



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

**Framework como guía para la definición del alcance de
la externalización del desarrollo de aplicaciones
mediante un modelo de software factory**

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Ingeniería de Software

AUTOR

Albert Einstein AZABACHE PEÑA

ASESOR

Dr. Frank Edmundo ESCOBEDO BAILÓN

Lima, Perú

2024



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

Azabache, A. (2024). *Framework como guía para la definición del alcance de la externalización del desarrollo de aplicaciones mediante un modelo de software factory*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Albert Einstein Azabache Peña.
Tipo de documento de identidad	DNI
Número de documento de identidad	03661580
URL de ORCID	https://orcid.org/0009-0007-0714-5367
Datos de asesor	
Nombres y apellidos	Frank Edmundo Escobedo Bailón
Tipo de documento de identidad	DNI
Número de documento de identidad	41671087
URL de ORCID	https://orcid.org/0000-0002-2058-0976
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cayo Víctor León Fernández
Tipo de documento	DNI
Número de documento de identidad	07001405
Miembro del jurado 1	
Nombres y apellidos	Ciro Rodriguez Rodriguez
Tipo de documento	DNI
Número de documento de identidad	06020241
Miembro del jurado 2	
Nombres y apellidos	Nehil Indalicio Muñoz Casildo
Tipo de documento	DNI
Número de documento de identidad	09727623
Miembro del jurado 3	
Nombres y apellidos	Frank Edmundo Escobedo Bailón
Tipo de documento	DNI

Número de documento de identidad	41671087
Datos de investigación	
Línea de investigación	C.0.3.22 Ingeniería de Software
Grupo de investigación	No aplica.
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -12.055926 Longitud: -77.083469
Año o rango de años en que se realizó la investigación	2022 – 2023
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO DE MAGÍSTER
EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON MENCIÓN EN
INGENIERÍA DE SOFTWARE**

A los nueve (09) días del mes de febrero de 2024, siendo las 11:00 am., se reunieron en el Auditorio, Profesor: Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. Cayo Victor León Fernández (Presidente)
Dr. Ciro Rodríguez Rodríguez (Miembro)
Mg. Nehil Indalicio Muñoz Casildo (Miembro)
Dr. Frank Edmundo Escobedo Bailón (Miembro Asesor)

Se inició la Sustentación invitando al candidato a Magister **ALBERT EINSTEIN AZABACHE PEÑA**, para que realice la exposición oral de la tesis para optar el Grado de Magister en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, siendo la Tesis intitulada:

**"FRAMEWORK COMO GUÍA PARA LA DEFINICIÓN DEL ALCANCE DE LA
EXTERNALIZACIÓN DEL DESARROLLO DE APLICACIONES MEDIANTE UN
MODELO DE SOFTWARE FACTORY"**

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) MUY BUENO

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Bachiller **ALBERT EINSTEIN AZABACHE PEÑA** el Grado de Magister en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software.

Siendo las 13:00 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.

Dr. Cayo Víctor León Fernández
(Presidente)

Dr. Ciro Rodríguez Rodríguez
(Miembro)

Mg. Nehil Indalicio Muñoz Casildo
(Miembro)

Dr. Frank Edmundo Escobedo Bailón
(Miembro Asesor)



CERTIFICADO DE SIMILITUD

Yo Frank Edmundo Escobedo Bailón en mi condición de asesor acreditado con Dictamen N° 000299-2022-UPG-VDIP-FISI/UNMSM de la tesis, cuyo título es Framework como guía para la definición del alcance de la externalización del desarrollo de aplicaciones mediante un modelo de software Factory, presentado por el Bach. Albert Einstein Azabache Peña para optar el grado de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 2% (dos por ciento) de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado correspondiente.

Firma del Asesor _____

DNI: 41671087

Nombres y apellidos del asesor:

Frank Edmundo Escobedo Bailón



DEDICATORIA

Dedico este logro a Dios todo poderoso y a la Santísima Virgen María, por guiarme, darme la fuerza y la sabiduría necesaria para completar este trabajo de investigación.

A mi madre Celsa, que en este mundo es a quien todo le debo.

A mi hermana Liliana, con mucho cariño y amor.

A Adrián y Fabrizio, como ejemplo de que con fe y tesón logramos nuestras metas.

A la memoria de Catalino y de mis abuelos Marcos y Marcelina, cuya vida integra fue su ejemplo y su legado para mí.

AGRADECIMIENTOS

A mi madre una resiliente y triunfadora, guía para mi vida y soporte para alcanzar mis metas.

A la eminente Universidad Nacional Mayor de San Marcos, por haber sido el claustro que me ha formado como Magíster.

A mis profesores de maestría, por sus enseñanzas y por el nivel de exigencia académica que establecieron y que nos retó a dar nuestro mejor esfuerzo.

A mi asesor de tesis, por la revisión de este trabajo de investigación y por animarme a culminarlo.

Al actual equipo administrativo de la Unidad de Posgrado de la Facultad de Ingeniería de Sistemas e Informática, por su orientación oportuna sobre los diversos trámites que debemos seguir los tesisistas para llegar a sustentar nuestra tesis.

A quienes me dieron ánimo durante este prolijo proceso de elaboración de mi tesis, para poder lograr el objetivo de finalizarla, sustentarla y lograr el grado de Magister.

TABLA DE CONTENIDOS

1	PLANTEAMIENTO DEL PROBLEMA	1
1.1	Situación Problemática.....	1
1.1.1	Información no disponible sobre como externalizar vía una SWF	3
1.1.2	Complejidades para definir el alcance a externalizar.....	3
1.2	Formulación del Problema	6
1.3	Justificación teórica	6
1.4	Justificación práctica	7
1.5	Objetivos	8
1.5.1	Objetivo general.....	8
1.5.2	Objetivos específicos	8
1.6	Propuesta	8
1.7	Hipótesis.....	9
1.7.1	Hipótesis General.....	9
1.7.2	Hipótesis específicas	9
1.8	Organización de la tesis.....	9
2	MARCO TEÓRICO	11
2.1	Marco epistemológico	11
2.2	Antecedentes de la investigación.....	11
2.3	Bases Teóricas	12
2.3.1	Externalización de funciones de TI.....	12
2.3.2	Contextualizando la software factory de esta investigación	14
2.3.3	Alcance de la externalización- AE.....	15
2.3.4	Requisitos técnicos de la externalización (RTE)	17

2.3.5	Definiendo el término aplicación.....	17
2.3.6	SLCM, SLC y SDLC	17
2.3.7	El correcto significado de metodología.....	18
2.3.8	Marco metodológico para el DMS a usarse en la SWF	18
2.3.9	Los principales SLCM y sus PS.....	18
2.3.10	Entendiendo los principales SLCM	20
2.3.11	Un metamodelo para definir métodos de desarrollo de software.....	22
2.3.12	key performance indicators (KPI) para aplicaciones	24
2.3.13	Clasificación de aplicaciones	26
2.3.14	Offshoring	27
2.3.15	eSourcing	27
2.3.16	IDEF0.....	27
2.4	Estado del Arte	29
2.4.1	De la planificación de la RSL	30
2.4.2	De la de ejecución de la RSL	30
2.4.3	De los resultados de la RSL	30
3	METODOLOGÍA.....	34
3.1	Tipo y diseño de la investigación	34
3.1.1	Constructos de MEM	35
3.1.2	Adaptación de MEM.....	36
3.1.3	Aplicación de la adaptación de MEM.....	39
3.2	Unidad de análisis.....	40
3.3	Población de estudio.....	40
3.4	Tamaño de la muestra.....	40
3.5	Diseño del experimento.....	40
3.5.1	Selección de sujetos para el experimento.....	41
3.5.2	Variables	41

3.5.3	Tratamiento experimental	42
3.5.4	Instrumentos	42
3.5.5	Operación del experimento	44
4	Framework propuesto	45
4.1	Macroproceso uno (macropro-1): Determinar el alcance técnico de la externalización.....	46
4.1.1	Diagrama de contexto y descripción de macropro-1.....	46
4.1.2	Diagrama de descomposición del macropro-1.....	48
4.1.3	Procesos del macroproceso 01	49
4.1.4	Prácticas sugeridas para cada proceso del macropro-1	53
4.2	Macroproceso dos (macropro-2): Definir un marco metodológico de DMS para la externalización.....	59
4.2.1	Diagrama de contexto y descripción del macropro-2.	59
4.2.2	Diagrama de descomposición y procesos de macropro-2.....	61
4.2.3	Procesos del macroproceso 02	62
4.2.4	Prácticas sugeridas para cada proceso de macropro02	64
5	Validación y Resultados	69
5.1	Validez y fiabilidad de los datos recopilados en la encuesta.....	69
5.1.1	Validez	69
5.1.2	Fiabilidad	70
5.2	Pruebas de hipótesis	72
5.2.1	Evalando H1 y H2 analizando los resultados para las variables de percepción y de desempeño.	72
5.2.2	Evalando H6, la relación causales entre PEOU y PU.....	74
5.3	Presentación de resultados.....	75
6	Conclusiones y trabajos futuros.....	76
6.1	Conclusiones	76
6.2	Trabajos futuros.....	76

7	Bibliografía	78
8	Anexos	84
8.1	Anexo 01. Principales problemas de la externalización de servicios de TI.	84

LISTA DE TABLAS

Tabla 1. Resultados iniciales de la RSL.....	30
Tabla 2. Constructos de MEM	36
Tabla 3. Hipótesis de MEM	39
Tabla 4. Ítems de la encuesta tipo Likert para medir las variables de percepción.....	43
Tabla 5. Descripción de los elementos del macropro-1	47
Tabla 6. Descripción del proceso revisión del contenido de la arquitectura.....	49
Tabla 7. Descripción del proceso caracterización de externalizables.....	50
Tabla 8. Descripción del proceso Desarrollar los RTE.....	51
Tabla 9. Descripción del proceso V&V de los DDA.....	52
Tabla 10. Descripción de los elementos del macropro-2 de SWFrame	60
Tabla 11. Descripción del proceso seleccionar el SLCM para la SWF	62
Tabla 12. Descripción de preparar una metodología de DMS para el SLCM	63
Tabla 13. Descripción de asignar los Roles de la Metodología a los de la SWF.....	64
Tabla 14. Correlación inter-ítems a los datos recopilados	71
Tabla 15. Análisis de fiabilidad para los ítems de la encuesta.....	72
Tabla 16. Datos paramétricos de las variables basadas en desempeño.....	72
Tabla 17. Estadísticos descriptivos de PEOU y PU.....	73
Tabla 18. Estadísticos descriptivos por cada ítem de PEOU	73
Tabla 19 Estadísticos descriptivos por cada ítem de PU.....	73
Tabla 20. Prueba de normalidad shapiro-Wilk para PEOU y PU	74
Tabla 21. Estadísticas descriptivas y prueba t de 1 cola con valor de prueba tres.....	74
Tabla 22. Correlación de Pearson entre PU y PEOU.....	74
Tabla 23. Listado de 23 problemas de la externalización de servicios de TI.....	84

LISTA DE FIGURAS

Figura 1. El AE y el PI como parte del ciclo de vida genérico de una SWF	2
Figura 2. Vista de procesos genéricos de una Software Factory.....	15
Figura 3. Variantes de los SLCM.....	19
Figura 4. Modelos y Procesos de Software.....	20
Figura 5. Posicionamiento de conceptos clave de UMA	23
Figura 6. Vista de un proceso de entrega basado en UMA.....	24
Figura 7. Costos de interfaces y aplicaciones	26
Figura 8. Caja o Box de una diagrama IDEF0.....	28
Figura 9. Diagrama padre e hijo en notación IDEF0	29
Figura 10. Fases y las CA de eSCM-C	31
Figura 11. Prácticas del área de capacidad Sourcing Planing.....	32
Figura 12. Metas y prácticas genéricas del área ARD de CMMI-ACQ.....	32
Figura 13. El Modelo de Evaluación de Métodos.....	35
Figura 14. Variables basadas en percepción del MEM.....	37
Figura 15. Constructos e hipótesis del modelo MEM.....	38
Figura 16. Procesos de SWFrame	45
Figura 17. Diagrama de contexto para el macroproceso 01 de SWFrame.....	46
Figura 18. Diagrama de descomposición del macropro1 de SWFrame.....	48
Figura 19. Diagrama de contexto del macropro-2	59
Figura 20. Diagrama de descomposición del macropro-2	61

RESUMEN

Externalizar el desarrollo de software es una práctica frecuente en la estrategia de las organizaciones, una modalidad adoptada por compañías cuyos procesos de negocio están altamente soportados en sistemas de software y que buscan eficacia en sus proyectos que implementan sus requerimientos de TI, es la de tercerizar la función misma de desarrollo y mantenimiento del software (DMS), en lugar de sólo el alcance de un proyecto determinado. Así, se delega esta función a un proveedor experimentado por un plazo determinado y generalmente amplio, en el que este atiende todas las necesidades de DMS que surjan en ese periodo, para tal efecto este dispone una unidad de desarrollo constituida por un equipo especializado y con los medios necesarios para atender de forma exclusiva las mismas, a este set de recursos se le conoce como software factory (SWF). En muchos ámbitos como el peruano a esta modalidad se le conoce como externalización por SWF, a la que nosotros hemos llamado modelo de SWF. Uno de los principales desafíos de este tipo de outsourcing y que surge una vez tomada la decisión, es el de definir un esmerado alcance de la externalización (AE) que precise qué se delega y cómo, es decir los requisitos y condiciones establecidas por la organización contratante, adquiriente en adelante. La calidad del resultado marcará el derrotero de esta empresa. Este alcance debe abarcar diversos aspectos administrativos, financieros, técnicos entre otros, pero es el técnico la parte medular del mismo. Este trabajo propone SWFrame, un framework que ayuda en la definición de este AE, específicamente para la parte sustancial relacionada con la de identificar los componentes técnicos que deben tercerizarse y las condiciones en que debe hacerse. Este está compuesto por procesos con sus elementos adecuadamente descritos y prácticas recomendadas para su implementación. La percepción de utilidad y facilidad de su uso fue validada con buenos resultados siguiendo un método adecuado que midió la efectividad sobre su finalidad y la eficiencia de su aplicación respectivamente. SWFrame como propuesta ayuda a afrontar ese desafío principal que explicamos previamente, que cuando no se logra y aun así se delega se convierte en una de las principales causa del fracaso de este tipo externalizaciones.

