP136  
 SP010 SP030 SP036 SP079  
 SP011 SP036 SP136  
 SP012 SP136  
 SP013 SP065  
 SP014 SP030 SP036 SP079  
 SP015 SP036 SP136  
 SP016 SP036 SP136  
 SP017 SP036 SP136  
 SP018 SP036 SP136  
 SP019 SP079 SP030 SP036  
 SP020 SP146 SP136  
 SP021 SP036 SP136 SP146  
 SP022 SP079 SP030 SP036  
 SP023 SP023  
 SP024 SP079 SP046 SP030  
 SP025 SP079 SP030 SP036  
 SP026 SP079 SP030 SP036  
 SP027 SP122 SP036

SP028	SP079	SP030	SP036
SP029	SP030	SP036	SP079
SP030	SP030		
SP031	SP030	SP036	SP079
SP032	SP036	SP136	
SP033	SP146	SP095	
SP034	SP030	SP046	SP122
SP035	SP030	SP036	SP079
SP036	SP036		
SP037	SP030	SP122	SP036
SP038	SP122	SP036	
SP039	SP030	SP122	SP036
SP040	SP146	SP036	SP136
SP041	SP079	SP146	SP036
SP042	SP146	SP095	
SP043	SP146	SP095	
SP044	SP036	SP079	SP030
SP045	SP046	SP030	SP079
SP046	SP046		
SP047	SP136	SP146	SP036
SP048	SP146	SP036	SP136
SP049	SP079	SP046	SP030
SP050	SP079	SP030	SP036
SP051	SP036	SP136	SP146
SP052	SP146	SP036	SP136
SP053	SP079	SP036	SP146
SP054	SP030	SP036	SP079
SP055	SP030	SP036	SP079
SP056	SP146	SP036	SP136
SP057	SP030	SP122	SP036
SP058	SP036	SP122	
SP059	SP079	SP030	SP036
SP060	SP079	SP030	SP036
SP061	SP030	SP036	SP079
SP062	SP079	SP030	SP036
SP063	SP030	SP122	SP036
SP064	SP036	SP030	SP122
SP065	SP065		
SP066	SP079	SP030	SP036
SP067	SP146	SP036	SP136
SP068	SP146	SP036	SP136
SP069	SP036	SP122	SP030
SP070	SP146	SP036	SP136
SP071	SP036	SP136	
SP072	SP036	SP136	
SP073	SP030	SP122	SP036
SP074	SP036	SP136	SP146

SP075 SP036 SP136  
SP076 SP036 SP136  
SP077 SP122 SP036  
SP078 SP036 SP122  
SP079 SP079  
SP080 SP036 SP136  
SP081 SP036 SP136  
SP082 SP036 SP122  
SP083 SP036 SP136 SP146  
SP084 SP036 SP136  
SP085 SP146 SP036 SP136  
SP086 SP036 SP122  
SP087 SP079 SP030 SP036  
SP088 SP146 SP036 SP136  
SP089 SP036 SP079 SP030  
SP090 SP036 SP122  
SP091 SP036 SP079 SP030  
SP092 SP079 SP030 SP036  
SP093 SP036 SP136  
SP094 SP136 SP146 SP036  
SP095 SP095  
SP096 SP122 SP036  
SP097 SP122 SP036  
SP098 SP030 SP122 SP036  
SP099 SP030 SP122 SP036  
SP100 SP079 SP030 SP036  
SP101 SP079 SP030 SP036  
SP102 SP079 SP146 SP036  
SP103 SP079 SP030 SP036  
SP104 SP146 SP079 SP036  
SP105 SP065 SP095  
SP106 SP030 SP122 SP036  
SP107 SP146 SP079 SP036  
SP108 SP036 SP030 SP122  
SP109 SP036 SP122  
SP110 SP079 SP030 SP036  
SP111 SP036 SP030 SP122  
SP112 SP146 SP036 SP136  
SP113 SP146 SP036 SP136  
SP114 SP036 SP030 SP122  
SP115 SP136 SP146 SP036  
SP116 SP136 SP146 SP036  
SP117 SP030 SP036 SP079  
SP118 SP136 SP146 SP036  
SP119 SP030 SP036 SP079  
SP120 SP146 SP136  
SP121 SP030 SP122 SP036

SP122 SP122  
 SP123 SP036 SP122  
 SP124 SP030 SP122 SP036  
 SP125 SP122 SP036 SP030  
 SP126 SP136 SP146 SP036  
 SP127 SP146 SP036 SP136  
 SP128 SP030 SP122 SP036  
 SP129 SP136 SP146 SP036  
 SP130 SP030 SP122 SP036  
 SP131 SP030 SP122 SP036  
 SP132 SP030 SP036 SP079  
 SP133 SP030 SP036 SP079  
 SP134 SP030 SP036 SP079  
 SP135 SP036 SP136  
 SP136 SP136  
 SP137 SP036 SP122  
 SP138 SP036 SP136  
 SP139 SP036 SP136  
 SP140 SP079 SP036 SP146  
 SP141 SP030 SP036 SP079  
 SP142 SP079 SP146 SP036  
 SP143 SP036 SP030 SP122  
 SP144 SP079 SP030 SP036  
 SP145 SP122 SP036 SP030  
 SP146 SP146  
 SP147 SP036 SP136 SP146  
 SP148 SP030 SP122 SP036  
 SP149 SP079 SP030 SP036  
 SP150 SP036 SP122  
 SP151 SP136  
 SP152 SP036 SP122  
 SP153 SP036 SP122  
 SP154 SP136 SP146 SP036  
 SP155 SP079 SP030 SP036  
 SP156 SP079 SP030 SP036  
 SP157 SP079 SP030 SP036  
 SP158 SP079 SP030 SP036  
 SP159 SP146 SP136  
 SP160 SP036 SP030 SP122

**SUMMARY OF PEER WEIGHTS:**  
 (in same order as above)

firm peer weights:  
 SP001 0.848 0.029 0.123  
 SP002 0.726 0.274  
 SP003 0.604 0.396

SP004 0.473 0.527  
SP005 0.455 0.545  
SP006 0.861 0.139  
SP007 0.207 0.050 0.743  
SP008 0.243 0.757  
SP009 0.438 0.563  
SP010 0.066 0.389 0.545  
SP011 0.469 0.531  
SP012 1.000  
SP013 1.000  
SP014 0.086 0.382 0.532  
SP015 0.594 0.406  
SP016 0.250 0.750  
SP017 0.562 0.437  
SP018 0.031 0.969  
SP019 0.193 0.214 0.593  
SP020 0.863 0.137  
SP021 0.125 0.611 0.264  
SP022 0.280 0.103 0.618  
SP023 1.000  
SP024 0.815 0.122 0.064  
SP025 0.386 0.215 0.399  
SP026 0.359 0.308 0.333  
SP027 0.299 0.701  
SP028 0.746 0.024 0.231  
SP029 0.216 0.468 0.316  
SP030 1.000  
SP031 0.458 0.157 0.386  
SP032 0.594 0.406  
SP033 0.347 0.653  
SP034 0.207 0.630 0.163  
SP035 0.010 0.832 0.159  
SP036 1.000  
SP037 0.150 0.029 0.821  
SP038 0.449 0.551  
SP039 0.462 0.112 0.426  
SP040 0.695 0.125 0.180  
SP041 0.389 0.234 0.377  
SP042 0.151 0.849  
SP043 0.108 0.892  
SP044 0.196 0.668 0.137  
SP045 0.128 0.141 0.731  
SP046 1.000  
SP047 0.176 0.449 0.375  
SP048 0.654 0.094 0.252  
SP049 0.695 0.100 0.205  
SP050 0.357 0.020 0.623

SP051 0.500 0.279 0.221  
SP052 0.380 0.594 0.027  
SP053 0.523 0.231 0.247  
SP054 0.047 0.424 0.528  
SP055 0.160 0.314 0.525  
SP056 0.359 0.156 0.485  
SP057 0.079 0.246 0.675  
SP058 0.734 0.266  
SP059 0.084 0.055 0.861  
SP060 0.460 0.041 0.499  
SP061 0.048 0.385 0.567  
SP062 0.513 0.037 0.450  
SP063 0.126 0.030 0.844  
SP064 0.615 0.285 0.100  
SP065 1.000  
SP066 0.230 0.062 0.708  
SP067 0.441 0.062 0.496  
SP068 0.771 0.094 0.135  
SP069 0.899 0.045 0.057  
SP070 0.402 0.219 0.379  
SP071 0.125 0.875  
SP072 0.156 0.844  
SP073 0.115 0.004 0.881  
SP074 0.719 0.203 0.078  
SP075 0.031 0.969  
SP076 0.594 0.406  
SP077 0.640 0.360  
SP078 0.897 0.103  
SP079 1.000  
SP080 0.062 0.938  
SP081 0.969 0.031  
SP082 0.790 0.210  
SP083 0.781 0.156 0.063  
SP084 0.531 0.469  
SP085 0.030 0.781 0.189  
SP086 0.682 0.318  
SP087 0.165 0.034 0.802  
SP088 0.300 0.219 0.481  
SP089 0.913 0.035 0.053  
SP090 0.953 0.047  
SP091 0.962 0.007 0.032  
SP092 0.027 0.127 0.846  
SP093 0.094 0.906  
SP094 0.354 0.490 0.156  
SP095 1.000  
SP096 0.243 0.757  
SP097 0.491 0.509



SP098 0.138 0.139 0.723  
SP099 0.197 0.023 0.780  
SP100 0.201 0.056 0.743  
SP101 0.244 0.383 0.372  
SP102 0.391 0.014 0.595  
SP103 0.467 0.027 0.506  
SP104 0.227 0.250 0.523  
SP105 0.348 0.652  
SP106 0.204 0.141 0.656  
SP107 0.051 0.236 0.713  
SP108 0.398 0.131 0.471  
SP109 0.523 0.477  
SP110 0.292 0.029 0.678  
SP111 0.260 0.383 0.357  
SP112 0.230 0.313 0.457  
SP113 0.162 0.438 0.401  
SP114 0.701 0.219 0.080  
SP115 0.394 0.418 0.187  
SP116 0.070 0.836 0.094  
SP117 0.082 0.407 0.512  
SP118 0.561 0.408 0.031  
SP119 0.177 0.479 0.345  
SP120 0.399 0.601  
SP121 0.119 0.184 0.697  
SP122 1.000  
SP123 0.537 0.463  
SP124 0.342 0.229 0.430  
SP125 0.113 0.820 0.066  
SP126 0.138 0.455 0.406  
SP127 0.440 0.031 0.529  
SP128 0.362 0.010 0.628  
SP129 0.356 0.425 0.219  
SP130 0.105 0.300 0.595  
SP131 0.249 0.005 0.746  
SP132 0.031 0.411 0.558  
SP133 0.066 0.497 0.436  
SP134 0.436 0.430 0.134  
SP135 0.469 0.531  
SP136 1.000  
SP137 0.883 0.117  
SP138 0.781 0.219  
SP139 0.313 0.687  
SP140 0.443 0.056 0.502  
SP141 0.034 0.317 0.649  
SP142 0.294 0.346 0.360  
SP143 0.564 0.357 0.079  
SP144 0.049 0.442 0.510

SP145	0.032	0.906	0.062
SP146	1.000		
SP147	0.469	0.253	0.278
SP148	0.096	0.044	0.861
SP149	0.123	0.328	0.549
SP150	0.953	0.047	
SP151	1.000		
SP152	0.879	0.121	
SP153	0.771	0.229	
SP154	0.103	0.678	0.219
SP155	0.101	0.142	0.757
SP156	0.433	0.005	0.562
SP157	0.608	0.003	0.389
SP158	0.490	0.025	0.485
SP159	0.780	0.220	
SP160	0.212	0.532	0.256

PEER COUNT SUMMARY:

(i.e., no. times each firm is a peer for another)

firm peer count:

SP001	0
SP002	0
SP003	0
SP004	0
SP005	0
SP006	0
SP007	0
SP008	0
SP009	0
SP010	0
SP011	0
SP012	0
SP013	0
SP014	0
SP015	0
SP016	0
SP017	0
SP018	0
SP019	0
SP020	0
SP021	0
SP022	0
SP023	0
SP024	0
SP025	0
SP026	0