ABSTRACT

Outsourcing software development is a frequent practice in the strategy of organizations. A modality adopted by companies whose business processes are highly supported by software systems and that seek efficiency in their projects that implement their IT requirements is to outsource the software development and maintenance (DMS) function itself, rather than just the scope of a given project. Thus, this function is delegated to an experienced provider for a specific and generally long period of time, in which he attends to all the DMS needs that arise in that period. For this purpose, he has a development unit of a specialized team with the necessary means to exclusively address them, this set of resources is known as a software factory (SWF). In many areas, such as Peru, this modality is known as SWF outsourcing, which we have called the SWF model. One of the main challenges of this type of outsourcing, which arises once the decision has been made, is to define a careful scope of outsourcing (AE) that specifies what is delegated and how, that is, the requirements and conditions established by the organization contracting party, acquirer from now on. The quality of the result will determine the course of this company. This scope must cover various administrative, financial, and technical aspects, among others, but the technical aspect is the core part of it. This work proposes SWFrame, a framework that helps in the definition of this EA, specifically for the substantial part related to identifying the technical components that should be outsourced and the conditions under which it should be done. This is made up of processes with their elements adequately described and recommended practices for their implementation. The perception of usefulness and ease of use was validated with good results following an appropriate method that measured the effectiveness of its purpose and the efficiency of its application respectively. SWFrame as a proposal helps to face the main challenge that we explained previously, that when it is not achieved and even so delegated, it becomes one of the main causes of the failure of this type of outsourcing.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Situación Problemática

Diversas organizaciones externalizan la función del desarrollo de software y mantenimiento de software (DMS), a empresas proveedoras especializadas y con vasta experiencia en este quehacer, ya sea para construir nuevas aplicaciones o para mantener las existentes; una manera de hacerlo es delegar parcial o totalmente el alcance de un proyecto, en el marco de condiciones que el contratante o adquiriente define por contrato. Esta forma requiere un proceso de licitación por cada proyecto que se requiera externalizar, evidentemente esto resta agilidad a la organización adquiriente. Otra modalidad adoptada por compañías cuyos procesos de negocio están altamente soportados en sistemas de software y que buscan eficacia en sus proyectos que implementan sus requerimientos de TI, es la de delegar ya no un proyecto, sino la función misma de DMS a un proveedor experimentado por un plazo determinado y amplio durante el cual este se hace cargo de todos los proyectos que surjan; en muchos ámbitos como el peruano a esta modalidad se le conoce como tercerización por software factory (SWF), a la que nosotros llamaremos modelo de SWF, que además implica delegar también los servicios de atención de incidentes en aplicaciones y los de certificación del software, que se suele delegar a un proveedor diferente al que desarrolla. El modelo tiene variantes como cuando se contrata más de un proveedor, entre los cuales se reparte la función de DMS de cada aplicación.

La externalización vía SWF le brinda muchos beneficios al adquiriente como es tener un centro de desarrollo dedicado a atender sus requerimientos de software, evitar tener que realizar licitaciones cada vez que surge un proyecto a externalizar, aumento de la eficiencia en los costos de los recursos dedicados al DMS, permitirle al adquiriente enfocarse en su giro de negocio principal, entre otras; sin embargo su adopción demanda una planificación prolija que debe lograr definir adecuadamente el alcance de la externalización (AE) y el plan de implementación (PI); el primero debe indicar con claridad y precisión las características de las aplicaciones, los elementos de TI, procesos y responsabilidades que se delegaran al proveedor de SWF, indica el qué se delega y bajo qué condiciones y/o mecanismos de operación, lo que representa el cómo el proveedor realizará las funciones que se le van a delegar; el segundo es el plan

para poner en operación la SWF, para lo cual deben definirse la estructura y los procesos de gobierno de esta, procedimientos de transferencia de los activos tecnológicos necesarios y del conocimiento desde el adquirente al proveedor de SWF, además de dejar establecido cómo se realizará el proceso de cierre operativo, llegado el momento, donde se debe precisar los mecanismos de transferencia de lo administrado por esta a otro proveedor o al adquirente cuando finalice el contrato por fin de plazo u otro motivo, entre otras cosas. Tanto el AE como el PI son documentos interrelacionados que se deben complementar, para su elaboración se requiere de un proyecto que los logre, cuyo equipo debe considerar todos los aspectos involucrados: técnicos, administrativos, financieros, normativos entre otros. La figura 1, permite ubicar el AE y PI como el resultado de este proyecto y como el insumo de las etapas subsiguientes del ciclo de la externalización vía SWF.

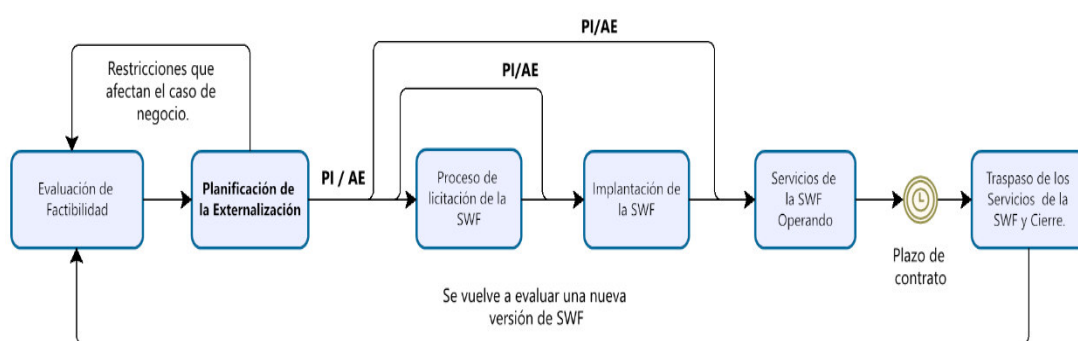


Figura 1. El AE y el PI como parte del ciclo de vida genérico de una SWF
Fuente: (Azabache, 2022).

Una de las cosas que se debe tener en cuenta y que a priori no se considera es que externalizar bajo este modelo implica no solo delegar el DMS propiamente dicho, sino además la administración de la infraestructura que soporta a las aplicaciones cuya función de DMS se delega (en adelante las llamaremos aplicaciones delegadas), los procesos que permiten administrarlas y mantenerlas; se pudiera pensar qué sólo es delegar la codificación únicamente y no es así como funciona esta práctica de externalización.

En los ítems siguientes, 1.1.1 y 1.1.2 ampliamos los retos y problemática en la adopción de esta modalidad de externalización.

1.1.1 Información no disponible sobre como externalizar vía una SWF

Aunque se han realizado muchas externalizaciones de este tipo, no hay información disponible de las actividades que se llevaron a cabo para definir el AE o el PI. En las empresas privadas los TOR (Terms of Reference) de sus licitaciones no suelen ser públicos, si en las entidades gubernamentales, al menos de Perú por la legislación vigente, pero en ningún caso se hace público el proceso seguido previo a la licitación, ni tampoco las lecciones aprendidas sobre cómo les fue con la adopción del modelo; así las compañías con planes de externalizar vía SWF parten sin un referente documental al respecto.

1.1.2 Complejidades para definir el alcance a externalizar

1.1.2.1 Diversos frentes técnicos que considerar

Tenemos entre otros las siguientes:

- a. Revisar la arquitectura empresarial con énfasis en la perspectiva de aplicaciones y la infraestructura tecnológica para determinar los ítems que se requiere externalizar y bajo qué requisitos técnicos y de gestión hacerlo.
- b. Definir y/o adoptar el más adecuado marco metodológico de DMS para usar en la relación adquiriente SWF.
- c. Definir procesos adecuados para gestionar las otras funciones delegadas a la SWF, p.ej. la gestión de las pruebas de software, de la demanda.
- d. Definir un proceso de gestión de necesidades, a usarse para canalizar, definir, estimar y priorizar los requerimientos informáticos hacia la SWF.
- e. Definir y consensuar acuerdos de niveles de servicio (ANS) para precisar la efectividad requerida en brindar los servicios para las funciones delegadas.
- f. Definir métodos de estimación para cuantificar el esfuerzo, plazos y costes de los requerimientos informáticos que el adquiriente solicitará a la SWF.

1.1.2.2 Dimensionar la software factory (capacity)

Consiste en definir los tipos de recursos, sus características, cantidades, entre otros que serán parte de la SWF. Para efectos de la licitación y de administración de la SWF se necesita estimar de antemano cuántas horas hombre por aplicación o por área usuaria

u otro agrupador se asignarán y se presupuestarán para consumirse por cada servicio contratado a la SWF. P.ej. para la aplicación App del área A se estiman H horas-hombre para desarrollos durante el presenta año y se requieren tales y tantos recursos. Obtener esa estimación requiere de un análisis adecuado de la demanda futura en función a la pasada y al portafolio de proyectos planificado, en concordancia con los objetivos y presupuestos del adquiriente.

Ciertamente dimensionar el capacity es una tarea compleja, pero ineludible, pues de esta estimación dependerá la asignación del presupuesto a asignar a la externalización y los recursos a contratar por el proveedor de SWF; por exceso ocasionará desperdicios y por defecto demoras o negativas en la atención de los requerimientos de TI que surjan.

1.1.2.3 Determinar la estructura organizacional de la SWF y sus procesos de gobernanza.

Implica definir una adecuada estructura de gobierno que responda eficazmente a las necesidades del adquiriente. Debe estar soportada por los procesos administrativos, financieros y operativos necesarios que se deben definir para que la SWF lleve a cabo su misión.

1.1.2.4 Establecer el proceso de implantación y cierre de la SWF.

Hay muchos aspectos técnicos y administrativos que se deben considerar para implantar la SWF cuándo se tenga un proveedor seleccionado, p.ej. la sesión de la administración de activos tecnológicos, temas de seguridad informática para la habilitación de accesos a sistemas y aplicaciones, transferencia de conocimiento explícito, tácito, entre otros; sin lugar a dudas este proceso es complejo y tiene que definirse de antemano en el PI, lo mismo que el cierre de la SWF cuando el contrato finalice por fin de plazo u otros motivos, hay que establecer el proceso de traspaso a un nuevo proveedor o de retorno de lo delegado al adquiriente (back sourcing).

1.1.2.5 Adecuación a aspectos normativos, legales y culturales

La definición del AE debe respetar y adecuarse a la normas organizacionales y a las leyes vigentes, p.ej. tener en cuenta políticas de confidencialidad, regulaciones sobre uso de datos o normas y leyes sobre tercerizaciones. También se deben de considerar

aspectos de la cultura organizacional del adquiriente que deben armonizarse con las del proveedor a seleccionar.

Por lo indicado previamente podemos concluir que definir el AE es ciertamente una tarea compleja, extensa y que debe de realizarse con prolijidad, toda vez que, si se soslayan definiciones o se omiten ítems externalizables, esto se hará visibles durante la operación de la SWF cuando surjan actividades o incidentes que pongan el foco sobre ciertos elementos que no aparecen en el AE, o las definiciones asociadas son poco claras e insuficientes. El AE es parte nuclear del contrato entre las partes.

1.2 Formulación del Problema

Problema General

¿Un framework será de utilidad para ayudar en la definición de los aspectos técnicos del AE mediante un modelo de SWF?

Problema específico N°1 (PE01)

¿Un framework con procesos apropiados, podrá ayudar a determinar los ítems a externalizar y los requisitos técnicos bajo las cuales se deben delegar a la SWF la función de DMS de la organización adquiriente?

Problema específico N°2 (PE02)

¿Un framework con procesos apropiados, permitirá establecer los criterios apropiados para la selección de un adecuado marco metodológico para el ciclo de vida del desarrollo del software a usarse en el ámbito de la SWF?

Fuera de alcance de la investigación

La investigación se enmarca en lo enunciado en los problemas específicos y no contempla:

1. Lo relacionado a gestión de incidentes, gestión de necesidades o demanda, al proceso de licitación, al dimensionamiento, implantación y cierre de la SWF.
2. Lo relacionado a procesos de gobernanza de la SWF como gestión de recursos humanos, financieros, operacionales y otros similares.
3. Lo relacionado a factibilidad financiera o técnica.
4. A métodos o técnicas de estimación de esfuerzos para requisitos de DMS.

1.3 Justificación teórica

El estado del arte de nuestra investigación indicó que al momento de la revisión no se encontraron estudios que propongan como definir el AE de la función de DMS por parte de una compañía vía la modalidad como la descrita o similar. En esa línea (Ikram y otros, 2018) indica que, aunque la externalización del mantenimiento del software es un proceso complejo que abarca diferentes prácticas, no existen ninguna publicación que aborde las prácticas más significativas y describa los elementos básicos de este

proceso, como actividades, roles, productos de trabajo y similares. En el mismo sentido (Naciri, 2014) indica que hay pocos enfoques propuestos en la literatura que han sido capaces de enfrentar los desafíos de esta área de ingeniería de software en condiciones específicas y en un contexto de outsourcing; así mismo (Ahmad y otros, 2017) señala que el desarrollo de modelos de madures es un área nueva para externalizaciones de TI en general. La literatura encontrada sobre la externalización del DMS, toca temas como resolución de conflictos (Cho, 2020), temas financieros (Montequín y otros, 2013), la decisión de tercerizar (SHAHZAD y otros, 2017), ingeniería de requisitos (IQBAL y otros, 2022) y de cuestiones sobre offshoring (RAHMAN y otros, 2021) (Barney y otros, 2014) (Petersen y otros, 2013), entre otros tópicos.

El proceso y resultados de la RSL que hemos realizado lo mostramos en el apartado estado del arte del marco teórico de este trabajo.

1.4 Justificación práctica

El framework que se propone como resultado de este trabajo será de utilidad para los profesionales de TI involucrados en la planificación de externalizaciones de la función del DMS, porque tendrán una guía que los ayude a definir el AE, no en todas sus partes, pero si en una de las más importantes que es lo relacionado a los aspectos técnicos de ingeniería a considerar; también les permitirá visualizar la magnitud de esta empresa al considerar aspectos técnicos importantes durante la etapa factibilidad, que pudieran soslayarse en ese estadio, lo que les ayudará en las estimaciones iniciales, en la identificación de riesgos, en la planificación del road map del proyecto entre otros aspectos.

Debemos tener en cuenta que un AE mal elaborado ya sea porque no considera algún aspecto importante, porque es ambiguo, porque no es específico, ni medible o verificable es fuente de problemas durante la operación y de conflictos entre las partes cuando se hagan evidentes sus carencias, los que afectarán a los objetivos de los proyectos del adquiriente que estén en marcha y ejecutándose por la SWF. Nuestra propuesta de un framework, implícita en los objetivos y descrita en el capítulo correspondiente, coadyuvará a una correcta definición del AE en los aspectos planteados en nuestro objetivos de investigación, ayudando al equipo del adquiriente

a enfocarse en aspectos que le ayudarán a considerar e identificar los requisitos técnicos de la externalización (RTE).

Nuestra propuesta también se justifica, porque este modelo de externalización es de frecuente adopción, y casi siempre recurrente pues las organizaciones suelen lanzar un nuevo proceso de externalización cuando está finalizando el contrato con su actual proveedor de SWF, lo cual implica definir el AE respectivo. Finalmente, una justificación adicional es el que hay ausencia de información sobre cómo llevar a cabo este tipo de outsourcing, tal como lo explicamos en el apartado 1.1.1.

1.5 Objetivos

1.5.1 Objetivo general

Desarrollar un framework que sirva de guía consultiva para definir apropiadamente el AE del DMS mediante un modelo de SWF.

1.5.2 Objetivos específicos

El framework se circunscribirá al ámbito de los siguientes objetivos específicos:

Objetivos Específicos

OE01: Definir procesos del framework que permitan determinar los requisitos técnicos de la externalización bajo las cuales se debe delegar a la SWF la función de DMS de la organización adquirente.

OE02: Definir procesos del framework que permitan establecer criterios apropiados para seleccionar un marco metodológico de desarrollo de software en un contexto de externalización vía un modelo de SWF.

1.6 Propuesta

Desarrollar el framework en el ámbito de los objetivos planteados, el cual estará compuesto por procesos con objetivos específicos, entradas, salidas y prácticas específicas posibles de aplicar para implementar cada proceso.

1.7 Hipótesis

1.7.1 Hipótesis General

HG: El Framework propuesto será de utilidad y de fácil uso para coadyuvar a definir el AE del DMS vía un modelo de SWF, en lo relacionado a aspectos técnicos en un contexto de externalización a largo plazo.

La HG la precisamos y dividimos en función a los objetivos específicos en las siguientes hipótesis específicas.

1.7.2 Hipótesis específicas

HE1: El Framework es útil y fácil de usar para coadyuvar a determinar los ítems a externalizar y los requisitos técnicos bajo los cuales estos se deben externalizar a la SWF para que esta pueda realizar la función del DMS delegada.

HE2: El Framework es útil y fácil de usar para coadyuvar a establecer criterios apropiados para seleccionar un marco metodológico de DMS adecuado en un contexto de externalización de largo plazo como el modelo de SWF.

Las cualidades de útil y fácil de usar se refieren a que el framework cumple con sus objetivos y es fácilmente comprensible respectivamente.

1.8 Organización de la tesis

En el capítulo uno se planteó nuestro problema de investigación contextualizándolo adecuadamente, también los objetivos de la investigación, la propuesta de la tesis y las hipótesis relacionadas con la misma. En el capítulo dos corresponde al marco teórico, que nos permitió fundamentalmente definir algunos términos relevantes que nos ayudaron a contextualizar nuestro tema de investigación y a definir conceptos que hemos referenciado desde determinadas prácticas recomendadas del framework propuesto. El capítulo tres expone la metodología que hemos usado en la investigación donde explicamos nuestro tipo y diseño de la investigación, de la cual es parte MEM, método de evaluación de modelos, que nos permitirá validar nuestra propuesta. El capítulo cuatro desarrollamos SWFrame nuestra propuesta de framework, esto es sus

procesos con sus respectivos elementos, sus descripciones y las prácticas sugeridas como actividades para lograr su finalidad. En el capítulo cinco presentamos los resultados de la validación de nuestras hipótesis de investigación usando lo propuesto por MEM, explicado en el capítulo tres. Finalmente, el capítulo seis presenta nuestras conclusiones y recomendaciones de trabajos futuros relacionados.

2 MARCO TEÓRICO

2.1 Marco epistemológico

Gran parte del conocimiento es tácito, lo que implica que no se puede acceder directamente y para hacerlo explícito se requiere recopilar experiencias y transformarlas desde alguna fuente a alguna forma de conocimiento explícito (Birk y otros, 1999). La fuente del conocimiento tácito es la mente de las personas, que guarda el conocimiento de sus experiencias pasadas. Existe un proceso llamado exteriorización, por el cual se pasa del conocimiento tácito al explícito, que puede adoptar forma de metáforas, analogías, conceptos, hipótesis o modelos (Bjørnson & Dingsøyr, 2008). Las empresas que hacen ingeniería del software tienen el reto de aplicar con mucha intensidad este proceso para materializar todo el saber hacer adquirido por su personal en muchísimos proyectos y mantenimientos de sus sistemas y que la empresa no documenta y que pierde con el paso del tiempo o cuando estas personas dejan la compañía.

En esta ilación de ideas, el problema planteado en este trabajo ha tenido seguramente respuestas en el ámbito empresarial, pero estas o no han quedado documentadas y si lo fueron no son públicas, con lo cual no se pueden transferir, analizar, criticar, mejorar, replicar. Estas respuestas se convirtieron en conocimiento tácito. En el ámbito académico las fuentes primarias sólo plantean tangencialmente el problema sin brindar respuestas específicas a cómo definir el alcance de este tipo de externalizaciones. Por lo expuesto creemos que nuestro trabajo es en buena parte un proceso de exteriorización que busca generar conocimiento explícito sobre cómo abordar el problema planteado desde la perspectiva de la experiencia.

2.2 Antecedentes de la investigación

Al momento de la revisión realizada para constatar el estado del arte, no se encontraron estudios que propongan de forma específica y cabal como definir el AE de la función de DMS en la forma precisada en este trabajo o similar.

Las publicaciones que hemos encontrado y que se acercan al ámbito de nuestros objetivos de investigación son las de (Ikram y otros, 2018), (POLO y otros, 2002),

(Phillips, 2011), (Brownsword y otros, 2014), (Kronawitter y otros, 2013) y (Hefley & Loesche, 2006) las que parcialmente plantean aspectos relacionados con la definición del alcance a externalizar en lo relacionado a aspectos técnicos del DMS. Hemos realizado una síntesis de estas publicaciones en el apartado estado del arte de este capítulo, donde también explicamos el proceso de la RSL seguido.

2.3 Bases Teóricas

2.3.1 Externalización de funciones de TI

Externalizar es el equivalente en español a outsourcing, consiste en transferir una función propia de una compañía a una empresa tercera especializada (Ketler & Walstrom, 1993). Lo transferido podría ser algo muy acotado o abarcar todo lo realizado por un área empresarial. En el entorno de TI donde esta estrategia es ampliamente usada se le conoce como *Information Technology Outsourcing*, o simplemente ITO y en esta industria es usada desde hace décadas, así tenemos p.ej. que en los 60 se contrataban servicios de procesamiento de datos para evitar adquirir hardware informático que en aquella época era muy costoso y de gran tamaño, por lo que requería ambientes de espacio considerable y adecuadamente acondicionados. En los 70 la preocupación fue el costo del desarrollo de software, debido al incremento de la demanda de sistemas de software y a la escases de personal especializado, esto obligo a externalizar también esta función. (Ketler & Walstrom, 1993). Un hito importante de la práctica de externalizar fue bautizado como evento Kodak, que destaco por lo amplio y agresivo para su época, ocurrió en 1989 cuando la Eastman Kodak Company externalizo sus operaciones de data center (mainframes) a IBM, su desarrollo de software a Anderson Consulting, lo relacionado a personal computing a Businessland Inc. y telecomunicaciones y redes a Digital Equipment Corporation (Ketler & Walstrom, 1993). A partir de este acontecimiento las grandes corporaciones aplicaron con más énfasis esta estrategia de negocios, que de hecho se ha diversificado abarcando una serie de servicios que se contratan a terceros como los servicios de SaaS (software as a Service), o los servicios SOC (Security Operations Center) entre muchos otros.

Entre los principales beneficios de la externalización esta permitir al adquiriente enfocarse en su núcleo negocio, reducir costos, aumentar su rendimiento

aprovechando la experiencia y la especialización de sus proveedores como ventaja competitiva. En la actualidad muchos procesos de una organización están parcial o totalmente externalizados, y con ellos el software que los automatiza y la infraestructura tecnológica que lo soporta, en esta relación los cambios cada vez más frecuentes en los procesos de negocio detonan requerimientos informáticos que buscan adaptar el software a los mismos, impactando también en la infraestructura tecnológica, y en general en todos los dominios de la arquitectura empresarial.

La externalización de servicios de TI no está exenta de problemas, así (Hefley & Loesche, 2006) indica que las fallas o fracasos de esta están centradas mayormente en un conjunto principal de cuestiones críticas, y describe 23 problemas que esta enfrenta, que se indican en el anexo 1, de estos por lo menos tres están relacionados con la definición del AE y son causa de fracaso de esta: (1) traducir lo que requiere el negocio en requisitos correctamente especificados con los niveles de calidad acordados (2) lograr contratos esmerados (3) gestionar las relaciones entre clientes y proveedores de servicios para asegurar que se cumplan los compromisos. Coincidimos que estas causales pueden hacer que la externalización falle, porque como se desprende de (1), el no tener un alcance adecuado de lo que se externaliza y sobre la forma de hacerlo, hará que se tenga un contrato débil, dado que aquel es su insumo principal, que dificultará controlar el trabajo del proveedor porque hay falta de claridad, ambigüedad o ausencia de definiciones sobre las responsabilidades de este, lo que ocasionará un impacto en los servicios que brinda ya sea en calidad o completitud, ocasionando conflictos y hasta el fracaso mismo de la externalización. Los problemas (2) y (3) y otros de los listados son consecuencia principalmente de alcances mal definidos.

De lo anterior se desprende que externalizar las funciones de TI requiere de una buena planificación, que debe tener como base un robusto alcance de la externalización, también que esta no es un simple traslado de funciones a un proveedor, ni un posterior desentendimiento del quehacer de este, donde simplemente el adquiriente se sienta de brazos cruzados a esperar resultados, esta concepción es nefasta para lograr una externalización exitosa y para la organización adquiriente.

2.3.2 Contextualizando la software factory de esta investigación

El tipo de SWF al que nos referimos en esta investigación es aquella unidad de desarrollo de software administrada por un proveedor especializado que realiza la función de DMS para una compañía que la ha contratado para tal efecto de forma continua, bajo sus condiciones, por un periodo amplio determinado y generalmente a exclusividad. En el plazo la SWF se hace cargo de los requerimientos informáticos relacionados con las aplicaciones delegadas. Esta unidad está conformada por recursos humanos y materiales, cuyo sizing ha sido establecido en función al análisis de capacity y capability realizado por el adquiriente, previo a la contratación. Las empresas que brindan este servicio de SWF lo hacen a más de una compañía, obviamente con otros recursos y condiciones diferentes (Azabache, 2022).

En la Figura 2, muestra como fluyen los procesos básicos en una SWF para atender los requerimientos del adquiriente.

Ciertamente existen fábricas de software con otro enfoque, como las de aquellas que fabrican aplicaciones comerciales listas para instalar y usar u otras que requieren cierto grado de personalización antes de desplegarse para su uso como las aplicaciones de facturación, de puntos de venta u otras más complejas como sistemas ERP o CRM (Azabache, 2022), a este grupo de aplicaciones se les conoce con el acrónimo anglosajón COTS de Commercial off-the-shelf, generalmente más a las primeras.

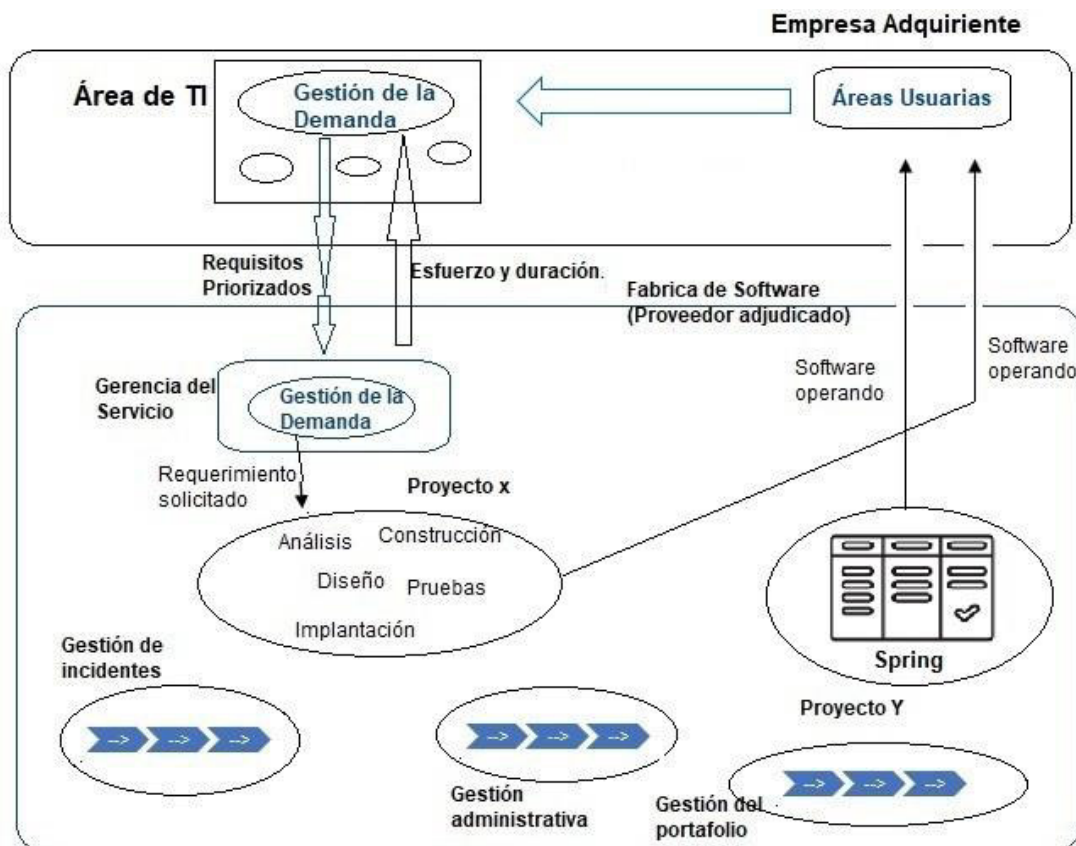


Figura 2. Vista de procesos genéricos de una Software Factory

Fuente. Propia.

2.3.3 Alcance de la externalización- AE

El AE es un documento que contiene de manera específica lo qué se externalizará y cómo el proveedor debe brindar sus servicios para cumplir con la función que le ha sido delegada por el adquiriente. El AE debe abarcar los requerimientos técnicos, financieros, de operación y los necesarios para garantizar que el proveedor cumpla a cabalidad la función que se le delega y para poder asegurar una supervisión exitosa mediante requisitos verificables. En lo referente a los técnicos debe establecer responsabilidades sobre los ítems a externalizar relacionados con las aplicaciones delegadas, debe abarcar también los procesos que regularán la operación y el gobierno de la SWF, como p.ej. la metodología de DMS, el proceso de gestión de demanda, el método para estimar esfuerzos de construcción, entre otros que se deben definir durante la planificación de la externalización. El AE es la parte principal del contrato entre las partes (Azabache, 2022).

Una de las causas del fracaso de la externalización es la de tener un contrato pobre, es decir incompleto, poco prolijo, inespecífico, lo que dificultará enormemente gobernar

la externalización y probablemente la haga fracasar. Una ilustración de esto son estas cifras, 30% de la empresas cancelan sus contratos de externalización antes de su fin, entre el 20 a 25% fracasan en los primeros dos años de contrato y el 50% no sobrevive más allá de los primeros 5 años (Grossi & Calvo-Manzano, 2012 citado en Núñez-Sánchez & González-Torres, 2020), y aunque estas cifras no son recientes ni específicas para externalizaciones de desarrollo de software, reflejan el impacto de un mal contrato, que además como mencionamos es uno de los problemas más frecuentes en las ITO (Hefley & Loesche, 2006).

En el caso específico de la externalización del DMS, las causales que con mayor frecuencia la hacen fracasar son la definición inadecuada del alcance a externalizar, una mala selección de los proveedores y de metodologías de desarrollo, así como el bajo involucramiento de los interesados, entre ellos los usuarios finales, en cada fase del proceso de desarrollo (Núñez-Sánchez & González-Torres, 2020). Todos los anteriores puntos deben ser parte del AE, incluidos los aspectos de gestión de las comunicaciones, del monitoreo y control, roles requeridos entre otros aspectos ya comentados; si este AE se define mal se tendrá un contrato pobre, con el que no se podrá determinar si una de las partes está cumpliendo con sus obligaciones, generará conflictos sino litigios, probablemente adendas que impactarán en costos y plazos, además del deterioro de las relaciones entre las partes y de la falta de confianza a todo nivel de la organización.

2.3.4 Requisitos técnicos de la externalización (RTE)

Aquellas condiciones que se exigirán al proveedor de SWF sobre la administración y operación de los diversos componentes y activos de TI relacionados con las aplicaciones delegadas, las mismas que también definen las responsabilidades del proveedor sobre el ciclo de DMS de estas aplicaciones. Los RTE conforman el AE.

2.3.5 Definiendo el término aplicación

Ciertamente una aplicación es un software, pero que para (Bente y otros, 2012) tiene estas características: proporciona funcionalidad al usuario final, esta funcionalidad esta lógicamente conectada (cohesiva), son unidades de despliegue con cierta autonomía, tienen un propietario que se encarga de su mantenimiento. Podemos complementar con lo indicado en TOGAF 9.1, que estas soportan los procesos de negocio de las compañías y que manipulan datos. (The Open Group, 2011).

2.3.6 SLCM, SLC y SDLC

El SLCM (Software Life Cycle Model) es un modelo de referencia que indica como usar los procesos de software bajo determinadas condiciones, define la secuencia de estos, las interacciones y los resultados esperados a alto nivel. Un SLCM no es una definición del proceso de desarrollo de software, este suele tener muchas más complejidad e incluye muchas actividades que no se describen en el modelo de ciclo de vida, tampoco es una metodología, pues no precisa reglas (Comer, 1991).