SP027	0
SP028	0
SP029	0
SP030	68
SP031	0
SP032	0
SP033	0
SP034	0
SP035	0
SP036	131
SP037	0
SP038	0
SP039	0
SP040	0
SP041	0
SP042	0
SP043	0
SP044	0
SP045	0
SP046	4
SP047	0
SP048	0
SP049	0
SP050	0
SP051	0
SP052	0
SP053	0
SP054	0
SP055	0
SP056	0
SP057	0
SP058	0
SP059	0
SP060	0
SP061	0
SP062	0
SP063	0
SP064	0
SP065	5
SP066	0
SP067	0
SP068	0
SP069	0
SP070	0
SP071	0
SP072	0
SP073	0

SP074	0
SP075	0
SP076	0
SP077	0
SP078	0
SP079	52
SP080	0
SP081	0
SP082	0
SP083	0
SP084	0
SP085	0
SP086	0
SP087	0
SP088	0
SP089	0
SP090	0
SP091	0
SP092	0
SP093	0
SP094	0
SP095	7
SP096	0
SP097	0
SP098	0
SP099	0
SP100	0
SP101	0
SP102	0
SP103	0
SP104	0
SP105	0
SP106	0
SP107	0
SP108	0
SP109	0
SP110	0
SP111	0
SP112	0
SP113	0
SP114	0
SP115	0
SP116	0
SP117	0
SP118	0
SP119	0
SP120	0

SP121	0
SP122	41
SP123	0
SP124	0
SP125	0
SP126	0
SP127	0
SP128	0
SP129	0
SP130	0
SP131	0
SP132	0
SP133	0
SP134	0
SP135	0
SP136	49
SP137	0
SP138	0
SP139	0
SP140	0
SP141	0
SP142	0
SP143	0
SP144	0
SP145	0
SP146	40
SP147	0
SP148	0
SP149	0
SP150	0
SP151	0
SP152	0
SP153	0
SP154	0
SP155	0
SP156	0
SP157	0
SP158	0
SP159	0
SP160	0

**SUMMARY OF OUTPUT TARGETS:**

firm output:	FP
SP001	49.000
SP002	37.371
SP003	35.000

SP004	35.000
SP005	36.000
SP006	35.000
SP007	69.000
SP008	120.000
SP009	50.000
SP010	62.000
SP011	51.000
SP012	36.000
SP013	35.000
SP014	65.000
SP015	55.000
SP016	44.000
SP017	54.000
SP018	37.000
SP019	91.000
SP020	36.000
SP021	40.000
SP022	74.000
SP023	270.000
SP024	65.000
SP025	86.000
SP026	99.000
SP027	132.000
SP028	51.000
SP029	88.000
SP030	200.000
SP031	118.000
SP032	55.000
SP033	35.347
SP034	185.000
SP035	65.000
SP036	68.000
SP037	94.000
SP038	164.000
SP039	153.000
SP040	40.000
SP041	50.000
SP042	35.151
SP043	35.108
SP044	68.000
SP045	78.000
SP046	155.000
SP047	48.000
SP048	39.000
SP049	85.000
SP050	61.000

SP051	52.000
SP052	55.000
SP053	46.000
SP054	60.000
SP055	75.000
SP056	41.000
SP057	131.000
SP058	125.000
SP059	73.000
SP060	61.000
SP061	59.000
SP062	59.000
SP063	91.000
SP064	127.000
SP065	35.000
SP066	70.000
SP067	38.000
SP068	39.000
SP069	85.000
SP070	43.000
SP071	40.000
SP072	41.000
SP073	84.000
SP074	59.000
SP075	37.000
SP076	55.000
SP077	205.000
SP078	90.000
SP079	41.000
SP080	38.000
SP081	67.000
SP082	113.000
SP083	61.000
SP084	53.000
SP085	61.000
SP086	136.000
SP087	68.000
SP088	43.000
SP089	74.000
SP090	78.000
SP091	72.000
SP092	84.000
SP093	39.000
SP094	41.000
SP095	35.000
SP096	120.000
SP097	173.000

SP098	116.000
SP099	99.000
SP100	70.000
SP101	112.000
SP102	57.000
SP103	59.000
SP104	54.000
SP105	35.000
SP106	125.000
SP107	60.000
SP108	186.000
SP109	170.000
SP110	64.000
SP111	195.000
SP112	46.000
SP113	50.000
SP114	114.000
SP115	42.000
SP116	39.000
SP117	65.000
SP118	37.000
SP119	82.000
SP120	36.000
SP121	123.000
SP122	282.000
SP123	167.000
SP124	162.000
SP125	101.000
SP126	49.000
SP127	37.000
SP128	118.000
SP129	43.000
SP130	146.000
SP131	102.000
SP132	57.000
SP133	65.000
SP134	122.000
SP135	51.000
SP136	36.000
SP137	93.000
SP138	61.000
SP139	46.000
SP140	40.000
SP141	55.000
SP142	49.000
SP143	132.000
SP144	125.000



SP145	83.000
SP146	36.000
SP147	51.000
SP148	90.000
SP149	108.000
SP150	78.000
SP151	36.000
SP152	94.000
SP153	117.000
SP154	43.000
SP155	84.000
SP156	57.000
SP157	52.000
SP158	58.000
SP159	36.000
SP160	193.000

SUMMARY OF INPUT TARGETS:

firm input:	TE	PC
SP001	1395.139	44230.272
SP002	1074.702	34591.220
SP003	1013.420	39215.362
SP004	1013.238	39409.559
SP005	1030.524	39974.541
SP006	1014.023	38568.807
SP007	1904.750	73638.819
SP008	3399.771	137337.752
SP009	1383.397	58246.071
SP010	1743.430	61457.125
SP011	1408.961	59239.441
SP012	1025.500	44338.900
SP013	1012.000	40736.840
SP014	1832.293	64491.849
SP015	1511.217	63212.918
SP016	1230.012	52285.855
SP017	1485.653	62219.549
SP018	1051.064	45332.269
SP019	2568.792	94522.155
SP020	1035.039	36053.057
SP021	1130.676	45776.398
SP022	2067.039	77125.022
SP023	13524.050	235557.620
SP024	2026.326	53538.798
SP025	2446.261	86630.070
SP026	2835.761	99290.268
SP027	3758.900	151463.375

SP028	1440.393	47803.506
SP029	2496.323	89670.611
SP030	5840.000	200074.580
SP031	3415.065	116681.617
SP032	1511.217	63212.918
SP033	1022.058	37010.836
SP034	6159.992	161812.226
SP035	1776.451	70680.103
SP036	1843.550	76126.720
SP037	2628.547	102030.546
SP038	4716.574	189131.702
SP039	4408.634	161668.284
SP040	1135.433	41643.753
SP041	1394.685	50121.057
SP042	1017.707	37692.483
SP043	1016.750	37842.374
SP044	1943.414	65060.124
SP045	2415.338	66803.814
SP046	5726.020	106266.170
SP047	1337.234	51946.009
SP048	1109.419	41041.324
SP049	2587.389	75400.305
SP050	1684.989	63639.250
SP051	1436.972	58107.226
SP052	1515.413	59568.689
SP053	1295.556	44009.275
SP054	1680.382	59855.635
SP055	2134.079	73990.003
SP056	1157.288	45859.080
SP057	3732.593	147813.995
SP058	3549.408	143223.428
SP059	2007.529	79425.640
SP060	1700.420	61929.154
SP061	1655.922	58272.082
SP062	1647.783	59174.011
SP063	2537.672	99245.559
SP064	3622.407	136603.947
SP065	1012.000	40736.840
SP066	1938.528	74189.766
SP067	1081.505	42089.316
SP068	1110.714	39916.263
SP069	2354.919	94359.955
SP070	1208.891	47433.604
SP071	1127.756	48312.378
SP072	1153.320	49305.747
SP073	2327.702	91332.698
SP074	1614.339	66434.936

SP075	1051.064	45332.269
SP076	1511.217	63212.918
SP077	5943.595	237394.247
SP078	2501.951	102023.695
SP079	1175.690	34196.030
SP080	1076.628	46325.639
SP081	1817.986	75133.351
SP082	3190.280	129097.806
SP083	1665.298	68568.291
SP084	1460.089	61226.179
SP085	1664.932	68886.451
SP086	3878.609	156171.916
SP087	1868.161	73399.413
SP088	1207.768	48409.086
SP089	2030.420	81183.042
SP090	2142.823	87898.072
SP091	1965.664	79770.863
SP092	2332.066	90696.613
SP093	1102.192	47319.008
SP094	1158.736	44601.415
SP095	1014.350	38218.370
SP096	3399.771	137337.752
SP097	4985.920	199725.920
SP098	3286.439	128275.834
SP099	2780.386	106413.073
SP100	1934.125	74677.739
SP101	3212.251	113392.180
SP102	1571.242	59160.349
SP103	1640.860	59941.231
SP104	1493.745	56269.781
SP105	1013.533	39094.441
SP106	3558.790	136817.980
SP107	1644.879	64125.583
SP108	5381.008	210909.980
SP109	4896.138	196194.514
SP110	1766.147	67527.878
SP111	5661.951	213586.835
SP112	1283.683	52064.143
SP113	1385.184	56693.897
SP114	3230.312	123376.544
SP115	1183.504	46286.161
SP116	1111.435	39290.257
SP117	1829.247	64829.440
SP118	1055.571	41417.728
SP119	2318.984	83558.130
SP120	1029.907	40510.524
SP121	3495.023	137135.210

SP122	8248.000	328033.660
SP123	4806.356	192663.108
SP124	4672.444	176041.799
SP125	2834.202	112889.813
SP126	1362.866	52881.048
SP127	1055.925	41109.691
SP128	3356.576	123609.604
SP129	1209.143	47214.420
SP130	4182.713	164645.084
SP131	2872.546	108320.581
SP132	1594.051	56540.309
SP133	1817.964	66079.726
SP134	3498.466	124615.765
SP135	1408.961	59239.441
SP136	1025.500	44338.900
SP137	2591.733	105555.101
SP138	1664.602	69173.134
SP139	1281.141	54272.594
SP140	1143.198	36811.154
SP141	1547.024	53163.488
SP142	1368.231	49491.685
SP143	3775.343	140237.089
SP144	3576.572	128842.686
SP145	2295.308	91839.379
SP146	1036.550	34740.520
SP147	1412.033	56570.871
SP148	2506.351	99019.974
SP149	3072.937	111654.071
SP150	2142.823	87898.072
SP151	1025.500	44338.900
SP152	2621.661	106732.236
SP153	3309.989	133806.346
SP154	1211.941	44784.580
SP155	2343.144	89468.942
SP156	1575.241	58624.699
SP157	1450.115	51020.282
SP158	1614.146	58607.461
SP159	1034.122	36849.625
SP160	5608.956	206549.099

FIRM BY FIRM RESULTS:

Results for SP001:

Technical efficiency = 0.977

Scale efficiency = 0.995 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
----------	----------	--------	-------	-----------

	value	movement	movement	value
output FP	49.000	0.000	0.000	49.000
input TE	1427.840	-32.701	0.000	1395.139
input PC	45267.010	-1036.738	0.000	44230.272

LISTING OF PEERS:

peer	lambda	weight
SP079	0.848	
SP030	0.029	
SP036	0.123	

Results for SP002:

Technical efficiency = 0.840

Scale efficiency = 0.736 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	28.000	0.000	9.371	37.371
input TE	1279.900	-205.198	0.000	1074.702
input PC	41195.870	-6604.650	0.000	34591.220

LISTING OF PEERS:

peer	lambda	weight
SP146	0.726	
SP079	0.274	

Results for SP003:

Technical efficiency = 0.953

Scale efficiency = 0.677 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	25.000	0.000	10.000	35.000
input TE	1063.740	-50.320	0.000	1013.420
input PC	41162.560	-1947.198	0.000	39215.362

LISTING OF PEERS:

peer	lambda	weight
SP095	0.604	
SP065	0.396	

Results for SP004:

Technical efficiency = 0.971

Scale efficiency = 0.676 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	25.000	0.000	10.000	35.000
input TE	1043.020	-29.782	0.000	1013.238
input PC	40567.900	-1158.341	0.000	39409.559

LISTING OF PEERS:

peer lambda weight  
SP065 0.473  
SP095 0.527

Results for SP005:

Technical efficiency = 0.758

Scale efficiency = 0.958 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1358.800	-328.276	0.000	1030.524
input PC	52708.510	-12733.969	0.000	39974.541

LISTING OF PEERS:

peer lambda weight  
SP146 0.455  
SP136 0.545

Results for SP006:

Technical efficiency = 0.732

Scale efficiency = 0.271 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	10.000	0.000	25.000	35.000
input TE	1385.020	-370.997	0.000	1014.023
input PC	52679.840	-14111.033	0.000	38568.807

LISTING OF PEERS:

peer lambda weight  
SP095 0.861  
SP065 0.139

Results for SP007:

Technical efficiency = 0.956

Scale efficiency = 0.994 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	69.000	0.000	0.000	69.000
input TE	1991.800	-87.050	0.000	1904.750
input PC	77004.220	-3365.401	0.000	73638.819

LISTING OF PEERS:

peer lambda weight  
SP079 0.207  
SP030 0.050  
SP036 0.743

Results for SP008:

Technical efficiency = 0.874

Scale efficiency = 0.957 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	120.000	0.000	0.000	120.000
input TE	3888.000	-488.229	0.000	3399.771
input PC	178606.370	-22428.169	-18840.449	137337.752

LISTING OF PEERS:

peer	lambda	weight
SP122	0.243	
SP036	0.757	

Results for SP009:

Technical efficiency = 0.904

Scale efficiency = 0.980 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	50.000	0.000	0.000	50.000
input TE	1530.500	-147.103	0.000	1383.397
input PC	69905.580	-6718.935	-4940.574	58246.071

LISTING OF PEERS:

peer	lambda	weight
SP036	0.438	
SP136	0.563	

Results for SP010:

Technical efficiency = 0.757

Scale efficiency = 0.991 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	62.000	0.000	0.000	62.000
input TE	2304.000	-560.570	0.000	1743.430
input PC	81217.630	-19760.505	0.000	61457.125

LISTING OF PEERS:

peer	lambda	weight
SP030	0.066	
SP036	0.389	
SP079	0.545	

Results for SP011:

Technical efficiency = 0.958

Scale efficiency = 0.981 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	51.000	0.000	0.000	51.000
input TE	1470.500	-61.539	0.000	1408.961
input PC	70339.110	-2943.627	-8156.043	59239.441

LISTING OF PEERS:

peer	lambda	weight
SP036	0.469	
SP136	0.531	

Results for SP012:

Technical efficiency = 0.979

Scale efficiency = 0.952 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1048.000	-22.500	0.000	1025.500
input PC	47493.690	-1019.664	-2135.126	44338.900

LISTING OF PEERS:

peer	lambda	weight
SP136	1.000	

Results for SP013:

Technical efficiency = 0.996

Scale efficiency = 0.938 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.000	35.000
input TE	1016.000	-4.000	0.000	1012.000
input PC	44809.730	-176.416	-3896.474	40736.840

LISTING OF PEERS:

peer	lambda	weight
SP065	1.000	

Results for SP014:

Technical efficiency = 0.809

Scale efficiency = 0.989 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	65.000	0.000	0.000	65.000
input TE	2264.000	-431.707	0.000	1832.293
input PC	79686.780	-15194.931	0.000	64491.849

LISTING OF PEERS:



peer lambda weight  
 SP030 0.086  
 SP036 0.382  
 SP079 0.532

Results for SP015:

Technical efficiency = 0.891

Scale efficiency = 0.987 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	55.000	0.000	0.000	55.000
input TE	1696.500	-185.283	0.000	1511.217
input PC	75068.480	-8198.585	-3656.977	63212.918

LISTING OF PEERS:

peer lambda weight  
 SP036 0.594  
 SP136 0.406

Results for SP016:

Technical efficiency = 0.968

Scale efficiency = 0.970 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	44.000	0.000	0.000	44.000
input TE	1270.250	-40.238	0.000	1230.012
input PC	63221.850	-2002.668	-8933.327	52285.855

LISTING OF PEERS:

peer lambda weight  
 SP036 0.250  
 SP136 0.750

Results for SP017:

Technical efficiency = 0.950

Scale efficiency = 0.985 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	54.000	0.000	0.000	54.000
input TE	1563.300	-77.647	0.000	1485.653
input PC	77807.300	-3864.577	-11723.174	62219.549

LISTING OF PEERS:

peer lambda weight  
 SP036 0.562  
 SP136 0.437

Results for SP018:

Technical efficiency = 0.994

Scale efficiency = 0.954 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	37.000	0.000	0.000	37.000
input TE	1057.400	-6.336	0.000	1051.064
input PC	52628.050	-315.347	-6980.434	45332.269

LISTING OF PEERS:

peer	lambda	weight
SP036	0.031	
SP136	0.969	

Results for SP019:

Technical efficiency = 0.971

Scale efficiency = 0.980 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	91.000	0.000	0.000	91.000
input TE	2645.780	-76.988	0.000	2568.792
input PC	97355.050	-2832.895	0.000	94522.155

LISTING OF PEERS:

peer	lambda	weight
SP079	0.193	
SP030	0.214	
SP036	0.593	

Results for SP020:

Technical efficiency = 0.784

Scale efficiency = 0.971 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1320.000	-284.961	0.000	1035.039
input PC	45978.980	-9925.923	0.000	36053.057

LISTING OF PEERS:

peer	lambda	weight
SP146	0.863	
SP136	0.137	

Results for SP021:

Technical efficiency = 0.785

Scale efficiency = 0.963 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	40.000	0.000	0.000	40.000
input TE	1440.000	-309.324	0.000	1130.676
input PC	58299.660	-12523.262	0.000	45776.398

LISTING OF PEERS:

peer	lambda	weight
SP036	0.125	
SP136	0.611	
SP146	0.264	

Results for SP022:

Technical efficiency = 0.807

Scale efficiency = 0.988 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	74.000	0.000	0.000	74.000
input TE	2560.000	-492.961	0.000	2067.039
input PC	95518.310	-18393.288	0.000	77125.022

LISTING OF PEERS:

peer	lambda	weight
SP079	0.280	
SP030	0.103	
SP036	0.618	

Results for SP023:

Technical efficiency = 1.000

Scale efficiency = 0.786 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	270.000	0.000	0.000	270.000
input TE	13524.050	0.000	0.000	13524.050
input PC	235557.620	0.000	0.000	235557.620

LISTING OF PEERS:

peer	lambda	weight
SP023	1.000	

Results for SP024:

Technical efficiency = 0.624

Scale efficiency = 0.975 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	65.000	0.000	0.000	65.000
input TE	3248.800	-1222.474	0.000	2026.326

input PC 85838.550 -32299.752 0.000 53538.798

LISTING OF PEERS:

peer lambda weight  
SP079 0.815  
SP046 0.122  
SP030 0.064

Results for SP025:

Technical efficiency = 0.974

Scale efficiency = 0.979 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	86.000	0.000	0.000	86.000
input TE	2512.000	-65.739	0.000	2446.261
input PC	88958.100	-2328.030	0.000	86630.070

LISTING OF PEERS:

peer lambda weight  
SP079 0.386  
SP030 0.215  
SP036 0.399

Results for SP026:

Technical efficiency = 0.987

Scale efficiency = 0.974 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	99.000	0.000	0.000	99.000
input TE	2871.780	-36.019	0.000	2835.761
input PC	100551.420	-1261.152	0.000	99290.268

LISTING OF PEERS:

peer lambda weight  
SP079 0.359  
SP030 0.308  
SP036 0.333

Results for SP027:

Technical efficiency = 0.979

Scale efficiency = 0.952 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	132.000	0.000	0.000	132.000
input TE	3840.000	-81.100	0.000	3758.900
input PC	181851.530	-3840.689	-26547.466	151463.375

LISTING OF PEERS:

peer lambda weight  
 SP122 0.299  
 SP036 0.701

Results for SP028:

Technical efficiency = 0.973

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	51.000	0.000	0.000	51.000
input TE	1480.000	-39.607	0.000	1440.393
input PC	49117.970	-1314.464	0.000	47803.506

LISTING OF PEERS:

peer lambda weight  
 SP079 0.746  
 SP030 0.024  
 SP036 0.231

Results for SP029:

Technical efficiency = 0.975

Scale efficiency = 0.979 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	88.000	0.000	0.000	88.000
input TE	2560.000	-63.677	0.000	2496.323
input PC	91957.970	-2287.359	0.000	89670.611

LISTING OF PEERS:

peer lambda weight  
 SP030 0.216  
 SP036 0.468  
 SP079 0.316

Results for SP030:

Technical efficiency = 1.000

Scale efficiency = 0.959 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	200.000	0.000	0.000	200.000
input TE	5840.000	0.000	0.000	5840.000
input PC	200074.580	0.000	0.000	200074.580

LISTING OF PEERS:

peer lambda weight  
 SP030 1.000

Results for SP031:

Technical efficiency = 0.993

Scale efficiency = 0.968 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	118.000	0.000	0.000	118.000
input TE	3440.000	-24.935	0.000	3415.065
input PC	117533.560	-851.943	0.000	116681.617

LISTING OF PEERS:

peer	lambda weight
SP030	0.458
SP036	0.157
SP079	0.386

Results for SP032:

Technical efficiency = 0.954

Scale efficiency = 0.987 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	55.000	0.000	0.000	55.000
input TE	1584.000	-72.783	0.000	1511.217
input PC	73107.120	-3359.180	-6535.021	63212.918

LISTING OF PEERS:

peer	lambda weight
SP036	0.594
SP136	0.406

Results for SP033:

Technical efficiency = 0.784

Scale efficiency = 0.950 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.347	35.347
input TE	1304.000	-281.942	0.000	1022.058
input PC	47220.540	-10209.704	0.000	37010.836

LISTING OF PEERS:

peer	lambda weight
SP146	0.347
SP095	0.653

Results for SP034:

Technical efficiency = 0.683

Scale efficiency = 0.916 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	185.000	0.000	0.000	185.000
input TE	9017.800	-2857.808	0.000	6159.992
input PC	236881.840	-75069.614	0.000	161812.226

LISTING OF PEERS:

peer	lambda	weight
SP030	0.207	
SP046	0.630	
SP122	0.163	

Results for SP035:

Technical efficiency = 0.716

Scale efficiency = 0.999 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	65.000	0.000	0.000	65.000
input TE	2481.800	-705.349	0.000	1776.451
input PC	98744.000	-28063.897	0.000	70680.103

LISTING OF PEERS:

peer	lambda	weight
SP030	0.010	
SP036	0.832	
SP079	0.159	

Results for SP036:

Technical efficiency = 1.000

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	68.000	0.000	0.000	68.000
input TE	1843.550	0.000	0.000	1843.550
input PC	76126.720	0.000	0.000	76126.720

LISTING OF PEERS:

peer	lambda	weight
SP036	1.000	

Results for SP037:

Technical efficiency = 0.958

Scale efficiency = 0.980 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	94.000	0.000	0.000	94.000
input TE	2744.400	-115.853	0.000	2628.547

input PC 106527.520 -4496.974 0.000 102030.546

LISTING OF PEERS:

peer lambda weight  
SP030 0.150  
SP122 0.029  
SP036 0.821

Results for SP038:

Technical efficiency = 0.985

Scale efficiency = 0.947 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	164.000	0.000	0.000	164.000
input TE	4787.660	-71.086	0.000	4716.574
input PC	193359.380	-2870.941	-1356.737	189131.702

LISTING OF PEERS:

peer lambda weight  
SP122 0.449  
SP036 0.551

Results for SP039:

Technical efficiency = 0.989

Scale efficiency = 0.961 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	153.000	0.000	0.000	153.000
input TE	4457.960	-49.326	0.000	4408.634
input PC	163477.120	-1808.836	0.000	161668.284

LISTING OF PEERS:

peer lambda weight  
SP030 0.462  
SP122 0.112  
SP036 0.426

Results for SP040:

Technical efficiency = 0.979

Scale efficiency = 0.975 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	40.000	0.000	0.000	40.000
input TE	1160.000	-24.567	0.000	1135.433
input PC	42544.770	-901.017	0.000	41643.753

LISTING OF PEERS:

peer lambda weight



SP146 0.695  
 SP036 0.125  
 SP136 0.180

Results for SP041:

Technical efficiency = 0.963

Scale efficiency = 0.996 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	50.000	0.000	0.000	50.000
input TE	1448.030	-53.345	0.000	1394.685
input PC	52038.130	-1917.073	0.000	50121.057

LISTING OF PEERS:

peer	lambda	weight
SP079	0.389	
SP146	0.234	
SP036	0.377	

Results for SP042:

Technical efficiency = 0.936

Scale efficiency = 0.815 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	30.000	0.000	5.151	35.151
input TE	1087.810	-70.103	0.000	1017.707
input PC	40288.870	-2596.387	0.000	37692.483

LISTING OF PEERS:

peer	lambda	weight
SP146	0.151	
SP095	0.849	

Results for SP043:

Technical efficiency = 0.865

Scale efficiency = 0.951 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.108	35.108
input TE	1176.000	-159.250	0.000	1016.750
input PC	43769.490	-5927.116	0.000	37842.374

LISTING OF PEERS:

peer	lambda	weight
SP146	0.108	
SP095	0.892	

Results for SP044:

Technical efficiency = 0.802

Scale efficiency = 0.983 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	68.000	0.000	0.000	68.000
input TE	2423.100	-479.686	0.000	1943.414
input PC	81118.690	-16058.566	0.000	65060.124

LISTING OF PEERS:

peer	lambda	weight
SP036	0.196	
SP079	0.668	
SP030	0.137	

Results for SP045:

Technical efficiency = 0.784

Scale efficiency = 0.955 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	78.000	0.000	0.000	78.000
input TE	3082.720	-667.382	0.000	2415.338
input PC	85262.370	-18458.556	0.000	66803.814

LISTING OF PEERS:

peer	lambda	weight
SP046	0.128	
SP030	0.141	
SP079	0.731	

Results for SP046:

Technical efficiency = 1.000

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	155.000	0.000	0.000	155.000
input TE	5726.020	0.000	0.000	5726.020
input PC	106266.170	0.000	0.000	106266.170

LISTING OF PEERS:

peer	lambda	weight
SP046	1.000	

Results for SP047:

Technical efficiency = 0.748

Scale efficiency = 0.984 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	48.000	0.000	0.000	48.000
input TE	1787.300	-450.066	0.000	1337.234
input PC	69429.190	-17483.181	0.000	51946.009

LISTING OF PEERS:

peer	lambda	weight
SP136	0.176	
SP146	0.449	
SP036	0.375	

Results for SP048:

Technical efficiency = 0.689

Scale efficiency = 0.972 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	39.000	0.000	0.000	39.000
input TE	1610.980	-501.561	0.000	1109.419
input PC	59595.820	-18554.496	0.000	41041.324

LISTING OF PEERS:

peer	lambda	weight
SP146	0.654	
SP036	0.094	
SP136	0.252	

Results for SP049:

Technical efficiency = 0.881

Scale efficiency = 0.942 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	85.000	0.000	0.000	85.000
input TE	2936.570	-349.181	0.000	2587.389
input PC	85575.940	-10175.635	0.000	75400.305

LISTING OF PEERS:

peer	lambda	weight
SP079	0.695	
SP046	0.100	
SP030	0.205	

Results for SP050:

Technical efficiency = 0.943

Scale efficiency = 0.997 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
----------	-------------------	--------------------	-------------------	--------------------

output FP	61.000	0.000	0.000	61.000
input TE	1786.220	-101.231	0.000	1684.989
input PC	67462.590	-3823.340	0.000	63639.250

LISTING OF PEERS:

peer	lambda	weight
SP079	0.357	
SP030	0.020	
SP036	0.623	

Results for SP051:

Technical efficiency = 0.950

Scale efficiency = 0.985 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	52.000	0.000	0.000	52.000
input TE	1512.180	-75.208	0.000	1436.972
input PC	61148.430	-3041.204	0.000	58107.226

LISTING OF PEERS:

peer	lambda	weight
SP036	0.500	
SP136	0.279	
SP146	0.221	

Results for SP052:

Technical efficiency = 0.957

Scale efficiency = 0.993 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	55.000	0.000	0.000	55.000
input TE	1583.860	-68.447	0.000	1515.413
input PC	62259.260	-2690.571	0.000	59568.689

LISTING OF PEERS:

peer	lambda	weight
SP146	0.380	
SP036	0.594	
SP136	0.027	

Results for SP053:

Technical efficiency = 0.968

Scale efficiency = 0.995 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	46.000	0.000	0.000	46.000
input TE	1338.420	-42.864	0.000	1295.556

input PC 45465.330 -1456.055 0.000 44009.275

LISTING OF PEERS:

peer lambda weight  
SP079 0.523  
SP036 0.231  
SP146 0.247

Results for SP054:

Technical efficiency = 0.958

Scale efficiency = 0.993 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	60.000	0.000	0.000	60.000
input TE	1754.800	-74.418	0.000	1680.382
input PC	62506.420	-2650.785	0.000	59855.635

LISTING OF PEERS:

peer lambda weight  
SP030 0.047  
SP036 0.424  
SP079 0.528

Results for SP055:

Technical efficiency = 0.984

Scale efficiency = 0.982 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	75.000	0.000	0.000	75.000
input TE	2169.200	-35.121	0.000	2134.079
input PC	75207.690	-1217.687	0.000	73990.003

LISTING OF PEERS:

peer lambda weight  
SP030 0.160  
SP036 0.314  
SP079 0.525

Results for SP056:

Technical efficiency = 0.971

Scale efficiency = 0.968 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	41.000	0.000	0.000	41.000
input TE	1192.000	-34.712	0.000	1157.288
input PC	47234.580	-1375.500	0.000	45859.080

LISTING OF PEERS:

peer lambda weight  
 SP146 0.359  
 SP036 0.156  
 SP136 0.485

Results for SP057:

Technical efficiency = 0.780

Scale efficiency = 0.959 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	131.000	0.000	0.000	131.000
input TE	4786.980	-1054.387	0.000	3732.593
input PC	189568.640	-41754.645	0.000	147813.995

LISTING OF PEERS:

peer lambda weight  
 SP030 0.079  
 SP122 0.246  
 SP036 0.675

Results for SP058:

Technical efficiency = 0.807

Scale efficiency = 0.956 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	125.000	0.000	0.000	125.000
input TE	4399.490	-850.082	0.000	3549.408
input PC	180575.030	-34891.215	-2460.387	143223.428

LISTING OF PEERS:

peer lambda weight  
 SP036 0.734  
 SP122 0.266

Results for SP059:

Technical efficiency = 0.952

Scale efficiency = 0.994 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	73.000	0.000	0.000	73.000
input TE	2109.660	-102.131	0.000	2007.529
input PC	83466.350	-4040.710	0.000	79425.640

LISTING OF PEERS:

peer lambda weight  
 SP079 0.084  
 SP030 0.055

SP036 0.861

Results for SP060:

Technical efficiency = 0.967

Scale efficiency = 0.994 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	61.000	0.000	0.000	61.000
input TE	1759.200	-58.780	0.000	1700.420
input PC	64069.900	-2140.746	0.000	61929.154

LISTING OF PEERS:

peer	lambda	weight
SP079	0.460	
SP030	0.041	
SP036	0.499	

Results for SP061:

Technical efficiency = 0.970

Scale efficiency = 0.993 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	59.000	0.000	0.000	59.000
input TE	1707.000	-51.078	0.000	1655.922
input PC	60069.510	-1797.428	0.000	58272.082

LISTING OF PEERS:

peer	lambda	weight
SP030	0.048	
SP036	0.385	
SP079	0.567	

Results for SP062:

Technical efficiency = 0.955

Scale efficiency = 0.995 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	59.000	0.000	0.000	59.000
input TE	1725.080	-77.297	0.000	1647.783
input PC	61949.830	-2775.819	0.000	59174.011

LISTING OF PEERS:

peer	lambda	weight
SP079	0.513	
SP030	0.037	
SP036	0.450	

Results for SP063:

Technical efficiency = 0.801

Scale efficiency = 0.982 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	91.000	0.000	0.000	91.000
input TE	3167.990	-630.318	0.000	2537.672
input PC	123896.600	-24651.041	0.000	99245.559

LISTING OF PEERS:

peer	lambda	weight
SP030	0.126	
SP122	0.030	
SP036	0.844	

Results for SP064:

Technical efficiency = 0.975

Scale efficiency = 0.966 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	127.000	0.000	0.000	127.000
input TE	3714.010	-91.603	0.000	3622.407
input PC	140058.380	-3454.433	0.000	136603.947

LISTING OF PEERS:

peer	lambda	weight
SP036	0.615	
SP030	0.285	
SP122	0.100	

Results for SP065:

Technical efficiency = 1.000

Scale efficiency = 0.942 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.000	35.000
input TE	1012.000	0.000	0.000	1012.000
input PC	40736.840	0.000	0.000	40736.840

LISTING OF PEERS:

peer	lambda	weight
SP065	1.000	

Results for SP066:

Technical efficiency = 0.954

Scale efficiency = 0.992 (drs)

PROJECTION SUMMARY:



variable	original value	radial movement	slack movement	projected value
output FP	70.000	0.000	0.000	70.000
input TE	2032.000	-93.472	0.000	1938.528
input PC	77767.050	-3577.284	0.000	74189.766

LISTING OF PEERS:

peer	lambda	weight
SP079	0.230	
SP030	0.062	
SP036	0.708	

Results for SP067:

Technical efficiency = 0.998

Scale efficiency = 0.963 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	38.000	0.000	0.000	38.000
input TE	1084.000	-2.495	0.000	1081.505
input PC	42186.410	-97.094	0.000	42089.316

LISTING OF PEERS:

peer	lambda	weight
SP146	0.441	
SP036	0.062	
SP136	0.496	

Results for SP068:

Technical efficiency = 0.988

Scale efficiency = 0.975 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	39.000	0.000	0.000	39.000
input TE	1124.000	-13.286	0.000	1110.714
input PC	40393.710	-477.447	0.000	39916.263

LISTING OF PEERS:

peer	lambda	weight
SP146	0.771	
SP036	0.094	
SP136	0.135	

Results for SP069:

Technical efficiency = 0.950

Scale efficiency = 0.984 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
----------	-------------------	--------------------	-------------------	--------------------

output FP	85.000	0.000	0.000	85.000
input TE	2480.000	-125.081	0.000	2354.919
input PC	99371.860	-5011.905	0.000	94359.955

LISTING OF PEERS:

peer	lambda	weight
SP036	0.899	
SP122	0.045	
SP030	0.057	

Results for SP070:

Technical efficiency = 0.975

Scale efficiency = 0.973 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	43.000	0.000	0.000	43.000
input TE	1240.000	-31.109	0.000	1208.891
input PC	48654.240	-1220.636	0.000	47433.604

LISTING OF PEERS:

peer	lambda	weight
SP146	0.402	
SP036	0.219	
SP136	0.379	

Results for SP071:

Technical efficiency = 0.966

Scale efficiency = 0.962 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	40.000	0.000	0.000	40.000
input TE	1168.000	-40.244	0.000	1127.756
input PC	51718.360	-1781.970	-1624.013	48312.378

LISTING OF PEERS:

peer	lambda	weight
SP036	0.125	
SP136	0.875	

Results for SP072:

Technical efficiency = 0.974

Scale efficiency = 0.964 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	41.000	0.000	0.000	41.000
input TE	1184.000	-30.680	0.000	1153.320
input PC	58929.080	-1526.964	-8096.369	49305.747

LISTING OF PEERS:

peer lambda weight  
SP036 0.156  
SP136 0.844

Results for SP073:

Technical efficiency = 0.951

Scale efficiency = 0.987 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	84.000	0.000	0.000	84.000
input TE	2448.000	-120.298	0.000	2327.702
input PC	96052.880	-4720.182	0.000	91332.698

LISTING OF PEERS:

peer lambda weight  
SP030 0.115  
SP122 0.004  
SP036 0.881

Results for SP074:

Technical efficiency = 0.945

Scale efficiency = 0.991 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	59.000	0.000	0.000	59.000
input TE	1708.000	-93.661	0.000	1614.339
input PC	70289.390	-3854.454	0.000	66434.936

LISTING OF PEERS:

peer lambda weight  
SP036 0.719  
SP136 0.203  
SP146 0.078

Results for SP075:

Technical efficiency = 0.988

Scale efficiency = 0.954 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	37.000	0.000	0.000	37.000
input TE	1064.000	-12.936	0.000	1051.064
input PC	47976.160	-583.286	-2060.604	45332.269

LISTING OF PEERS:

peer lambda weight  
SP036 0.031

SP136 0.969

Results for SP076:

Technical efficiency = 0.940

Scale efficiency = 0.987 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	55.000	0.000	0.000	55.000
input TE	1607.450	-96.233	0.000	1511.217
input PC	69713.570	-4173.525	-2327.127	63212.918

LISTING OF PEERS:

peer lambda weight

SP036 0.594

SP136 0.406

Results for SP077:

Technical efficiency = 0.993

Scale efficiency = 0.935 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	205.000	0.000	0.000	205.000
input TE	5986.000	-42.405	0.000	5943.595
input PC	272528.200	-1930.594	-33203.359	237394.247

LISTING OF PEERS:

peer lambda weight

SP122 0.640

SP036 0.360

Results for SP078:

Technical efficiency = 0.954

Scale efficiency = 0.975 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	90.000	0.000	0.000	90.000
input TE	2621.410	-119.459	0.000	2501.951
input PC	112423.180	-5123.165	-5276.320	102023.695

LISTING OF PEERS:

peer lambda weight

SP036 0.897

SP122 0.103

Results for SP079:

Technical efficiency = 1.000

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	41.000	0.000	0.000	41.000
input TE	1175.690	0.000	0.000	1175.690
input PC	34196.030	0.000	0.000	34196.030

LISTING OF PEERS:

peer	lambda	weight
SP079	1.000	

Results for SP080:

Technical efficiency = 0.983

Scale efficiency = 0.957 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	38.000	0.000	0.000	38.000
input TE	1095.070	-18.442	0.000	1076.628
input PC	47219.760	-795.219	-98.902	46325.639

LISTING OF PEERS:

peer	lambda	weight
SP036	0.062	
SP136	0.938	

Results for SP081:

Technical efficiency = 0.933

Scale efficiency = 0.999 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	67.000	0.000	0.000	67.000
input TE	1948.170	-130.184	0.000	1817.986
input PC	84003.090	-5613.403	-3256.336	75133.351

LISTING OF PEERS:

peer	lambda	weight
SP036	0.969	
SP136	0.031	

Results for SP082:

Technical efficiency = 0.968

Scale efficiency = 0.960 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	113.000	0.000	0.000	113.000
input TE	3295.650	-105.370	0.000	3190.280
input PC	144301.000	-4613.650	-10589.544	129097.806

LISTING OF PEERS:

peer lambda weight  
SP036 0.790  
SP122 0.210

Results for SP083:

Technical efficiency = 0.945

Scale efficiency = 0.994 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	61.000	0.000	0.000	61.000
input TE	1763.030	-97.732	0.000	1665.298
input PC	72592.390	-4024.099	0.000	68568.291

LISTING OF PEERS:

peer lambda weight  
SP036 0.781  
SP136 0.156  
SP146 0.063

Results for SP084:

Technical efficiency = 0.942

Scale efficiency = 0.984 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	53.000	0.000	0.000	53.000
input TE	1549.950	-89.861	0.000	1460.089
input PC	67186.210	-3895.233	-2064.798	61226.179

LISTING OF PEERS:

peer lambda weight  
SP036 0.531  
SP136 0.469

Results for SP085:

Technical efficiency = 0.938

Scale efficiency = 0.993 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	61.000	0.000	0.000	61.000
input TE	1775.920	-110.988	0.000	1664.932
input PC	73478.590	-4592.139	0.000	68886.451

LISTING OF PEERS:

peer lambda weight  
SP146 0.030  
SP036 0.781

SP136 0.189

Results for SP086:

Technical efficiency = 0.980

Scale efficiency = 0.951 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	136.000	0.000	0.000	136.000
input TE	3957.350	-78.741	0.000	3878.609
input PC	169140.110	-3365.455	-9602.739	156171.916

LISTING OF PEERS:

peer	lambda	weight
SP036	0.682	
SP122	0.318	

Results for SP087:

Technical efficiency = 0.947

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	68.000	0.000	0.000	68.000
input TE	1972.160	-103.999	0.000	1868.161
input PC	77485.500	-4086.087	0.000	73399.413

LISTING OF PEERS:

peer	lambda	weight
SP079	0.165	
SP030	0.034	
SP036	0.802	

Results for SP088:

Technical efficiency = 0.969

Scale efficiency = 0.970 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	43.000	0.000	0.000	43.000
input TE	1246.940	-39.172	0.000	1207.768
input PC	49979.160	-1570.074	0.000	48409.086

LISTING OF PEERS:

peer	lambda	weight
SP146	0.300	
SP036	0.219	
SP136	0.481	

Results for SP089:

Technical efficiency = 0.945

Scale efficiency = 0.994 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	74.000	0.000	0.000	74.000
input TE	2148.080	-117.660	0.000	2030.420
input PC	85887.500	-4704.458	0.000	81183.042

LISTING OF PEERS:

peer	lambda	weight
SP036	0.913	
SP079	0.035	
SP030	0.053	

Results for SP090:

Technical efficiency = 0.940

Scale efficiency = 0.987 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	78.000	0.000	0.000	78.000
input TE	2279.630	-136.807	0.000	2142.823
input PC	102300.640	-6139.332	-8263.235	87898.072

LISTING OF PEERS:

peer	lambda	weight
SP036	0.953	
SP122	0.047	

Results for SP091:

Technical efficiency = 0.943

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	72.000	0.000	0.000	72.000
input TE	2085.570	-119.906	0.000	1965.664
input PC	84636.910	-4866.047	0.000	79770.863

LISTING OF PEERS:

peer	lambda	weight
SP036	0.962	
SP079	0.007	
SP030	0.032	

Results for SP092:

Technical efficiency = 0.953

Scale efficiency = 0.987 (drs)

PROJECTION SUMMARY:



variable	original value	radial movement	slack movement	projected value
output FP	84.000	0.000	0.000	84.000
input TE	2448.000	-115.934	0.000	2332.066
input PC	95205.420	-4508.807	0.000	90696.613

LISTING OF PEERS:

peer	lambda	weight
SP079	0.027	
SP030	0.127	
SP036	0.846	

Results for SP093:

Technical efficiency = 0.968

Scale efficiency = 0.959 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	39.000	0.000	0.000	39.000
input TE	1138.200	-36.008	0.000	1102.192
input PC	52412.300	-1658.103	-3435.189	47319.008

LISTING OF PEERS:

peer	lambda	weight
SP036	0.094	
SP136	0.906	

Results for SP094:

Technical efficiency = 0.984

Scale efficiency = 0.971 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	41.000	0.000	0.000	41.000
input TE	1178.000	-19.264	0.000	1158.736
input PC	45342.910	-741.495	0.000	44601.415

LISTING OF PEERS:

peer	lambda	weight
SP136	0.354	
SP146	0.490	
SP036	0.156	

Results for SP095:

Technical efficiency = 1.000

Scale efficiency = 0.951 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.000	35.000

input TE	1014.350	0.000	0.000	1014.350
input PC	38218.370	0.000	0.000	38218.370

LISTING OF PEERS:

peer	lambda	weight
SP095	1.000	

Results for SP096:

Technical efficiency = 0.971

Scale efficiency = 0.957 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	120.000	0.000	0.000	120.000
input TE	3501.860	-102.089	0.000	3399.771
input PC	159885.280	-4661.080	-17886.448	137337.752

LISTING OF PEERS:

peer	lambda	weight
SP122	0.243	
SP036	0.757	

Results for SP097:

Technical efficiency = 0.986

Scale efficiency = 0.941 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	173.000	0.000	0.000	173.000
input TE	5056.110	-70.190	0.000	4985.920
input PC	226510.980	-3144.459	-23640.601	199725.920

LISTING OF PEERS:

peer	lambda	weight
SP122	0.491	
SP036	0.509	

Results for SP098:

Technical efficiency = 0.971

Scale efficiency = 0.967 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	116.000	0.000	0.000	116.000
input TE	3384.370	-97.931	0.000	3286.439
input PC	132098.280	-3822.446	0.000	128275.834

LISTING OF PEERS:

peer	lambda	weight
SP030	0.138	
SP122	0.139	

SP036 0.723

Results for SP099:

Technical efficiency = 0.964

Scale efficiency = 0.979 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	99.000	0.000	0.000	99.000
input TE	2885.650	-105.264	0.000	2780.386
input PC	110441.830	-4028.757	0.000	106413.073

LISTING OF PEERS:

peer	lambda	weight
SP030	0.197	
SP122	0.023	
SP036	0.780	

Results for SP100:

Technical efficiency = 0.946

Scale efficiency = 0.993 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	70.000	0.000	0.000	70.000
input TE	2044.810	-110.685	0.000	1934.125
input PC	78951.370	-4273.631	0.000	74677.739

LISTING OF PEERS:

peer	lambda	weight
SP079	0.201	
SP030	0.056	
SP036	0.743	

Results for SP101:

Technical efficiency = 0.985

Scale efficiency = 0.971 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	112.000	0.000	0.000	112.000
input TE	3262.680	-50.429	0.000	3212.251
input PC	115172.310	-1780.130	0.000	113392.180

LISTING OF PEERS:

peer	lambda	weight
SP079	0.244	
SP030	0.383	
SP036	0.372	

Results for SP102:

Technical efficiency = 0.946

Scale efficiency = 1.000 (crs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	57.000	0.000	0.000	57.000
input TE	1661.660	-90.418	0.000	1571.242
input PC	62564.760	-3404.411	0.000	59160.349

LISTING OF PEERS:

peer	lambda	weight
SP079	0.391	
SP146	0.014	
SP036	0.595	

Results for SP103:

Technical efficiency = 0.950

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	59.000	0.000	0.000	59.000
input TE	1727.220	-86.360	0.000	1640.860
input PC	63095.990	-3154.759	0.000	59941.231

LISTING OF PEERS:

peer	lambda	weight
SP079	0.467	
SP030	0.027	
SP036	0.506	

Results for SP104:

Technical efficiency = 0.956

Scale efficiency = 0.996 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	54.000	0.000	0.000	54.000
input TE	1562.740	-68.995	0.000	1493.745
input PC	58868.850	-2599.069	0.000	56269.781

LISTING OF PEERS:

peer	lambda	weight
SP146	0.227	
SP079	0.250	
SP036	0.523	

Results for SP105:

Technical efficiency = 0.994

Scale efficiency = 0.948 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	35.000	0.000	0.000	35.000
input TE	1019.760	-6.227	0.000	1013.533
input PC	39334.650	-240.209	0.000	39094.441

LISTING OF PEERS:

peer	lambda	weight
SP065	0.348	
SP095	0.652	

Results for SP106:

Technical efficiency = 0.980

Scale efficiency = 0.965 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	125.000	0.000	0.000	125.000
input TE	3631.970	-73.180	0.000	3558.790
input PC	139631.380	-2813.400	0.000	136817.980

LISTING OF PEERS:

peer	lambda	weight
SP030	0.204	
SP122	0.141	
SP036	0.656	

Results for SP107:

Technical efficiency = 0.951

Scale efficiency = 0.999 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	60.000	0.000	0.000	60.000
input TE	1729.980	-85.101	0.000	1644.879
input PC	67443.250	-3317.667	0.000	64125.583