También hay que diferenciar entre el ciclo de vida del producto software (SLC) y el ciclo de desarrollo del software (SDLC), este abarca los procesos para capturar y especificar requisitos y para transformar estos en un producto software entregable, aquel incluye el SDLC más otros PS adicionales como los relacionados a los de mantenimiento que requiere el software y otros necesarios desde la puesta en uso hasta el retiro del producto software. Un SLC incluye variados SDLC que se llevan a cabo para mejorar y/o mantener el software (IEEE, 2004).

2.3.7 El correcto significado de metodología

También, queremos dejar precisada la definición de metodología como la especificación del proceso a ejecutar, generalmente como un conjunto de actividades, tareas y/o técnicas relacionadas, junto con qué productos de trabajo deben manipularse (crearse, usarse o modificarse) en cada momento y por quién, posiblemente incluyendo modelos, documentos y otras entradas y salidas (International Organization for Standardization, 2014). En otras palabras, las metodologías prescriben actividades en un orden, que se pueden desglosar en tareas, tienen entradas y salidas, productos de trabajo, algunas de los cuales son entregables, se ejecutan por roles predefinidos y deberían ofrecer guías y plantillas para generar los productos de trabajo que generan. Una metodología de DMS, no es la excepción. Tanto el SLCM y la metodología que lo implementa debieran ser parte del AE. Dependiendo del caso particular, pudieran plantearse más de un modelo en cuyo caso se deberá indicar cuándo se deberá optar por uno o por otro.

2.3.8 Marco metodológico para el DMS a usarse en la SWF

Por marco metodológico nos referimos al SLCM más adecuado para usarse por la SWF y a su aterrizaje en procesos de software (PS).

Este trabajo no tiene por finalidad proponer un SLCM o su correspondiente metodología, nuestro objetivo es plantear criterios de selección para que el adquirente opte por el SLCM que más le conviene aplicar según su realidad particular y en el contexto de una externalización de DMS de amplio plazo como la que nos ocupa.

2.3.9 Los principales SLCM y sus PS

Los SLCM van desde los puramente predictivos hasta los puramente adaptativos con variantes entre estos enfoques. Los modelos predictivos se caracterizan por lo detallado del desarrollo de requisitos, de la arquitectura de software y de la planificación del proyecto; casi no hay iteración entre las fases del proyecto, y los PS se ejecutan de manera lineal. Los adaptativos están diseñados para adecuarse a los requisitos de software emergentes, al ajuste iterativo de los planes y se caracterizan por ciclos de desarrollo iterativos. Algunas variantes, que fluctúan entre estos dos

modelos, producen incrementos de funcionalidad según cronograma planificado previamente, estos modelos están más cerca del lado predictivo, otros producen productos que se actualizan con frecuencia en varios ciclos de desarrollo, estos están más cerca del lado adaptativo (IEEE, 2004). La figura 3 ilustra lo indicado.

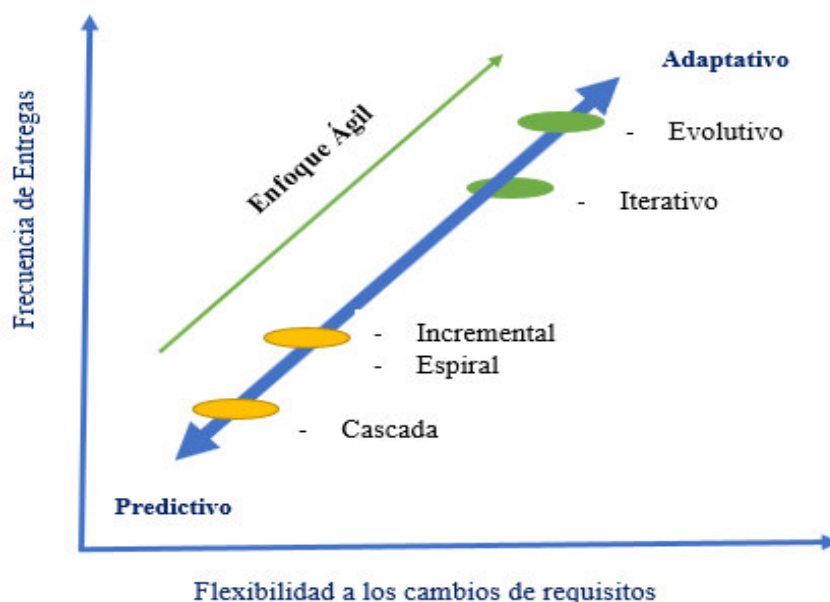


Figura 3. Variantes de los SLCM

Fuente. Basado en (Project Management Institute, 2017)

Los PS tienden a ser más detallados que los modelos, no pretenden seguir un orden en el tiempo, sino más bien pueden ordenarse según el modelo que se elija, los segundos brindan las fases del desarrollo, las actividades claves y sus interdependencias a alto nivel (IEEE, 2004). Los principales modelos y PS que se indican en el SWEBOK se muestran en la figura 4.

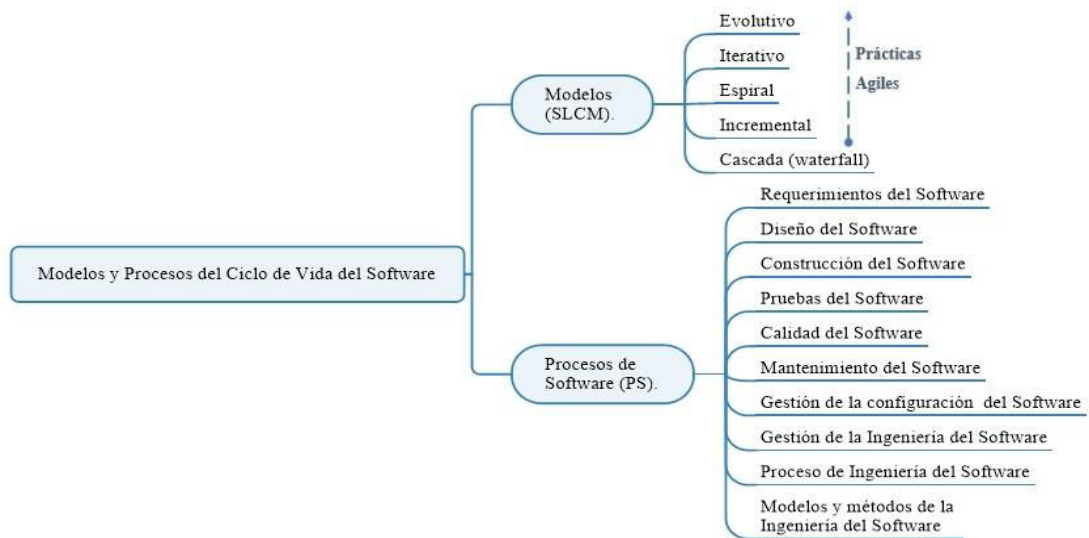


Figura 4. Modelos y Procesos de Software

Fuente. Contenido tomado de (IEEE, 2004)

2.3.10 Entendiendo los principales SLCM

El modelo predictivo a plenitud es el waterfall o cascada, donde los PS de requerimientos, diseño, construcción y pruebas, ocurren de manera secuencial; de hecho, cada uno de estos procesos representa una fase del modelo y para pasar de una a otra se requieren que cada una arroje productos de trabajo correctos antes de hacer la transición. En la realidad esto es difícil de alcanzar, salvo en cierto tipo de proyectos donde el alcance, plazos y costes se conocen de manera clara y exacta (International Organization for Standardization [ISO], 2017). Al realizar un desarrollo usando cascada existe una alta probabilidad de tener que rehacer el trabajo de etapas previas con el concerniente impacto en las variables de tiempo y costes del proyecto.

El modelo incremental propone construir una versión parcial del software, pero lista para su despliegue y utilización, e ir agregado gradualmente a esta mayor funcionalidad, implementando los requisitos que se pueden definir por adelantado entre incrementos (Comer, 1991). Para (IEEE, 2004) este modelo produce sucesivos incrementos de software funcionando, partiendo el conjunto de requisitos a implementarse por cada uno de los incrementos. Se puede ser muy riguroso con los cambios en los requisitos como en un modelo lineal o dar cierta flexibilidad a medida que evoluciona el producto software en cada incremento. Para (Project Management

Institute, 2017) el modelo incremental tiene por objetivo proporcionar entregables parciales pero operativos que se pueden poner en funcionamiento.

El modelo en espiral fue publicado por Barry Boehm en 1986 y fue diseñado para controlar los riesgos generales del proyecto. Propone ejecutar varios ciclos de desarrollo, llamados espirales o rondas, empezando por ronda cortas, con un alcance modesto para luego expandirse hacia afuera en espirales cada vez más amplias. En cada ronda se puede aplicar un enfoque de desarrollo tradicional como el de waterfall y entre cada una se realiza un análisis de riesgos (Conrad, 2011). Para (International Organization for Standardization [ISO], 2017) espiral es una variante del modelo incremental que ordena el desarrollo de la funcionalidad requerida en función del riesgo y propone implementar los requisitos más riesgosos en los primeros espirales y protegerse de costos inesperados que se presentan al final del SDLC.

En el modelo iterativo se hace una planificación inicial y luego se itera el proceso de creación de prototipos, pruebas, análisis, refinamiento de los requisitos y de la solución desarrollada; es decir se realizan repetidamente los procesos del ciclo de vida para ofrecer funcionalidad prioritaria cuanto antes, para luego refinarla en iteraciones posteriores y/o postergando la implementación de los requisitos más complejos para ciclos posteriores. (International Organization for Standardization [ISO], 2017). Para el (Project Management Institute, 2017) a diferencia del incremental, este modelo no ofrece entregas parciales al cliente, sino una única. En cierta literatura, no se considera a este modelo como tal, como en (IEEE, 2004), sino más bien que en general todos los PS deben ser iterativos en mayor o menor medida según como se apliquen.

El modelo evolutivo implica que no conocemos todos nuestros requisitos por adelantado, por lo que es útil experimentar con una solución de software operativa con los aspectos del sistema que se conocen mejor, un prototipo, para aprender de esta y evolucionarla (Comer, 1991).

El modelo evolutivo está destinado a tratar el conocimiento incompleto de los requisitos (International Organization for Standardization [ISO], 2017), en línea con (IEEE, 2004), quien considera que los modelos adaptativos permiten adaptarse a los requisitos de software emergentes y a ajustar los planes del proyecto iterativamente. Para (Comer, 1991) los modelos evolutivos extienden el concepto de desarrollo incremental al extremo, donde se crean numerosos prototipos que evolucionan o se

transforman en función de las necesidades del cliente. Un problema del método evolutivo es que es que no se puede planificar un plazo y un presupuesto con anticipación, otro es que un proyecto que lo aplica se puede expandir en demasiadas evoluciones, tornando complejo el control de sus variables, dado que el equipo y los usuarios pueden no ponerle un fin a las evoluciones, esto puede provocar que se pueda cortar el presupuesto de este tipo de proyectos, con lo que un prototipo no considerado un release final puede terminar siendo la solución implantada o que no se implante ninguno al considerar los creados como inadecuados.

Respecto a las prácticas ágiles , se pueden aplicar dentro de una variedad de modelos. Si bien los métodos ágiles podrían alcanzar su máxima intensidad de aplicabilidad dentro del modelo evolutivo, pudieran también aplicarse en los otros modelos (International Organization for Standardization [ISO], 2017), como se aprecia en la figura 4, donde la fecha punteada denota que las prácticas ágiles se pueden aplicar de menor a mayor intensidad en los modelos indicados.

Sintetizando los PS implementan al modelo y estos siempre va a tener que realizarse, sea con un modelo predictivo o adaptativo o cualquier variante, lo que va a cambiar es la forma de realizarse, p.ej. ejecuciones repetidas de los PS para diferentes incrementos del producto software. También hay que tener en cuenta lo indicado por (IEEE, 2004), que los PS engloban no sólo las actividades técnicas sino también las de gestión.

2.3.11 Un metamodelo para definir métodos de desarrollo de software

En este apartado nos referimos a UMA (Unified Method Architecture), un metamodelo de ingeniería de procesos que permite representar métodos, para ello plantea que estos se pueden esquematizar en 2 conjuntos: su contenido y su proceso, el primero está representado por los roles, tareas, productos de trabajo que se pueden usar en un método, el segundo es la forma como se toman los anteriores elementos y se organizan en una secuencia particular (IBM Corp, 2006), pudiendo variar el énfasis en su utilización según la índole del proyecto y la etapa de este en el cual se aplique. La figura 5 mostrada a continuación ilustra la propuesta de UMA para diseñar métodos.

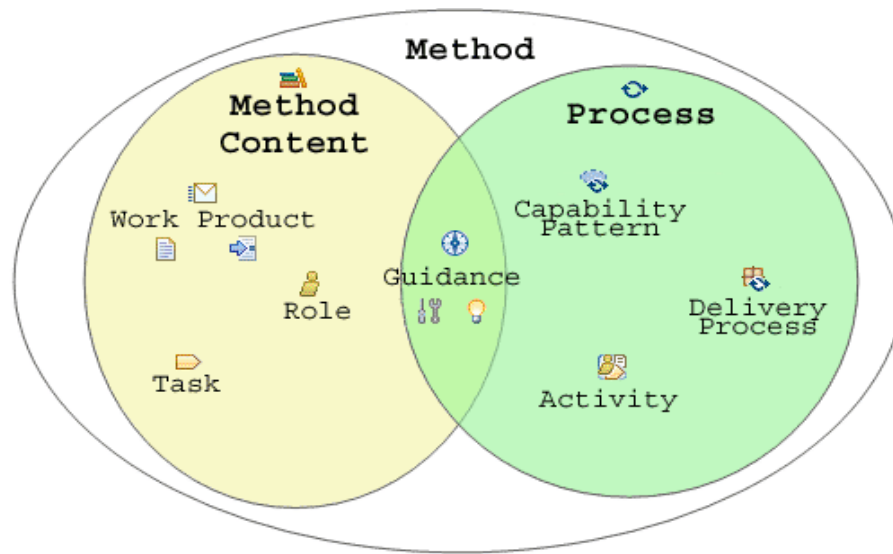


Figura 5. Posicionamiento de conceptos clave de UMA

Fuente. Tomada de (IBM Corp, 2006)

En el conjunto derecho de la figura previa se aprecia un elemento importante del proceso, la actividad, que en UMA hace referencias a elementos del contenido del método (tareas, roles y productos de trabajo), así por ejemplo en Rational Unified Process (RUP), que está basado en UMA, se tiene la actividad *realizar síntesis arquitectónica* que abarca tareas (como análisis arquitectónico, construir prueba arquitectónica de concepto y evaluar la viabilidad de la prueba de concepto arquitectónica), recibe productos de trabajo como inputs (p.ej. el modelo de análisis y el business case) , y crea o actualiza productos de trabajo como resultados (p.ej. crea la arquitectura de referencia); además esta actividad es ejecutada por roles que participan en ella (p.ej. el software architect y el system analyst). También del lado derecho tenemos el concepto de patrón de capacidad y proceso de entrega, sobre el primero es un conjunto de actividades en una secuencia particular que se ha demostrado permite lograr una capacidad u objetivo, p.ej. se puede tener un patrón de capacidad para realizar pruebas unitarias o para la gestión de requisitos, inclusive se pudiera tener un patrón de capacidad más amplio que abarque todas las actividades requeridas para una disciplina como la de ingeniería de requisitos o la de análisis y diseño. Sobre el segundo y último elemento del conjunto derecho, el proceso de entrega, el cual describe todas las fases y actividades requeridas desde el inicio hasta el final de un proyecto particular, es la implementación de un SCLM y que se puede

usar como metodología de referencia para proyectos con características similares. (IBM Corp, 2006). A manera de ilustración, en la figura 6, se muestra un proceso de entrega basado en RUP para proyectos pequeños. Si abrimos cada signo más se lograrán visualizar las actividades, subactividades y tareas.