LISTING OF PEERS:

peer	lambda	weight
SP146	0.051	
SP079	0.236	
SP036	0.713	

Results for SP108:

Technical efficiency = 0.992

Scale efficiency = 0.946 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
----------	----------	--------	-------	-----------

	value	movement	movement	value
output FP	186.000	0.000	0.000	186.000
input TE	5422.630	-41.622	0.000	5381.008
input PC	212541.350	-1631.370	0.000	210909.980

LISTING OF PEERS:

peer	lambda	weight
SP036	0.398	
SP030	0.131	
SP122	0.471	

Results for SP109:

Technical efficiency = 0.985

Scale efficiency = 0.941 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	170.000	0.000	0.000	170.000
input TE	4972.720	-76.582	0.000	4896.138
input PC	207924.790	-3202.117	-8528.159	196194.514

LISTING OF PEERS:

peer	lambda	weight
SP036	0.523	
SP122	0.477	

Results for SP110:

Technical efficiency = 0.954

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	64.000	0.000	0.000	64.000
input TE	1850.430	-84.283	0.000	1766.147
input PC	70750.410	-3222.532	0.000	67527.878

LISTING OF PEERS:

peer	lambda	weight
SP079	0.292	
SP030	0.029	
SP036	0.678	

Results for SP111:

Technical efficiency = 0.994

Scale efficiency = 0.949 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	195.000	0.000	0.000	195.000
input TE	5695.640	-33.689	0.000	5661.951

input PC 214857.690 -1270.855 0.000 213586.835

LISTING OF PEERS:

peer lambda weight  
SP036 0.260  
SP030 0.383  
SP122 0.357

Results for SP112:

Technical efficiency = 0.973

Scale efficiency = 0.975 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	46.000	0.000	0.000	46.000
input TE	1319.570	-35.887	0.000	1283.683
input PC	53519.660	-1455.517	0.000	52064.143

LISTING OF PEERS:

peer lambda weight  
SP146 0.230  
SP036 0.313  
SP136 0.457

Results for SP113:

Technical efficiency = 0.947

Scale efficiency = 0.980 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	50.000	0.000	0.000	50.000
input TE	1463.280	-78.096	0.000	1385.184
input PC	59890.280	-3196.383	0.000	56693.897

LISTING OF PEERS:

peer lambda weight  
SP146 0.162  
SP036 0.438  
SP136 0.401

Results for SP114:

Technical efficiency = 0.972

Scale efficiency = 0.970 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	114.000	0.000	0.000	114.000
input TE	3324.510	-94.198	0.000	3230.312
input PC	126974.300	-3597.756	0.000	123376.544

LISTING OF PEERS:

peer lambda weight  
 SP036 0.701  
 SP030 0.219  
 SP122 0.080

Results for SP115:

Technical efficiency = 0.981

Scale efficiency = 0.972 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	42.000	0.000	0.000	42.000
input TE	1206.210	-22.706	0.000	1183.504
input PC	47174.170	-888.009	0.000	46286.161

LISTING OF PEERS:

peer lambda weight  
 SP136 0.394  
 SP146 0.418  
 SP036 0.187

Results for SP116:

Technical efficiency = 0.973

Scale efficiency = 0.977 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	39.000	0.000	0.000	39.000
input TE	1142.250	-30.815	0.000	1111.435
input PC	40379.590	-1089.333	0.000	39290.257

LISTING OF PEERS:

peer lambda weight  
 SP136 0.070  
 SP146 0.836  
 SP036 0.094

Results for SP117:

Technical efficiency = 0.971

Scale efficiency = 0.989 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	65.000	0.000	0.000	65.000
input TE	1884.270	-55.023	0.000	1829.247
input PC	66779.490	-1950.050	0.000	64829.440

LISTING OF PEERS:

peer lambda weight  
 SP030 0.082



SP036 0.407  
 SP079 0.512

Results for SP118:

Technical efficiency = 0.998

Scale efficiency = 0.959 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	37.000	0.000	0.000	37.000
input TE	1057.660	-2.089	0.000	1055.571
input PC	41499.710	-81.982	0.000	41417.728

LISTING OF PEERS:

peer	lambda	weight
SP136	0.561	
SP146	0.408	
SP036	0.031	

Results for SP119:

Technical efficiency = 0.969

Scale efficiency = 0.982 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	82.000	0.000	0.000	82.000
input TE	2392.000	-73.016	0.000	2318.984
input PC	86189.070	-2630.940	0.000	83558.130

LISTING OF PEERS:

peer	lambda	weight
SP030	0.177	
SP036	0.479	
SP079	0.345	

Results for SP120:

Technical efficiency = 0.985

Scale efficiency = 0.956 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1045.420	-15.513	0.000	1029.907
input PC	41120.700	-610.176	0.000	40510.524

LISTING OF PEERS:

peer	lambda	weight
SP146	0.399	
SP136	0.601	

Results for SP121:

Technical efficiency = 0.975

Scale efficiency = 0.963 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	123.000	0.000	0.000	123.000
input TE	3584.000	-88.977	0.000	3495.023
input PC	140626.440	-3491.230	0.000	137135.210

LISTING OF PEERS:

peer	lambda	weight
SP030	0.119	
SP122	0.184	
SP036	0.697	

Results for SP122:

Technical efficiency = 1.000

Scale efficiency = 0.933 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	282.000	0.000	0.000	282.000
input TE	8248.000	0.000	0.000	8248.000
input PC	328033.660	0.000	0.000	328033.660

LISTING OF PEERS:

peer	lambda	weight
SP122	1.000	

Results for SP123:

Technical efficiency = 0.985

Scale efficiency = 0.942 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	167.000	0.000	0.000	167.000
input TE	4878.440	-72.084	0.000	4806.356
input PC	202007.980	-2984.864	-6360.008	192663.108

LISTING OF PEERS:

peer	lambda	weight
SP036	0.537	
SP122	0.463	

Results for SP124:

Technical efficiency = 0.989

Scale efficiency = 0.955 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
----------	----------	--------	-------	-----------

	value	movement	movement	value
output FP	162.000	0.000	0.000	162.000
input TE	4726.000	-53.556	0.000	4672.444
input PC	178059.600	-2017.801	0.000	176041.799

LISTING OF PEERS:

peer	lambda	weight
SP030	0.342	
SP122	0.229	
SP036	0.430	

Results for SP125:

Technical efficiency = 0.963

Scale efficiency = 0.972 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	101.000	0.000	0.000	101.000
input TE	2942.920	-108.718	0.000	2834.202
input PC	117220.180	-4330.367	0.000	112889.813

LISTING OF PEERS:

peer	lambda	weight
SP122	0.113	
SP036	0.820	
SP030	0.066	

Results for SP126:

Technical efficiency = 0.957

Scale efficiency = 0.986 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	49.000	0.000	0.000	49.000
input TE	1423.980	-61.114	0.000	1362.866
input PC	55252.370	-2371.322	0.000	52881.048

LISTING OF PEERS:

peer	lambda	weight
SP136	0.138	
SP146	0.455	
SP036	0.406	

Results for SP127:

Technical efficiency = 0.976

Scale efficiency = 0.960 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	37.000	0.000	0.000	37.000

input TE	1081.750	-25.825	0.000	1055.925
input PC	42115.110	-1005.419	0.000	41109.691

LISTING OF PEERS:

peer	lambda	weight
SP146	0.440	
SP036	0.031	
SP136	0.529	

Results for SP128:

Technical efficiency = 0.977

Scale efficiency = 0.972 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	118.000	0.000	0.000	118.000
input TE	3436.300	-79.724	0.000	3356.576
input PC	126545.520	-2935.916	0.000	123609.604

LISTING OF PEERS:

peer	lambda	weight
SP030	0.362	
SP122	0.010	
SP036	0.628	

Results for SP129:

Technical efficiency = 0.966

Scale efficiency = 0.974 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	43.000	0.000	0.000	43.000
input TE	1252.140	-42.997	0.000	1209.143
input PC	48893.350	-1678.930	0.000	47214.420

LISTING OF PEERS:

peer	lambda	weight
SP136	0.356	
SP146	0.425	
SP036	0.219	

Results for SP130:

Technical efficiency = 0.979

Scale efficiency = 0.955 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	146.000	0.000	0.000	146.000
input TE	4273.450	-90.737	0.000	4182.713
input PC	168216.780	-3571.696	0.000	164645.084

LISTING OF PEERS:

peer lambda weight  
SP030 0.105  
SP122 0.300  
SP036 0.595

Results for SP131:

Technical efficiency = 0.970

Scale efficiency = 0.978 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	102.000	0.000	0.000	102.000
input TE	2960.160	-87.614	0.000	2872.546
input PC	111624.400	-3303.819	0.000	108320.581

LISTING OF PEERS:

peer lambda weight  
SP030 0.249  
SP122 0.005  
SP036 0.746

Results for SP132:

Technical efficiency = 0.963

Scale efficiency = 0.995 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	57.000	0.000	0.000	57.000
input TE	1656.000	-61.949	0.000	1594.051
input PC	58737.630	-2197.321	0.000	56540.309

LISTING OF PEERS:

peer lambda weight  
SP030 0.031  
SP036 0.411  
SP079 0.558

Results for SP133:

Technical efficiency = 0.844

Scale efficiency = 0.991 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	65.000	0.000	0.000	65.000
input TE	2153.440	-335.476	0.000	1817.964
input PC	78273.660	-12193.934	0.000	66079.726

LISTING OF PEERS:

peer lambda weight

SP030 0.066  
 SP036 0.497  
 SP079 0.436

Results for SP134:

Technical efficiency = 0.982

Scale efficiency = 0.970 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	122.000	0.000	0.000	122.000
input TE	3562.230	-63.764	0.000	3498.466
input PC	126887.030	-2271.265	0.000	124615.765

LISTING OF PEERS:

peer	lambda	weight
SP030	0.436	
SP036	0.430	
SP079	0.134	

Results for SP135:

Technical efficiency = 0.945

Scale efficiency = 0.981 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	51.000	0.000	0.000	51.000
input TE	1491.150	-82.189	0.000	1408.961
input PC	65430.970	-3606.418	-2585.111	59239.441

LISTING OF PEERS:

peer	lambda	weight
SP036	0.469	
SP136	0.531	

Results for SP136:

Technical efficiency = 1.000

Scale efficiency = 0.952 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1025.500	0.000	0.000	1025.500
input PC	44338.900	0.000	0.000	44338.900

LISTING OF PEERS:

peer	lambda	weight
SP136	1.000	

Results for SP137:

Technical efficiency = 0.957

Scale efficiency = 0.973 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	93.000	0.000	0.000	93.000
input TE	2708.750	-117.017	0.000	2591.733
input PC	118263.790	-5108.934	-7599.755	105555.101

LISTING OF PEERS:

peer	lambda	weight
SP036	0.883	
SP122	0.117	

Results for SP138:

Technical efficiency = 0.944

Scale efficiency = 0.993 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	61.000	0.000	0.000	61.000
input TE	1763.100	-98.498	0.000	1664.602
input PC	81303.030	-4542.125	-7587.770	69173.134

LISTING OF PEERS:

peer	lambda	weight
SP036	0.781	
SP136	0.219	

Results for SP139:

Technical efficiency = 0.961

Scale efficiency = 0.973 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	46.000	0.000	0.000	46.000
input TE	1332.500	-51.359	0.000	1281.141
input PC	61362.100	-2365.118	-4724.389	54272.594

LISTING OF PEERS:

peer	lambda	weight
SP036	0.313	
SP136	0.687	

Results for SP140:

Technical efficiency = 0.985

Scale efficiency = 0.989 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
----------	-------------------	--------------------	-------------------	--------------------

output FP	40.000	0.000	0.000	40.000
input TE	1160.760	-17.562	0.000	1143.198
input PC	37376.640	-565.486	0.000	36811.154

LISTING OF PEERS:

peer	lambda	weight
SP079	0.443	
SP036	0.056	
SP146	0.502	

Results for SP141:

Technical efficiency = 0.964

Scale efficiency = 0.995 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	55.000	0.000	0.000	55.000
input TE	1605.300	-58.276	0.000	1547.024
input PC	55166.160	-2002.672	0.000	53163.488