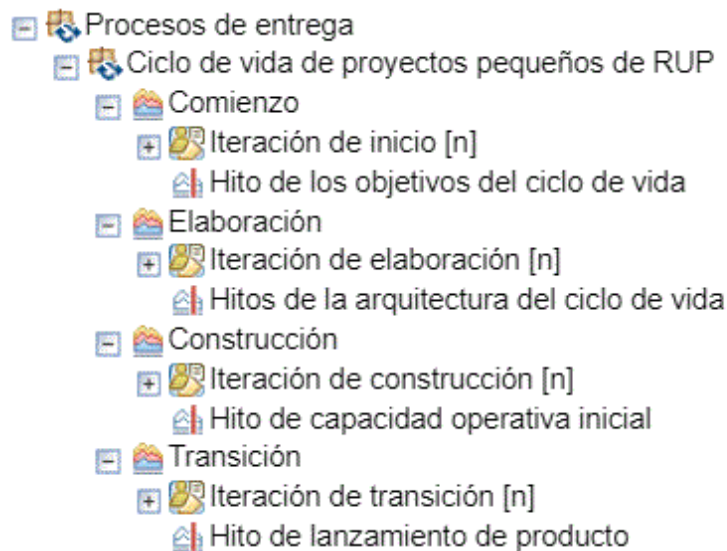


Figura 6. Vista de un proceso de entrega basado en UMA
 Fuente. Tomada de (IBM Corp, 2006)

Como resumen podríamos indicar que el lado izquierdo de la figura 5 representa la parte invariable o menos variable de un método, y el lado derecho la parte dinámica, que permite formar procesos a partir del method content (lado derecho) en función a las necesidades particulares de los definidores de procesos que representan al adquiriente, es por esto que hemos realizado este resumen de este metamodelo pues puede servir de ayuda para definir la metodología que implemente el SCLM que se elija para usarse en el contexto de la SWF. Para finalizar vale la pena mencionar que UMA está basado en la extensión SPEM e IBM Global Services Method (IBM Corp, 2006), entre otros (Dias & OliveiraJr, 2016).

2.3.12 key performance indicators (KPI) para aplicaciones

Los KPI de aplicaciones permiten cuantificarlas y poder con este número tomar decisiones respecto a ellas. En (Bente y otros, 2012) se proponen estos 4 KPI que resumimos a continuación:

Total cost of ownership (TCO): o costo total de propiedad, es el consolidado de todos los costos relacionados con una aplicación como mantenimientos de cualquier tipo, costos de la infraestructura que la soporta, pagos por licenciamiento y cualquier otro costo relacionado (Bente y otros, 2012).

Strategic fit (SF): o ajuste estratégico, este KPI va en una escala del 1 al 10 e indica hasta qué punto una aplicación se considera "heredada". Considera aspectos tanto tecnológicos como comerciales. Una aplicación obtiene una puntuación alta si encaja bien con la arquitectura empresarial to-be de la empresa, se basa en estándares y productos que se consideran preparados para el futuro y se encuentra en su mejor momento con respecto al ciclo de vida del software (Bente y otros, 2012).

Value contribution (VC), la contribución de valor refleja el valor comercial generado por la aplicación. Es poco común que se pueda asociar una cantidad de dinero como el aporte de una aplicación al negocio, por lo que en la mayoría de los casos, el VC será un valor sin unidades. VC refleja la importancia empresarial de la aplicación, así como el impacto de reemplazarla o retirarla en la organización, resume la importancia de la aplicación para los procesos de negocio de la cual es parte y los flujos de ingresos que están involucrados en estos procesos (Bente y otros, 2012).

Fan-in y fan-out of , estos KPI son medidas de complejidad procesal, así el fan-in de una aplicación APP cuenta los diferentes flujos de datos que ingresan a APP, estos flujos de datos pueden ser por ejemplo invocaciones de interfaz por otras aplicaciones, mensajes consumidos por APP, las tablas de bases de datos leídas por APP. El fan out, por otro lado, cuenta cuántos flujos de datos dejan APP. Estas pueden ser invocaciones de interfaz por parte de APP en otras aplicaciones, mensajes enviados por APP (Bente y otros, 2012). Estos KPI puede ayudar a encontrar un equilibrio entre distribuir funcionalidad a muchas aplicaciones (modularidad) o integrarla en una sólo aplicación, el primer escenario genere costes por cada interfaz, el segundo genera otros tipos de costes como los de hardware o de sincronizar los proyectos que actúan sobre

la aplicación que integra todo. Estos costos y el equilibrio buscado se aprecian en la figura 7, donde el eje y representa que tantas funciones tiene integradas una aplicación.

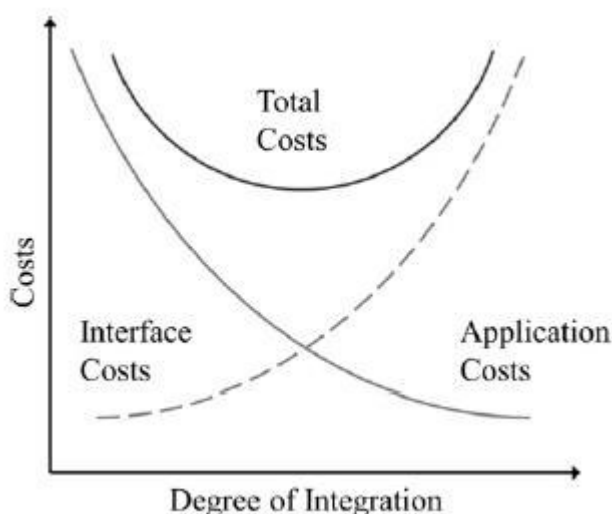


Figura 7. Costos de interfaces y aplicaciones

Fuente. Tomada de (Bente y otros, 2012)

2.3.13 Clasificación de aplicaciones

Cuando se externaliza aplicaciones, también es necesario categorizarlas para poder saber su importancia para el negocio, una forma de hacerlo es la propuesta por (Ward and Peppard, 2003) citado por (Bente y otros, 2012), la cual clasifica a estas en cuatro categorías, cuyas definiciones indicadas en Bente et al. parafraseamos:

The starts: Son aquellas que aseguran la operatividad actual y futura de la empresa, porque se utilizan en importantes procesos comerciales y permiten ofrecer productos y servicios innovadores que diferencian a la empresa de sus competidores. Las empresas tienen la expectativa de ampliarlas y mejorarlas aún más en el futuro, por lo que son parte del road map tecnológico de la compañía.

The Wild Cats: representadas por aquellas que tienen cierto potencial para negocios futuros, pero no contribuyen mucho a las operaciones comerciales actuales. Pueden ser prototipos que aún tienen que demostrar su utilidad y generalmente exploran cómo las nuevas tecnologías pueden ayudar a aprovechar nuevas oportunidades comerciales.

The Cash Cows: Sin estas aplicaciones toda la operativa actual de la empresa no pudiera ser posible, son su columna vertebral. Garantizan las operaciones centrales actuales y generan la mayor proporción de negocios provenientes de TI. Cualquier

interrupción de dicho sistema es una molestia pues generan ruido tanto para clientes externos como internos. Pero no representan una fuente de ingresos en el futuro y desde el punto de vista tecnológico se consideran sistemas heredados.

The Poor Dogs: Son aplicaciones sin importancia actual o futura, son residuos de alguna iniciativa anterior sin grandes beneficios o sistemas de apoyo en cierta medida necesarios pero que no representan una ventaja competitiva para la empresa.

2.3.14 Offshoring

Este término no es un sinónimo a los términos outsource u outsourcing que son los términos anglosajones equivalentes a externalización, sino que se refiere a la deslocalización de una función organizacional de una compañía en otro país, ya sea subcontratando el trabajo o manteniéndolo bajo el ámbito de la empresa (Salman y otros, 2017), generalmente esto se hace pensando en la reducción de costes o por algún otro motivo que permite una ventaja competitiva.

2.3.15 eSourcing

Son las externalizaciones que se soportan fuertemente en las tecnologías de la información para desarrollar la función externalizada, por ello también se les llama IT-enable sourcing (Hefley & Loesche, 2006), ejemplos de estas son los servicios de contact center, los pagos de planillas, o servicios de monitoreo de alertas de incidentes de ciberseguridad. En el eSourcing, parte del personal del adquiriente que antes realizaba la función delegada puede transferirse al proveedor, lo mismo que la infraestructura necesaria, pudiendo ocurrir lo opuesto si el adquiriente en el futuro decidiera recuperar el control. Los contratos de este tipo son de larga duración (Hyder y otros, 2006).

2.3.16 IDEF0

A continuación, vamos a comentar sobre este tipo de notación, porque es la hemos usado para modelar el framework que estamos proponiendo. IDEF0 (Integration DEFINition language 0), es una técnica para modelar basada en gráficos y texto combinados que se presentan de manera estructurada para ganar comprensión,

respaldar el análisis, proporcionar lógica para cambios potenciales, especificar requisitos o respaldar actividades de integración y diseño a nivel del sistema o área temática modelada. Un modelo IDEF0 se compone de varios gráficos que van mostrando de manera jerárquica niveles crecientes de detalle que describen funciones y sus interfaces dentro del contexto modelado (National Institute of Standards and Technology, 1993).

IDEF 0 a diferencia de sus variantes IDEF X permite representar el enfoque funcional, es decir las actividades a realizar desde un punto de vista (viewpoint) particular. Se empieza a modelar con IDEF0 creando primero un diagrama de contexto o padre, que está formado por un solo box o caja y sus correspondientes flechas ICOM (acrónimo de Input, Control, Output, Mechanism), luego se desglosa este en niveles crecientes de detalle en diagramas hijo o de descomposición. La figura 8 muestra la estructura de una caja en un diagrama de contexto y la figura 9 su correspondiente desglose (National Institute of Standards and Technology, 1993).

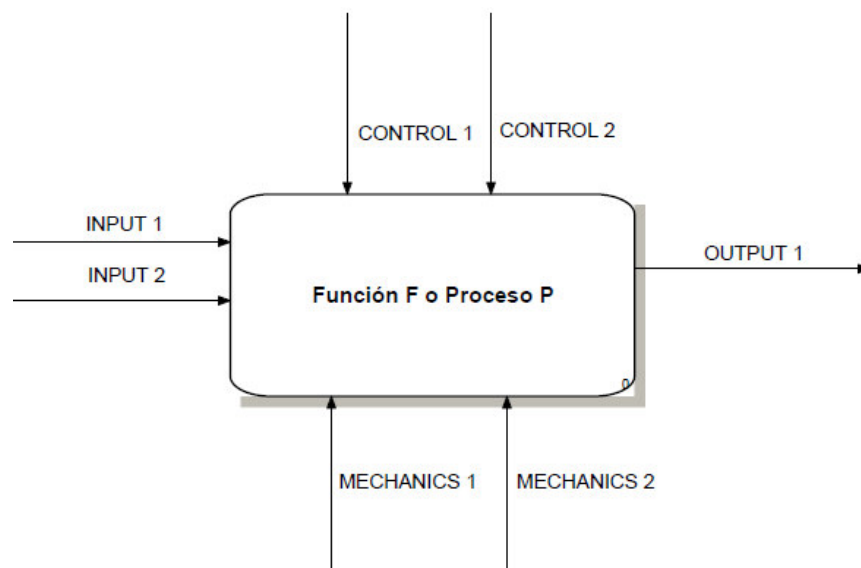


Figura 8. Caja o Box de una diagrama IDEF0
Fuente. Propia.

Las flechas ICOM indican lo siguiente, las del tipo input representan la entrada del proceso, es decir los datos u objetos que se transformaran en salidas y se asocian del lado izquierdo de la caja; las flechas control representan las condiciones necesarias para tener en cuenta para producir una salida correcta, acá tenemos por ejemplo normas, restricciones técnicas. Estas flechas se asocian a la parte superior de la caja;

las flechas de mecanismo representan los medios o recursos utilizados por el proceso actividad o función, se asocian a la parte inferior de la caja; finalmente las flechas output representan los datos, objetos u otro resultado del proceso, función o actividad (National Institute of Standards and Technology, 1993).

Las flechas descritas son la representación básica, pero tienen variantes, que no comentaremos en este documento porque no las hemos usado en este trabajo.

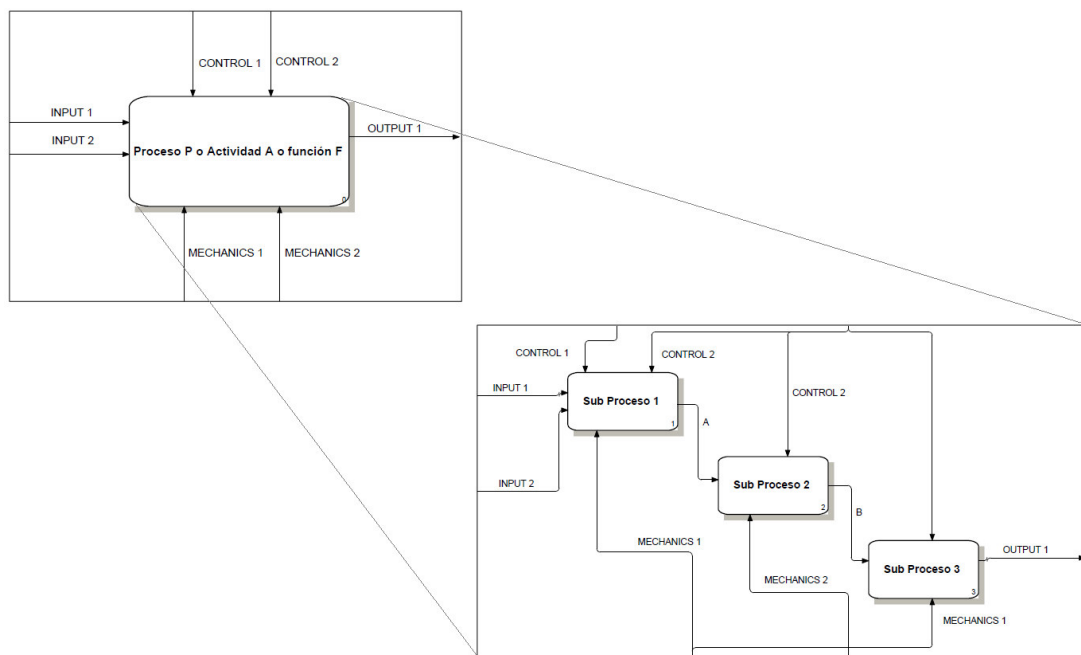


Figura 9. Diagrama padre e hijo en notación IDEF0

Fuente. Propia.

2.4 Estado del Arte

En (Azabache, 2022), un trabajo de investigación de este autor, exploramos que ofrece la literatura sobre el problema planteado en esta tesis. La revisión realizada es muy reciente con respecto a la fecha en que desarrollamos este trabajo por lo que sus resultados están vigentes. Esta se realizó siguiendo las pautas de (Cruz-Benito, 2016) y (Kitchenham & Charters, 2007) para revisiones de literatura. Seguidamente resumimos el proceso seguido, sus resultados y hallazgos encontrados.

2.4.1 De la planificación de la RSL

En esta etapa se planteó lo siguiente:

- Las preguntas de investigación en base al objetivo de la revisión que era el de determinar las publicaciones que proponen cómo elabora el AE para externalizaciones de la función de DMS.
- Se definió como fuentes de búsqueda Scopus, Web Of Science, Google Scholar y The SEI Library de la Carnegie Mellon University.
- Los criterios de selección (CS) a aplicarse a los resultados de las búsquedas.