LISTING OF PEERS:

peer	lambda	weight
SP030	0.034	
SP036	0.317	
SP079	0.649	

Results for SP142:

Technical efficiency = 0.963

Scale efficiency = 0.994 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	49.000	0.000	0.000	49.000
input TE	1421.530	-53.299	0.000	1368.231
input PC	51419.630	-1927.945	0.000	49491.685

LISTING OF PEERS:

peer	lambda	weight
SP079	0.294	
SP146	0.346	
SP036	0.360	

Results for SP143:

Technical efficiency = 0.853

Scale efficiency = 0.966 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	132.000	0.000	0.000	132.000
input TE	4427.350	-652.007	0.000	3775.343



input PC 164456.240 -24219.151 0.000 140237.089

LISTING OF PEERS:

peer lambda weight  
SP036 0.564  
SP030 0.357  
SP122 0.079

Results for SP144:

Technical efficiency = 0.792

Scale efficiency = 0.970 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	125.000	0.000	0.000	125.000
input TE	4514.940	-938.368	0.000	3576.572
input PC	162646.530	-33803.844	0.000	128842.686

LISTING OF PEERS:

peer lambda weight  
SP079 0.049  
SP030 0.442  
SP036 0.510

Results for SP145:

Technical efficiency = 0.953

Scale efficiency = 0.986 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	83.000	0.000	0.000	83.000
input TE	2408.100	-112.792	0.000	2295.308
input PC	96352.390	-4513.011	0.000	91839.379

LISTING OF PEERS:

peer lambda weight  
SP122 0.032  
SP036 0.906  
SP030 0.062

Results for SP146:

Technical efficiency = 1.000

Scale efficiency = 0.975 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1036.550	0.000	0.000	1036.550
input PC	34740.520	0.000	0.000	34740.520

LISTING OF PEERS:

peer lambda weight  
SP146 1.000

Results for SP147:

Technical efficiency = 0.960

Scale efficiency = 0.985 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	51.000	0.000	0.000	51.000
input TE	1470.580	-58.547	0.000	1412.033
input PC	58916.460	-2345.589	0.000	56570.871

LISTING OF PEERS:

peer lambda weight  
SP036 0.469  
SP136 0.253  
SP146 0.278

Results for SP148:

Technical efficiency = 0.956

Scale efficiency = 0.981 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	90.000	0.000	0.000	90.000
input TE	2622.410	-116.059	0.000	2506.351
input PC	103605.190	-4585.216	0.000	99019.974

LISTING OF PEERS:

peer lambda weight  
SP030 0.096  
SP122 0.044  
SP036 0.861

Results for SP149:

Technical efficiency = 0.980

Scale efficiency = 0.974 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	108.000	0.000	0.000	108.000
input TE	3135.390	-62.453	0.000	3072.937
input PC	113923.260	-2269.189	0.000	111654.071

LISTING OF PEERS:

peer lambda weight  
SP079 0.123  
SP030 0.328  
SP036 0.549

Results for SP150:

Technical efficiency = 0.949

Scale efficiency = 0.987 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	78.000	0.000	0.000	78.000
input TE	2257.000	-114.177	0.000	2142.823
input PC	108794.180	-5503.657	-15392.451	87898.072

LISTING OF PEERS:

peer	lambda	weight
SP036	0.953	
SP122	0.047	

Results for SP151:

Technical efficiency = 0.979

Scale efficiency = 0.952 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1048.000	-22.500	0.000	1025.500
input PC	48283.730	-1036.626	-2908.204	44338.900

LISTING OF PEERS:

peer	lambda	weight
SP136	1.000	

Results for SP152:

Technical efficiency = 0.955

Scale efficiency = 0.972 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	94.000	0.000	0.000	94.000
input TE	2746.000	-124.339	0.000	2621.661
input PC	119986.100	-5432.987	-7820.877	106732.236

LISTING OF PEERS:

peer	lambda	weight
SP036	0.879	
SP122	0.121	

Results for SP153:

Technical efficiency = 0.966

Scale efficiency = 0.958 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
----------	----------	--------	-------	-----------

	value	movement	movement	value
output FP	117.000	0.000	0.000	117.000
input TE	3426.000	-116.011	0.000	3309.989
input PC	143802.470	-4869.410	-5126.714	133806.346

LISTING OF PEERS:

peer	lambda	weight
SP036	0.771	
SP122	0.229	

Results for SP154:

Technical efficiency = 0.983

Scale efficiency = 0.981 (irs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	43.000	0.000	0.000	43.000
input TE	1233.510	-21.569	0.000	1211.941
input PC	45581.630	-797.050	0.000	44784.580

LISTING OF PEERS:

peer	lambda	weight
SP136	0.103	
SP146	0.678	
SP036	0.219	

Results for SP155:

Technical efficiency = 0.959

Scale efficiency = 0.986 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	84.000	0.000	0.000	84.000
input TE	2443.200	-100.056	0.000	2343.144
input PC	93289.400	-3820.458	0.000	89468.942

LISTING OF PEERS:

peer	lambda	weight
SP079	0.101	
SP030	0.142	
SP036	0.757	

Results for SP156:

Technical efficiency = 0.951

Scale efficiency = 0.999 (drs)

PROJECTION SUMMARY:

variable	original	radial	slack	projected
	value	movement	movement	value
output FP	57.000	0.000	0.000	57.000
input TE	1657.000	-81.759	0.000	1575.241

input PC 61667.460 -3042.761 0.000 58624.699

LISTING OF PEERS:

peer lambda weight  
SP079 0.433  
SP030 0.005  
SP036 0.562

Results for SP157:

Technical efficiency = 0.970

Scale efficiency = 0.999 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	52.000	0.000	0.000	52.000
input TE	1495.310	-45.195	0.000	1450.115
input PC	52610.410	-1590.128	0.000	51020.282

LISTING OF PEERS:

peer lambda weight  
SP079 0.608  
SP030 0.003  
SP036 0.389

Results for SP158:

Technical efficiency = 0.962

Scale efficiency = 0.996 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	58.000	0.000	0.000	58.000
input TE	1678.550	-64.404	0.000	1614.146
input PC	60945.870	-2338.409	0.000	58607.461

LISTING OF PEERS:

peer lambda weight  
SP079 0.490  
SP030 0.025  
SP036 0.485

Results for SP159:

Technical efficiency = 0.985

Scale efficiency = 0.968 (irs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	36.000	0.000	0.000	36.000
input TE	1050.000	-15.878	0.000	1034.122
input PC	37415.420	-565.795	0.000	36849.625

LISTING OF PEERS:

peer lambda weight  
SP146 0.780  
SP136 0.220

Results for SP160:

Technical efficiency = 0.994

Scale efficiency = 0.952 (drs)

PROJECTION SUMMARY:

variable	original value	radial movement	slack movement	projected value
output FP	193.000	0.000	0.000	193.000
input TE	5642.000	-33.044	0.000	5608.956
input PC	207765.930	-1216.831	0.000	206549.099

LISTING OF PEERS:

peer lambda weight  
SP036 0.212  
SP030 0.532  
SP122 0.256

## Anexo D: Publicaciones

<b>Título</b>	<b>Revista</b>	<b>Estado</b>	<b>Fecha de Presentación</b>	<b>Fecha de Aceptación</b>	<b>Año de Publicación</b>	<b>Volumen (Issue)</b>	<b>Páginas</b>
<i>A Review of Literature About Models and Factors of Productivity in the Software Factory</i>	International Journal of Information Technologies and Systems Approach (IJITSA)	Publicado	Julio 2017	Setiembre 2017	2018	11(1)	48-71
<i>New Factors Affecting Productivity of the Software Factory</i>	International Journal of Information Technologies and Systems Approach (IJITSA)	Aprobado	Mayo 2018	Mayo 2018	2020	13(1)	
<i>A Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Production Component of the Software Factory</i>	International Journal of Information Technologies and Systems Approach (IJITSA)	Aprobado	Junio 2018	Junio 2018	2020	13(2)	

## Anexo D.1: Cartas de Aceptación y/o Publicación

### A Review of Literature about Models and Factors of Productivity in the Software Factory Review Complete

9 mensajes

IGI Global Journal Submission System <journalsubmissionadmin@igi-global.com> 8 de septiembre de 2017, 18:15  
Responder a: mmora@securenym.net  
Para: pcastanedav@gmail.com



Dear Dr. Castañeda Vargas,

You will be happy to know that your manuscript #240717-074840, entitled "A Review of Literature about Models and Factors of Productivity in the Software Factory," submitted to the International Journal of Information Technologies and Systems Approach (IJITSA), has passed the journal's editorial review process, and academically has been ACCEPTED. Congratulations for this relevant research.

Next, your manuscript will be submitted to IGI Global for a final check to ensure that all publication requirements have been met. Your paper cannot be formally accepted for publication until this final step is complete. To ensure the timely and efficient completion of this step, please check that you have completed the following:

- All final submission requirements have been met as outlined in the Author's Checklist: <http://www.igi-global.com/publish/resources/journal-author-checklist.pdf>
- Each author on your manuscript has updated and confirmed their biography, email, and mailing address, and has signed the Author's Warranty and Transfer of Copyright Agreement. Any authors who have yet to complete any of these items may do so here: <https://www.igi-global.com/submission/copyright-agreement/?projectid=98eec5cc-c7d9-4389-8e34-302144c17827>

Should any of these materials be needed or need to be corrected, you will be contacted by a member of IGI Global's journal development team to secure the necessary details prior to publication.

No manuscript will be accepted unless it strictly follows the manuscript guidelines, i.e., must be professionally copyedited, references and citations formatted according to APA style guidelines, and includes all mandatory sections (Introduction, Conclusion, and References).

Thanks very much for this relevant contribution to software engineering area in IJITSA journal.

Sincerely,  
Prof. Mora - EIC of IJITSA  
Prof. O'Connor - Assoc. Editor for software engineering area in IJITSA

IGI Global  
eEditorial Discovery@



## Congratulations on Your Recent IGI Global Publication!

1 mensaje

IGI Global Notification <no-reply@igi-global.com>  
Para: pcastanedav@gmail.com

15 de enero de 2018, 10:01

Greetings Prof. Pedro Segundo Castañeda Vargas,

I trust this e-mail finds you well. First, I would like to congratulate you on your recent contributions to the *International Journal of Information Technologies and Systems Approach (IJITSA) 11(1)*. It is the contributions of prestigious and esteemed researchers such as yourself that help IGI Global remain as one of the top academic research publishers

All of us at IGI Global are very thankful to you for helping us fulfill our mission of offering content in all academic discipline areas. Undoubtedly, research institutions, academic and corporate libraries, as well as independent researchers all over the globe are benefitting from the advanced scholarly research and editorial support you have provided.

Additionally, we want to make sure that your institution's library, as well as your students and colleagues, can benefit from your scholarly research. By recommending the *International Journal of Information Technologies and Systems Approach (IJITSA) 11(1)* to your institution's library you can ensure that everyone can benefit from your valuable contributions. If you have not already recommended your journal to your institution's library, please use the link below to recommend your publication to your librarian:

[Send To Librarian Link](#)

We are also pleased to extend a 25% discount to your library for the purchase of any book or journal subscription through IGI Global's [online bookstore](#), by utilizing the discount code LIB25 upon checkout.

Please forward this discount, as well as the recommendation to your institution's Acquisition Librarian. Acquisition Librarians rely heavily on the scholarly expertise of faculty members such as yourself for recommendations on scholarly works that will enhance the university's library collection.

IGI Global currently has several opportunities to publish your valuable research including individual book chapters and journal articles as well as full book and journal editorship. IGI Global is also developing content-rich educational resources in the form of [streaming video](#) based on various topics across the broad spectrum of scholarly research. To take advantage of any of these publishing opportunities, please visit our [Publish with IGI Global](#) page for more information and to access our [Book Proposal Form](#) as well as our current [Call for Papers](#) for Book Chapters and Journal Articles. You may also contact our Acquisitions team directly at [acquisitions@igi-global.com](mailto:acquisitions@igi-global.com).