2.4.2 De la de ejecución de la RSL

Aquí se realizaron las búsquedas y a estas se aplicaron los CS definidos en base a una secuencia que los aplicaba a cada set de resultados luego de leer títulos, resúmenes e introducción y contenido. En la tabla 1 se indican los resultados que se obtuvieron antes de aplicar los CS.

Tabla 1. Resultados iniciales de la RSL

Fuente	Cantidad de resultados
SEI Digital Library	204
Scopus	160
Google Scholar	129
WoS	85

Fuente. (Azabache, 2022)

Las publicaciones que superaron los CS fueron los de (Ikram y otros, 2018) , (POLO y otros, 2002) , (Phillips, 2011) y (Brownsword y otros, 2014), (Kronawitter y otros, 2013) y el (Hefley & Loesche, 2006) según lo indicado en (Azabache, 2022).

2.4.3 De los resultados de la RSL

En esta etapa se revisaron los datos relevantes del set de resultados de la ejecución y se sintetizaron estos para poder responder a las preguntas de la revisión. Las mejores respuestas pero no suficientes las brinda (Hefley & Loesche, 2006) y (Phillips, 2011),

la primera hace una revisión eSCM-CL, una de las formas del modelo eSourcing Model for Client Organizations (eSCM) realizado por la Universidad Carnegie Mellon a través de su Centro de Calificación de Servicios de Tecnología de la Información, que propone noventa y cinco prácticas posibles de aplicar en tercerización de servicios que usan intensivamente las TIC, las que están agrupadas en diecisiete capability areas (CA), las que a su vez están distribuidas en alguna fase de la externalización, que para el modelo son cinco que se indican en la figura 10.

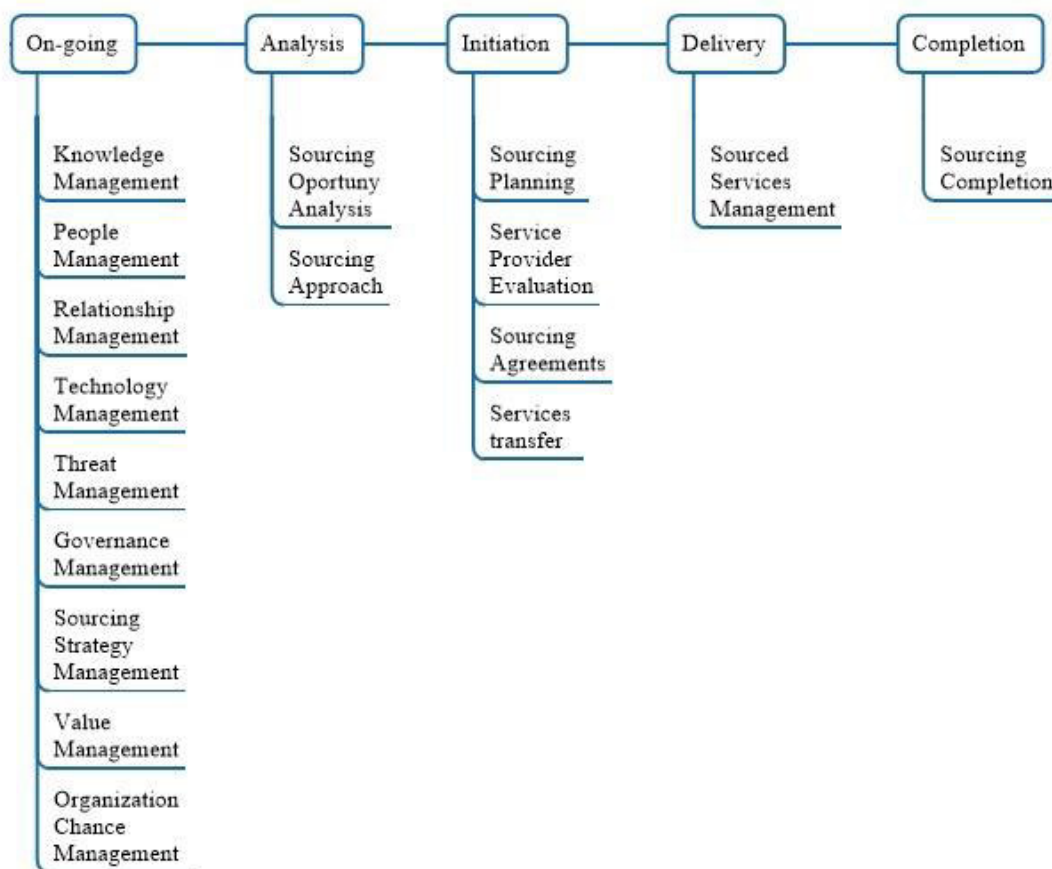


Figura 10. Fases y las CA de eSCM-C
Fuente. Tomado de (Azabache, 2022)

De estas CA, sourcing planing tiene entre sus cinco prácticas *services definition* que prescribe fijar con claridad los servicios delegados al proveedor y los niveles de servicio requeridos durante la fase inicial de ciclo de delegación del servicio, lo que es parte constitutiva de lo que acá hemos llamado el AE. La figura 11 permite ubicar esta práctica como parte de su CA.

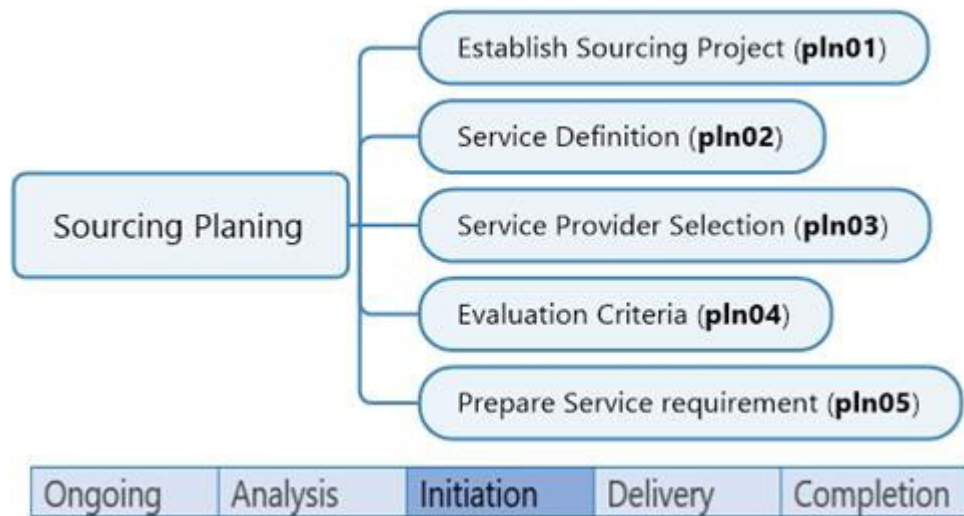


Figura 11. Prácticas del área de capacidad Sourcing Planing
 Fuente. Contenido tomado de (Hefley & Loesche, 2006)

La segunda publicación expone las mejores prácticas de CMMI-ACQ (CMMI for Acquisition) a manera de guía sobre cómo aplicarlas. Esta constelación de CMMI tiene seis áreas de las cuales la de desarrollar los requisitos de la adquisición, etiquetada como ARD se ocupa de capturar las necesidades del adquirente para transformarlas en requisitos contractuales. Las metas y prácticas genéricas de ARD se muestran en la figura siguiente.



Figura 12. Metas y prácticas genéricas del área ARD de CMMI-ACQ
 Fuente. Contenido tomado de (CMMI for Acquisition, Version 1.3, 2010)

Ninguna de estos estudios, indican el “cómo” llevar a cabo las prácticas de los modelos que explican, porque tales modelos son genéricos y no están instanciados para un tipo particular de tercerización, por eso indicamos inicialmente que se aproximan a lo que buscábamos en la RSL, prácticas para la externalización del tipo DMS.

Siendo así se llega a la conclusión de que al momento de la revisión hay escasa literatura que plantee o proponga formas o métodos o similares para lograr definir un adecuado AE cuando una compañía decide tercerizar la función de DMS, esta coincide con lo indicado por (Ikram y otros, 2018) y (Naciri, 2014) quienes indican que hay pocas publicaciones que planteen como enfrentar los aspectos más importantes de la externalización del DMS; en el mismo sentido (Ahmad y otros, 2017) concluye que en general para ITO los modelos de madurez están en desarrollo en la actualidad.

3 METODOLOGÍA

Presentamos aquí la metodología que hemos usado para demostrar que el framework que hemos propuesto ayuda a resolver el problema de investigación planteado, explicaremos como hemos corroborado nuestras hipótesis de investigación basándonos en un método adecuado, objetivo y validado por las técnicas estadísticas necesarias.

3.1 Tipo y diseño de la investigación

El tipo de investigación es cuantitativo, y el diseño cuasi experimental, que por definición consiste en operar una variable independiente con el fin de observar su resultado sobre la dependiente en una muestra no elegida al azar (no probabilística), sino que la misma se ha conformado previamente (Hernández Sampieri y otros, 2010). Este tipo de diseño se ajusta a nuestra estrategia para dar respuesta a las cuestiones de investigación planteadas y el cuál se soporta en una parte de la adaptación del The Method Evaluation Model (MEM) realizada por (Abrahamo Gonzales y otros, 2011), que permite predecir la probabilidad de que un método o modelo en particular sea aceptado en la práctica, basado en el esfuerzo de aplicarlo, la calidad de sus resultados y las percepciones de los usuarios sobre el modelo en evaluación. Llamaremos MEM Adaptado a los ajustes propuestos por Abrahamo et al., y para entenderlos tenemos que abordar los fundamentos de MEM, el cual es un modelo teórico para evaluar métodos de diseño de sistemas de información propuesto por (Moody y otros, 2002) cuyas constructos (variables relacionadas) están soportadas en el Technology Acceptance Model (TAM) de (Davis, 1989) , que ayuda con el pronóstico de que una nueva tecnología sea aceptada y/o adoptada por un grupo o por una organización. TAM es un modelo lo suficientemente validado y conocido en la industria. A continuación, explicaremos los constructos de MEM, el MEM Adaptado y la parte de este que aplicaremos a nuestro trabajo.

3.1.1 Constructos de MEM

MEM permite evaluar el éxito de un método tanto a nivel de su efectividad (eficacia y eficiencia) real, como el de su adopción en la práctica. En la siguiente figura se aprecia los constructos que forman parte de la teoría de este modelo, donde se aprecia que la percepción de utilidad (PU) está determinada por la percepción de facilidad de uso (PEOU), mientras que la intención de usar (IU) está determinada conjuntamente por la PEOU y la PU, también se aprecia que la eficiencia y efectividad reales determinarían la PEOU y PU respectivamente. En la gráfica los tres constructos centrales representan variables psicológicas internas, mientras que los restantes son variables conductuales externas que pueden medirse objetivamente (Moody, 2001; citado por Moody y otros, 2002).

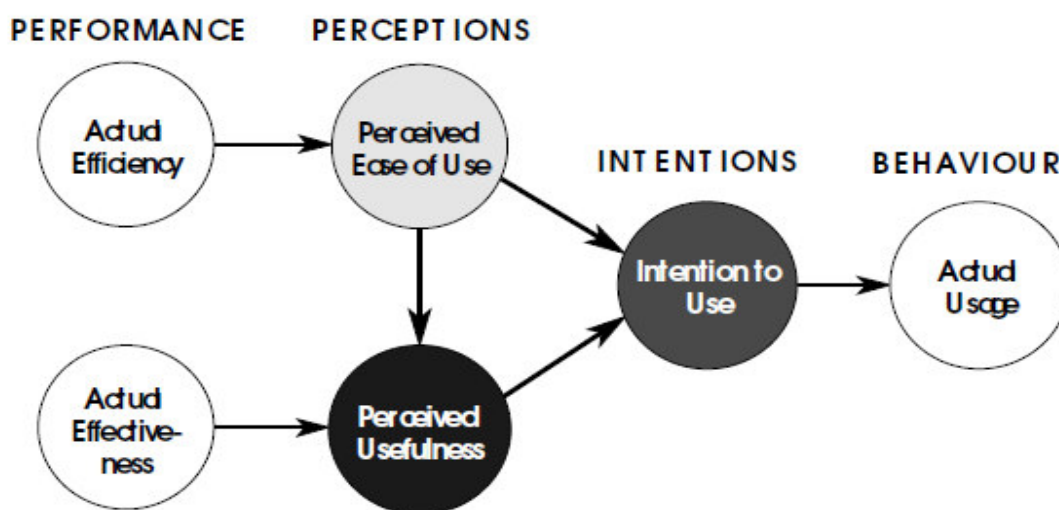


Figura 13. El Modelo de Evaluación de Métodos

Fuente. Tomado de (Moody y otros, 2002)

En la tabla 2, se indican las definiciones de las variables de MEM, las primeras tres variables miden el desempeño y las tres últimas la percepción sobre el modelo o framework (Moody y otros, 2002).

Tabla 2. Constructos de MEM

Conceptos	Variables	Definición
Actual Efficacy	Actual Efficiency (A-EFI)	El esfuerzo requerido para usar o poner en práctica el modelo o método.
	Actual Effectiveness (A-EFE)	El grado o nivel en el que un método o modelo logra sus objetivos.
Actual Usage	Actual Usage (AU)	la medida en que se utiliza un método o modelo en la práctica.
Perceived Efficacy	Perceived Ease of Use (PEOU)	El grado o nivel de esfuerzo que alguien cree o percibe le demandaría usar un método o modelo en particular.
	Perceived Usefulness (PU):	La creencia o percepción de una persona sobre el grado o nivel con que un método o modelo logrará sus objetivos.
Intention to Use (ITU)	Intention to Use (ITU)	La medida en que una persona tiene la intención de utilizar un método o modelo en particular.

Fuente. Contenido tomado de (Abrahamo Gonzales y otros, 2011).

3.1.2 Adaptación de MEM

La adaptación de Abrahamo Gonzales et al. no modifica los constructos del MEM original, si reformula los ítems del cuestionario de percepción de Moodys y plantea que para medir las variables de desempeño podemos medir la performance de una persona en poner en uso un método o modelo frente a cuestiones en una tarea de evaluación, así el tiempo que le tomaría en resolver lo podríamos usar para obtener la eficiencia y su tasa de éxito para la eficacia. Lo anterior lo fundamentan en que para dar una respuesta a una pregunta de resolución de problemas quién resuelve, debe razonar sobre el dominio que se representa en el modelo (Abrahamo Gonzales y otros, 2011).

3.1.2.1 Variables basadas en el desempeño en la adaptación de MEM

Abrahamo Gonzales et al. plantean medir el desempeño de una muestra de individuos, al resolver preguntas de un test enmarcadas en el dominio del modelo evaluado y usando este para resolverlas.

La A-EFE se determina midiendo la eficacia en la resolución, lo cual es una forma de medir la calidad del resultado del modelo o framework evaluado, verificando la comprensibilidad de este por los participantes. A-EFI Se determina midiendo el

esfuerzo requerido para comprender y aplicar el método, en este caso se considera la variable tiempo como determinante del esfuerzo.

3.1.2.2 Variables basadas en percepciones en la adaptación de MEM

Representadas por PEOU, PU e ITU, para medirlas Abrahao Gonzales et al. usan cuestionarios en escalas Likert tal cual el instrumento usado por MEM en (Moody y otros, 2002), pero cuyos enunciados de las preguntas fueron reformulados para que refirieran al método que evaluaban, lo que también haremos en este trabajo para nuestros fines.

En la figura siguiente se aprecia la relación entre los indicadores empíricos (ítems del cuestionario, los PEOU y PU de la gráfica) y sus constructos teóricos (las variables basadas en la percepción, representadas por los círculos de la gráfica).

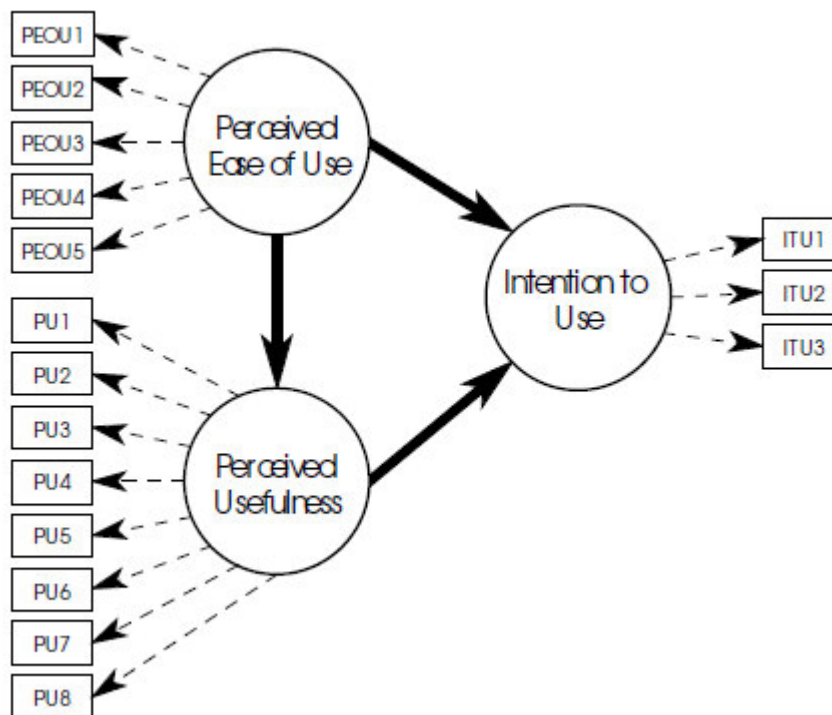


Figura 14. **Variables basadas en percepción del MEM**

Fuente. Tomado de (Moody y otros, 2002).

En el diagrama previo los círculos indican variables latentes o construcciones teóricas, los rectángulos representan las variables observadas (los elementos individuales de la encuesta), las líneas punteadas representan la relación de medición y las continuas las relaciones causales (Moody y otros, 2002).

En la figura 15 mostrada a continuación se incluyen las ocho hipótesis que suponen las relaciones entre las variables del modelo teórico.

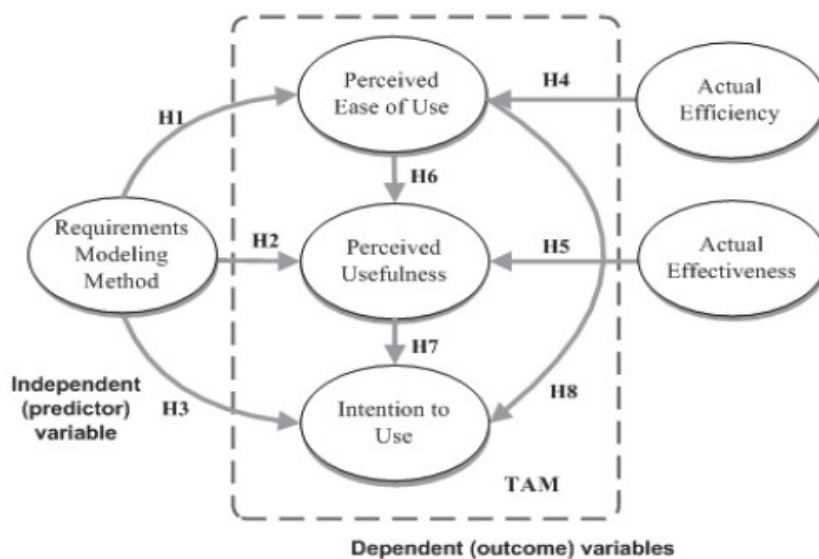


Figura 15. Constructos e hipótesis del modelo MEM
Fuente. Tomado de (Abrahamo Gonzales y otros, 2011).

Cada hipótesis de la figura previa trata de probar la relación de dependencia entre el par de constructos involucrados. En la tabla 3 se enuncian estas hipótesis haciendo referencia a un método, modelo o marco genérico.

Tabla 3. Hipótesis de MEM

Hipótesis nula	Enunciado
H1₀	El método, modelo o framework es percibido como difícil de usar. $H_{11} = \neg H_{10}$
H2₀	El método, modelo o framework es percibido como no útil. $H_{21} = \neg H_{20}$
H3₀	No hay intención de usar el método, modelado o framework en el futuro. $H_{31} = \neg H_{30}$.
H4₀	La percepción de la facilidad (PEOU) no es consecuencia de la eficiencia de comprensibilidad (AEFI). $H_{41} = \neg H_{40}$.
H5₀	La percepción de utilidad (PU) no es consecuencia de la eficacia de la comprensibilidad (AE). $H_{51} = \neg H_{50}$.
H6₀	La PU no es consecuencia de la PEOU. $H_{61} = \neg H_{60}$.
H7₀	La intención de uso (ITU) no es consecuencia de la PU. $H_{71} = \neg H_{70}$.
H8₀	La ITU no es consecuencia de la PEOU). $H_{81} = \neg H_{80}$.

Fuente. Contenido tomado de (Abraham Gonzales y otros, 2011).

3.1.3 Aplicación de la adaptación de MEM

De las hipótesis de MEM indicadas en la tabla 3, las que necesitamos evaluar son H1, H2 y H6 enunciadas así:

H1₀: El framework propuesto se percibe como difícil de usar. Con su hipótesis alternativa $H_{11} = \neg H_{10}$.

H2₀: El framework propuesto se percibe como no útil. Con su hipótesis alternativa $H_{21} = \neg H_{20}$.

H6₀: La PU del framework no es consecuencia de la PEOU.

Para tal efecto replicaremos parcialmente los mecanismos de la adaptación de MEM, para evaluar las variables de desempeño A-EFE, A-EFI y las de percepción PEOU y PU. A-EFE lo calcularemos como el promedio de las respuestas correctas dividido por el número total de preguntas; A-EFI como el promedio de las respuestas correctas dividido por el tiempo dedicado a proporcionar las respuestas. PEOU y PU como la media de los ítems correspondientes de la encuesta de percepción que se muestran más adelante.

3.2 Unidad de análisis

Nuestra unidad de análisis prevista es el profesional de TI con trayectoria en áreas de desarrollo de software y/o servicios TI.

Definir la unidad de análisis es determinar qué o quién va a ser estudiado (empresas, individuos, eventos, etc.) y esto en función a las preguntas de investigación. A partir de la unidad de análisis se determina la población y un subgrupo de esta, la muestra (Hernández Sampieri y otros, 2010). Para este trabajo hemos definido al respecto lo siguiente:

3.3 Población de estudio

De la unidad de análisis nos interesan como población los profesionales de TI con cinco o más años de experiencia en roles como especialistas o gestores de proyectos, especialistas o gestores de servicios TI, analistas de proyectos o de servicios TI, en empresas que brindan o contratan servicios de DMS; esto porque es necesario un nivel mínimo de experiencia en el área de ITO para poder considerar el framework que se propone para su evaluación vía los instrumentos de medición planteados.

3.4 Tamaño de la muestra

Dado que la investigación es cuasi experimental y que se requiere por la naturaleza de este trabajo una muestra dirigida o no probabilística, no se necesita determinar una muestra representativa. esta elección obedece que para nuestro caso de estudio se requiere la elección de una muestra con ciertas características específicas (conocimiento y experiencia). Como indica (Hernández Sampieri y otros, 2010) elegir los ítems de este tipo de muestras no depende de la probabilidad ni de formula, sino de motivos propios de la índole de la investigación o de la toma de decisión del o los investigadores.

3.5 Diseño del experimento

Para (Babbie, 2014) citado por (Hernández Sampieri y otros, 2010) un “experimento se refiere a “elegir o realizar una acción” y después observar las consecuencias”, en

nuestro caso hemos explicado nuestro framework a una muestra de individuos, les hemos aplicado dos instrumentos de evaluación (acciones realizadas) y hemos evaluado los resultados de estos test (para observar las consecuencias). A continuación, resumimos como realizamos lo anterior.

3.5.1 Selección de sujetos para el experimento.

Se le pidió a un grupo de profesionales de ingeniería de tecnologías de la información elegidos según la población de estudio definida, participar en la sesión explicativa del framework, en una tarea de evaluación y encuesta anónima sobre este. Los profesionales invitados trabajan en la industria de TI laborando para el sector financiero en su gran mayoría, sector donde el ITO es muy intensivo.

3.5.2 Variables

Las variables dependientes son las que plantea MEM y que describimos en términos de (Moody's y otros, 2002):

- Medidas basadas en desempeño (A-EFE y A-EFI): que busca medir qué tan bien realizaron los sujetos las preguntas de la tarea de evaluación. Miden la efectividad real.
- Medidas basadas en la percepción (PEOU y PU): que mide qué tan efectivo percibieron los sujetos el marco de evaluación. Miden la efectividad percibida.

Todas estas variables se describieron en la tabla 2, Constructos del MEM, y se medirán a través de la tarea de evaluación (las de desempeño) y de la encuesta inmediata posterior (las de percepción). Los ítems PEOU1...,PEOU4 y PU1...,PU7 que permiten medir los constructos PEOU y PU respectivamente se describen en la tabla 4.

Las variables independientes están representadas por:

- Los resultados de la tarea de evaluación por cada sujeto, que permite medir las variables dependientes de desempeño. Esto se ilustra en la figura 15, constructos e hipótesis del modelo MEM.
- Las respuestas a la encuesta aplicada de forma inmediata posterior a la tarea de evaluación, permite medir las variables dependientes de percepción. Esto se ilustra en la figura 14, variables basadas en percepción del MEM.

También hay que tener en cuenta que el MEM propone vínculos causales entre parejas de constructos, estas relaciones se dan entre las variables dependientes como por ejemplo las variables basadas en rendimiento sobre las variables basadas en percepciones, o que estas últimas influyan en la intención de uso (Abrahamo Gonzales y otros, 2011).

3.5.3 Tratamiento experimental

El experimento tuvo las siguientes actividades, en esta secuencia:

- Exposición a los participantes sobre el tipo de externalización que abarca el framework, sus ventajas, sus retos, las aristas a considerar cuando se externaliza, sus desventajas.
- Exposición del framework desarrollado a los participantes, como su ámbito de aplicación, sus objetivos, su finalidad y los procesos que lo conforman.
- Aplicación de la tarea de evaluación, donde se pidió a los sujetos que respondieran, ayudándose del framework explicado, un conjunto de siete preguntas objetivas sobre aspectos técnicos que considerarían como parte del alcance de la externalización de un escenario planteado. Esta actividad se usó para recopilar datos con los que medir las variables basadas en el desempeño. No sólo se recopilaron las respuestas sino la hora-minuto de inicio-fin que le llevo a cada participante proporcionar sus respuestas. Esto porque la variable tiempo la requerimos para calcular la variable A-EFI.
- Aplicación de la encuesta con el fin de evaluar la percepción sobre framework. Esta tuvo once ítems y se usó para recopilar datos con los que medir las variables basadas en percepción PEOU y PU ya definidas.

3.5.4 Instrumentos

Tenemos las ya mencionadas tarea de evaluación para medir el desempeño en la comprensibilidad del framework y la encuesta para medir la percepción sobre el framework, la cual fue del tipo escala de Likert de cinco puntos. La tabla 4 mostrada a continuación muestra los ítems de esta que se usaron para medir las variables basadas en percepción (PEOU y PU).

Tabla 4. Ítems de la encuesta tipo Likert para medir las variables de percepción

Ítem	Enunciado del Ítem
PEOU1	El framework es simple y fácil de entender.
PEOU2	Los procesos del framework son entendibles y fáciles de aplicar.
PEOU3	Con el framework fue fácil para mi entender que aspectos técnicos tengo que considerar para definir el alcance de la externalización si soy parte de un equipo de tercerización.
PEOU4	En general los procesos del framework sirven como guía para definir los requisitos técnicos de una externalización completa del desarrollo y mantenimiento de software hacia un proveedor de software factory.
PU1	Creo que el framework reducirá el tiempo necesario requerido para definir el AE del DMS, en comparación al que se necesitaría de no usarlo.
PU2	En general encontré útil el framework.
PU3	Creo que el framework es útil para ayudar a definir el AE técnico de la tercerización del DMS a una software factory.
PU4	Considero que el framework ayuda a definir un alcance correcto de la tercerización del DMS, pues ayuda a considerar los aspectos técnicos necesarios.
PU5	Creo que este Framework considera los aspectos técnicos necesarios para tener en cuenta para definir el AE de la función de DMS hacia una software factory.
PU6	En general, creo que este framework proporciona un medio eficaz para determinar qué externalizar y cómo cuando una empresa delega la función de desarrollo de software hacia una software factory.
PU7	Si tuviera el encargo de liderar a un equipo encargado de definir el AE del DMS en una compañía, creo que el uso del framework mejoraría mi desempeño para identificar los ítems o componentes a externalizar y las condiciones en que debería delegarse a la software factory.

Fuente. Propia.

3.5.5 Operación del experimento

3.5.5.1 Ejecución

Todas las actividades indicadas en el apartado 2.9.3 se realizaron de manera remota, las exposiciones vía MS Teams y la toma de los instrumentos de evaluación vía un vínculo a cada uno creado desde la herramienta class marker, que permitió capturar las respuestas de manera anónima, además del momento de inicio y fin de cada test.

3.5.5.2 Registro de datos

Los datos fueron recopilados en MS Excel, donde se verificó con fines de depuración que no hubiera tareas o encuestas respondidas incompletamente, asegurando así la integridad de las evaluaciones, luego fueron ingresados a IBM SPSS Statistics donde se realizaron la mayoría de los cálculos que nos llevó a los resultados que explicaremos en el capítulo cuatro.

4 Framework propuesto

En este capítulo presentamos el framework propuesto al que hemos llamado SWFrame y cuyos procesos se muestran en la figura 16, en esta se aprecia que tiene dos macroprocesos: (1) Determinar los requisitos técnicos de la externalización y (2) Definir un marco metodológico de DMS para SWF, y por cada uno se indican los procesos que los componen. Para modelar estos procesos se usó la notación IDEF0 (Integration Definition for Function Modeling), que permitió tener un enfoque top down, partiendo de un diagrama de contexto o diagrama cero tanto para (1) como para (2), para luego mostrar los procesos que los conforman mediante un diagrama de descomposición, que se describe y complementa con prácticas recomendadas para usar en cada uno de ellos. En el apartado 2.3.16 del marco teórico se describió esta notación.

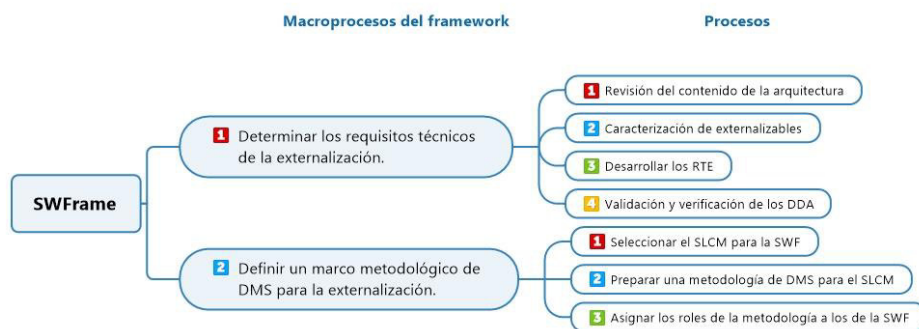


Figura 16. Procesos de SWFrame

Fuente. Elaborada por el autor

En el estado de arte se indicó que los modelos existentes para gestionar externalizaciones ofrecen sólo prácticas genéricas, SWFrame ofrece procesos con prácticas específicas para el caso particular de la externalización del DMS, para ayudar a las organizaciones adquirientes a definir los RTE que conforma el AE necesario para poner en marcha la externalización.