Again, congratulations on your recent publication, we thank you for supporting IGI Global over the past several years, and we look forward to our future collaborations for the years to come.

Sincerely,

The IGI Global Team

IGI Global (Disseminator of Knowledge Since 1988)  
International Publisher of Information Science and Technology Research  
Website: [www.igi-global.com](http://www.igi-global.com)

## New Factors Affecting Productivity of the Software Factory Review Complete

(Shared) Journal Submission System Admin <journalsubmissionadmin@igi-global.com>

8 de octubre de 2018,  
19:59

Responder a: Manuel Mora <mmora@securenym.net>

Para: "pcastanedav@gmail.com" <pcastanedav@gmail.com>

Cc: Manuel Mora <mmora@securenym.net>



# eEditorial Discovery

Supporting Your Editorial Experience

Dear Sr. Castañeda Vargas,

We report officially the ACCEPTANCE of your manuscript #230518-034135, entitled "New Factors Affecting Productivity of the Software Factory," submitted to the International Journal of Information Technologies and Systems Approach (IJITSA), which has passed the journal's editorial review process. Best congratulations!

Next, your manuscript will be submitted to IGI Global for a final check to ensure that all publication requirements have been met. Your paper cannot be formally accepted for publication until this final step is complete. To ensure the timely and efficient completion of this step, please check that you have completed the following:

- All final submission requirements have been met as outlined in the Author's Checklist: <https://www.igi-global.com/publish/contributor-resources/edited-book-author-checklist/>
- Each author on your manuscript has updated and confirmed their biography, email, and mailing address, and has signed the Author's Warranty and Transfer of Copyright Agreement. Any authors who have yet to complete any of these items may do so here: <https://www.igi-global.com/submission/copyright-agreement/?projectid=63175a2e-da04-49a9-9197-92182188c099>

Should any of these materials be needed or need to be corrected, you will be contacted by a member of IGI Global's journal development team to secure the necessary details prior to publication.

No manuscript will be accepted unless it strictly follows the manuscript guidelines, i.e., must be professionally copyedited, references and citations formatted according to APA style guidelines, and includes all mandatory sections (Introduction, Conclusion, and References).

IJITSA 2019-1 and 2019-2 issues are already completed, so your paper will be scheduled for 2020-1 issue.

Sincerely,  
Prof. Mora / EIC of IJITSA

IGI Global  
eEditorial Discovery®

## Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Factory Review Complete

2 mensajes

(Shared) Journal Submission System Admin <journalsubmissionadmin@igi-global.com>

8 de enero de 2019,  
19:40

Responder a: Manuel Mora <mmora@securenym.net>

Para: "pcastanedav@gmail.com" <pcastanedav@gmail.com>

Cc: Manuel Mora <mmora@securenym.net>



**eEditorial Discovery**  
Supporting Your Editorial Experience

Dear author Castañeda Vargas,

The third round review for your manuscript #220718-053828, entitled "Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Factory," submitted to the International Journal of Information Technologies and Systems Approach (IJITSA), has been completed. We are pleased to report you the ACCEPTANCE status.

Next, your manuscript will be submitted to IGI Global for a final check to ensure that all publication requirements have been met. Your paper cannot be formally accepted for publication until this final step is complete. To ensure the timely and efficient completion of this step, please check that you have completed the following:

- All final submission requirements have been met as outlined in the Author's Checklist: <https://www.igi-global.com/publish/contributor-resources/edited-book-author-checklist/>
- Each author on your manuscript has updated and confirmed their biography, email, and mailing address, and has signed the Author's Warranty and Transfer of Copyright Agreement. Any authors who have yet to complete any of these items may do so here: <https://www.igi-global.com/submission/copyright-agreement/?projectid=924ab056-477f-45e2-9498-e90ae7aa017d>

Should any of these materials be needed or need to be corrected, you will be contacted by a member of IGI Global's journal development team to secure the necessary details prior to publication.

No manuscript will be accepted unless it strictly follows the manuscript guidelines, i.e., must be professionally copyedited, references and citations formatted according to APA style guidelines, and includes all mandatory sections (Introduction, Conclusion, and References).

Thanks for your important contribution to IJITSA in the Software Engineering area.  
Sincerely,  
Prof. Mora / EIC of IJITSA

IGI Global  
eEditorial Discovery

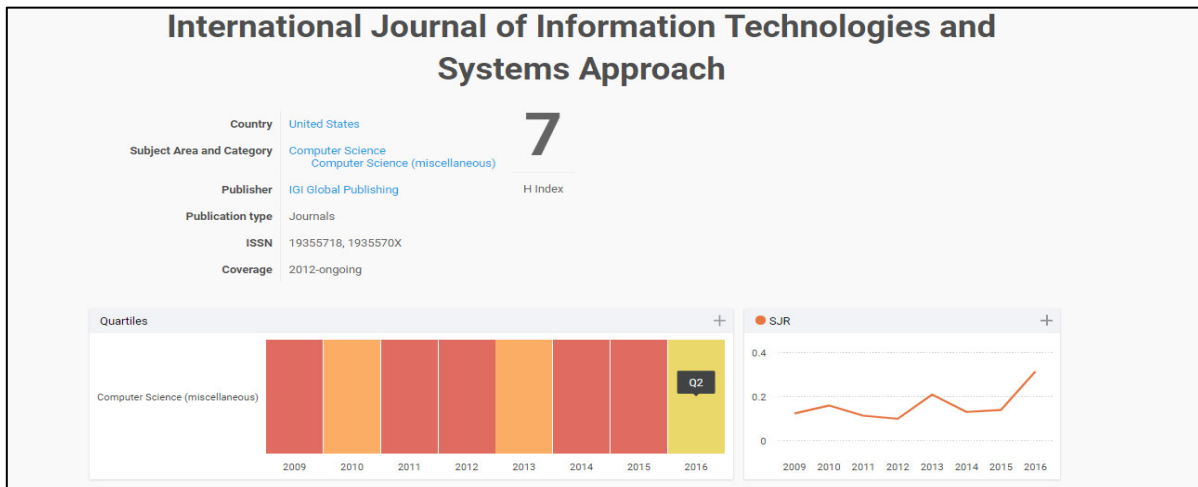
You have received this email because you are associated with a project in the IGI Global eEditorial Discovery® system. Adjust where notifications are sent by adding or updating your primary email address at <https://www.igi-global.com/account/e-mail/> (login required). Please contact [cust@igi-global.com](mailto:cust@igi-global.com) for assistance.

# Anexo D.2: A Review of Literature About Models and Factors of Productivity in the Software Factory

Source Title: International Journal of Information Technologies and Systems Approach (IJITSA) 11(1)

Copyright: © 2018 |Pages: 24

DOI: 10.4018/IJITSA.2018010103



International Journal of Information Technologies and Systems Approach  
Volume 11 • Issue 1 • January-June 2018

## A Review of Literature About Models and Factors of Productivity in the Software Factory

Pedro S. Castañeda Vargas, National University of San Marcos, Lima, Peru

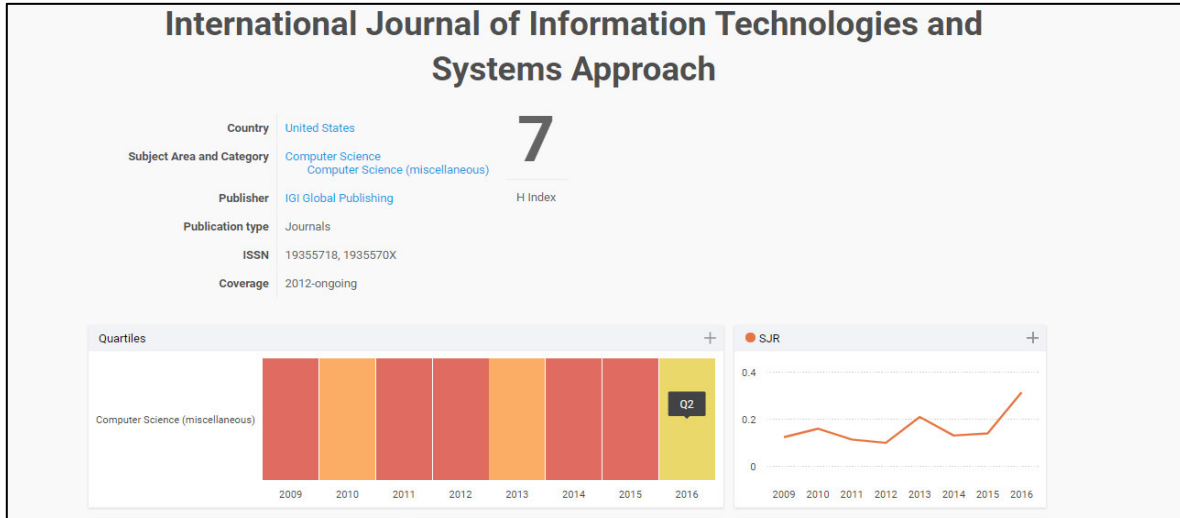
David Mauricio, National University of San Marcos, Lima, Peru

### ABSTRACT

Software factories seek to develop quality software in a manner comparable to the production of other industrial products, establishing improvements in their production process so as to be more competitive. Productivity, one of the competitiveness pillars, is related to the effort required to fulfill assigned tasks. However, there is no standard way of measuring productivity, making it difficult to establish policies and strategies to improve the factory. In this article, the authors perform a systematic review of the literature on factors affecting productivity of software factories, and models for measuring it. For the period 2005-2017, 74 factors and 10 models are identified. Most of the factors are related to Programming, and a few to Analysis and Design and Testing. Also, most models for measuring productivity only consider activities concerning programming.

# Anexo D.3: New Factors Affecting Productivity Of The Software Factory

Source Title: International Journal of Information Technologies and Systems Approach (IJITSA) 13(1)



## New Factors Affecting Productivity of the Software Factory

**Pedro Castañeda**  
Universidad Peruana de Ciencias Aplicadas (UPC); National University of San Marcos (UNMSM)

**David Mauricio**  
National University of San Marcos

**ABSTRACT**  
Productivity is very important because it allows organizations to achieve greater efficiency and effectiveness in their activities, however, is affected by numerous factors. While these factors have been identified for over two decades, all of the previous works limited the software factory to the Programming work unit and did not analyze other work units that are also relevant. 90% of a software factory's effort is absorbed by the Software Production component, 85% of which is concentrated in the efforts of the Analysis & Design, Programming, and Testing work units. The present work identifies three new factors that influence the software factory, demonstrating that the Use of Rules and Events influences Analysis & Design, Team Heterogeneity negatively affects Analysis & Design and positively affects Programming, and the Osmotic Communication affects in Programming. An empirical study on software factories in Peru, determined that 95% of the influence came from these factors, which corroborated as well that Team Size and Trust within the Team influences in Software Production.

*Keywords: Heterogeneity, Osmotic Communication, Productivity, Rules, Software Factory, Factors, Software Production, Trust*

# Anexo D.4: A Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Production Component of the Software Factory

Source Title: International Journal of Information Technologies and Systems Approach (IJITSA) 13(2)



## Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Factory

Pedro Castañeda

Universidad Peruana de Ciencias Aplicadas, Universidad Nacional Mayor de San Marcos

David Mauricio

Universidad Nacional Mayor de San Marcos

### ABSTRACT

Productivity in software factories is very important because it allows organizations to achieve greater efficiency and effectiveness in their activities. One of the pillars of competitiveness is productivity, and it is related to the effort required to accomplish the assigned tasks. However, there is no standard way to measure it, making it difficult to establish policies and strategies to improve the factory. In this work, a model based on Data Envelopment Analysis is presented to evaluate the relative efficiency of the software factories and their projects, to measure the productivity in the Software Production Component of the Software Factory through the activities that are carried out in their different work units. The proposed model consists of two phases in which the productivity of the software factory is evaluated and the productivity of the projects it conducts is assessed. Numerical tests on 6 software factories with 160 projects implemented show that the proposed model allows one to assess the software factories and the most efficient projects.

Keywords: productivity, software factory, software production, benchmarking, DEA, efficiency, effectiveness, measurement