4.1 Macroproceso uno (macropro-1): Determinar el alcance técnico de la externalización

Busca lograr el objetivo específico 01 de nuestro trabajo:

OE01: Definir procesos del framework que permitan determinar los requisitos técnicos de la externalización bajo las cuales se debe delegar a la SWF la función de DMS de la organización adquiriente.

4.1.1 Diagrama de contexto y descripción de macropro-1.

El diagrama de contexto para este proceso se muestra en la figura 17 y la descripción de sus elementos en la tabla 5, lo que se muestra a continuación.

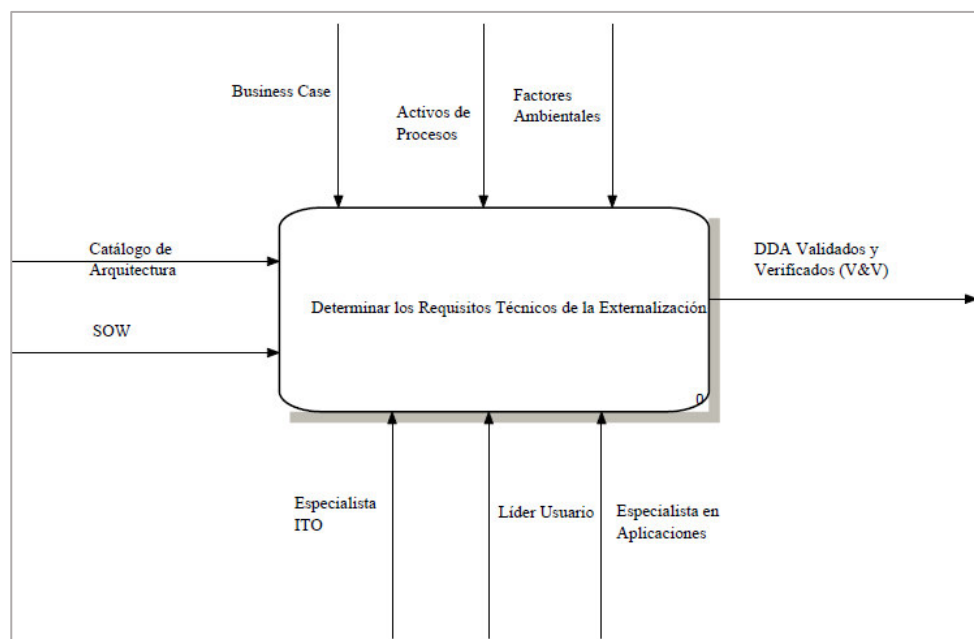


Figura 17. Diagrama de contexto para el macroproceso 01 de SWFrame
Fuente. Elaborado por el autor

Tabla 5. Descripción de los elementos del macropro-1

Nombre del macroproceso	
Determinar el alcance técnico de la externalización.	
Propósito	
Ayudar a identificar los ítems a externalizar, con la finalidad de definir los requisitos técnicos bajo los cuales se delegan a la SWF, los primeros son aquellos componentes de los sistemas de software que deben ser parte de la externalización del DMS, los segundos las condiciones técnicas sobre los primeros que la SWF está obligada a cumplir. Su resultado es el set de DDA (Documento de Definición de Alcance) verificado y validado por los roles indicados.	
Punto de vista	
Equipo del proyecto del adquiriente conformado por los roles que participan en cada proceso.	
Entradas	
SWO	Statement Of Work (SWO), es creado por el adquiriente en la etapa de prefactibilidad, describe las necesidades y el alcance de la externalización a alto nivel. p.ej. podría contener una sentencia como la siguiente, se externalizará el DMS de todos los aplicativos que soportan los procesos de compras, ventas y distribución de la compañía.
Elementos de Control	
Catálogo de arquitectura	Es la documentación de los diferentes dominios de arquitectura empresarial relacionados con las aplicaciones cuyo DMS se está externalizando. Es el contenido de la arquitectura empresarial, conformada por los dominios de procesos, aplicaciones, datos e infraestructura.
Business Case	Documento que contiene la justificación de porque el proyecto se realiza en términos de los beneficios esperados, delimita el alcance, contiene estimaciones de plazos y costes, e identifica riesgos del proyecto de externalización.
Activos de procesos	Agrupar normas, procedimientos, metodologías de la compañía que se deben tener en cuenta al ejecutar este proceso.
Factores ambientales	Se refiere a los factores del entorno como la cultura, sistemas existentes de administración, estructura organizacional y otros de la organización adquiriente.
Recursos	
Especialista ITO	Especialista en Information Technology Outsourcing o quien haga sus veces.
Especialista en Aplicaciones	Este rol se corresponde con el especialista que tiene el know how técnico en alguna de las aplicaciones que se externaliza. Se debe contar con uno por cada aplicación delegada.
Líder Usuario	Representante del área usuaria de la aplicación que se delega. Se debe contar con uno por cada aplicación delegada.
Salidas	
DDA V&V	Set de DDA validados y verificados (V & V) por los roles indicados, así como también se debe indicar que rol de la SWF es el responsable de los requisitos definidos en cada DDA.

Fuente. Elaborado por el autor.

4.1.2 Diagrama de descomposición del macropro-1.

Para este macroproceso su diagrama de descomposición se indica a continuación en figura 18, en ella apreciamos que está conformado por cuatro procesos que describimos en los siguientes ítems.

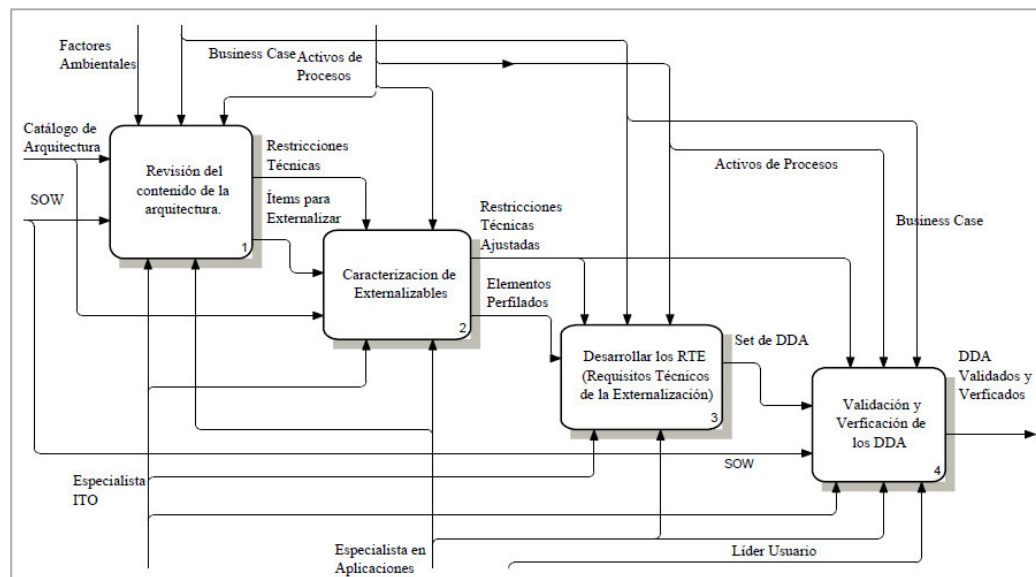


Figura 18. Diagrama de descomposición del macropro1 de SWFrame
Fuente. Elaborada por el autor

4.1.3 Procesos del macroproceso 01

A continuación se describe cada proceso del macroproceso 01 y seguidamente las prácticas recomendadas por cada uno de ellos.

4.1.3.1 Revisión del contenido de la Arquitectura

Tabla 6. Descripción del proceso revisión del contenido de la arquitectura

Nombre del proceso	
Revisión del contenido de la Arquitectura	
Propósito	
Revisar el catálogo de arquitectura empresarial (AE) para revisar aspectos de sus diversos dominios que tengan relación con las aplicaciones que se van a externalizar con la finalidad de identificar aquellos ítems o componentes que deben ser delegados por ser parte de o interactuar o ser soporte de estas aplicaciones. También se deben identificar restricciones técnicas asociadas.	
Entradas	
SWO	Ya descritos.
Catálogo de Arquitectura.	
Elementos de Control	
Business Cases	Ya descritos.
Activos de Procesos	
Factores ambientales	
Recursos	
Especialista ITO	Ya descritos.
Líder usuario	
Especialista en Aplicaciones	
Salidas	
Ítems para externalizar (IE)	Aplicaciones, interfaces o middlewares, servidores de aplicaciones, servidores físicos o en nube, software base, entornos de desarrollo, las API, microservicios y cualquier otro ítem que debe ser externalizado para que la SWF pueda cumplir su función de DMS.
Restricciones técnicas	Limitaciones o condiciones técnicas que deben tenerse en cuenta al externalizar los IE, p.ej. disponibilidad de licencias para un determinado framework de desarrollo.

Fuente. Elaborado por el autor.

4.1.3.2 Caracterización de externalizables

La descripción se muestra en la tabla siguiente:

Tabla 7. Descripción del proceso caracterización de externalizables.

Nombre del proceso	
Caracterización de externalizables	
Propósito	
El objetivo es realizar un perfilado de cada elemento de cada sistema de software a externalizar, identificando su función, sus características técnicas, sus requisitos para operar, el área y rol que lo mantiene, entre otras características. Hacemos énfasis en que la aplicación por sí misma es parte de un sistema que en conjunto con sus otras partes logra que esta funcione según su propósito.	
Entradas	
Ítems para externalizar	Ya descrito.
Elementos de Control	
Restricciones técnicas	Ya descritos.
Activos de procesos	
Recursos	
Especialista ITO	Ya descritos.
Especialista en aplicaciones	
Salidas	
Elementos perfilados	Perfilado de cada elemento del sistema del software a externalizar, identificando su función, sus características técnicas, sus requisitos para operar, el área y rol que lo mantiene, entre otras características
Restricciones técnicas ajustadas	Son las restricciones actualizadas de corresponder.

Fuente. Elaborado por el autor.

4.1.3.3 Desarrollar los requisitos técnicos de la externalización (RTE)

La descripción se muestra en la siguiente tabla:

Tabla 8. Descripción del proceso Desarrollar los RTE

Nombre del proceso	
Desarrollar los requisitos técnicos de la externalización (RTE)	
Propósito	
Tiene por objetivo especificar los RTE por cada ítem perfilado y teniendo en cuenta las restricciones técnicas identificadas en el proceso previo. Los RTE determinan las condiciones bajo las cuales se delega al proveedor de SWF los diversos elementos que administrará y mantendrá. La salida del proceso es el conjunto de estos RTE agrupados por ítems o componentes.	
Entradas	
Elementos Perfilados	Ya descritos.
Elementos de Control	
Restricciones Técnica Ajustadas.	Ya descritos.
Business Case	
Activos de Procesos	
Recursos	
Especialista ITO	Ya descritos.
Especialista en Aplicaciones	
Salidas	
Set de DDA	Set de Documentos de Definición de Alcance (DDA) conteniendo los RTE especificados, agrupados por cada uno de los ítems o componentes a externalizar identificados.

Fuente. Elaboración propia.

4.1.3.4 Validación y verificación de los DDA (V&V de los DDA)

La descripción se muestra en la siguiente tabla.

Tabla 9. Descripción del proceso V&V de los DDA

Nombre del proceso	
Validación y verificación de los DDA	
Propósito	
Se analizan los RTE de cada DDA definidos en el proceso previo para asegurarse que son los necesarios y suficientes (validación), pudiendo hacerse los cambios necesarios. También se establecen requisitos de verificación que permitan comprobar el cumplimiento de los RTE. Se propone también asignar a los RTE a roles de la SWF que deben ser los responsables de su cumplimiento. El especialista ITO, de aplicaciones y el Líder Usuario deben realizar este proceso.	
Entradas	
Set de DDA	Ya descritos.
SOW	
Elementos de Control	
Restricciones ajustadas.	Ya descritos.
Business Case	
Activos de procesos	
Recursos	
Especialista ITO	Ya descritos.
Especialista en Aplicaciones	
Líder usuario	
Salidas	
DDA validados y verificados	Set de DDA con sus RTE validados y complementados con los requisitos o criterios de verificación definidos, así como también con el rol de la SWF responsable de cumplimiento.

Fuente. Descripción propia.

4.1.4 Prácticas sugeridas para cada proceso del macropro-1

Por cada proceso se indican prácticas sugeridas para realizar y lograr el resultado o salidas esperadas. Aun cuando cada realidad es particular, realizar estas actividades contribuirá al objetivo esperado.

4.1.4.1 Prácticas sugeridas para el proceso revisión del contenido de la arquitectura

El objetivo de la revisión arquitectónica es conocer la composición y el ecosistema del software cuyos desarrollos se desea delegar a la SWF, para identificar los componentes que lo soportan y con que otros sistemas se comunica para determinar si es que estos también se deben externalizar y de qué manera. En la delegación del DMS no basta sólo con indicar el nombre y la versión del lenguaje de programación con el que fue construido el software o su manejador de base de datos, hay otras entidades necesarias para que este funcione, cuyo tratamiento y responsabilidades de las partes en un contexto de externalización tiene que ser definido en el AE. Deben participar un especialista por cada software que se externaliza, esto garantiza tener a un conocedor que pueda esclarecer las dudas técnica sobre la aplicación, un especialista en tercerizaciones de TI (ITO) y por supuesto el área usuaria representada por su líder usuario, que garantiza el conocimiento del o los procesos de los cuales las aplicaciones son parte. El Project manager del proyecto debe ser el orquestador de estos roles.

Para realizar la revisión planteada proponemos realizar lo que de (The Open Group, 2022) llama una evaluación de la capacidad arquitectónica actual de la organización, y para nuestro fines enfocada principalmente al dominio de aplicaciones, aunque tocará también en lo necesario extender la revisión a los otros dominios; procesos, data e infraestructura; esto permitirá revisar entre otros ítems el catálogo de aplicaciones, el catálogo de interfaces, la relación entre las aplicaciones y las unidades organizacionales, descubrir las tecnologías de las aplicaciones, su arquitectura interna y más. La magnitud de esta tarea puede escalar si es que la organización adquiriente no tiene documentada su arquitectura o si esta está incompleta, siendo un prerequisite para este proceso, que precede a los otros, tener completo el catálogo de arquitectura. Las prácticas propuesta son:

1. Revisar el catálogo de la cartera de aplicaciones.

Ayudará a determinar todos los componentes que forman parte del sistema al cual la aplicación a externalizar pertenece. Conviene iniciar esta revisión categorizando las aplicaciones según los criterios indicados en el ítem 2.3.13 clasificación de aplicaciones del marco teórico.

2. Elaborar una matriz de aplicación- unidad organizacional.

El objetivo es saber qué aplicaciones son consumidas por qué unidades organizacionales, esto permitirá conocer no sólo qué unidad es la dueña de la aplicación sino requisitos de soporte, brechas o deuda técnica con relación a determinada aplicación, además del set de aplicaciones usado por una unidad organizacional (The Open Group, 2022).

3. Elaborar una matriz de aplicación-función

El objetivo es saber qué funciones de negocio provee qué aplicación, esto permitirá entre otras cosas determinar qué aplicaciones están asociadas con qué procesos de negocio (críticos, estratégico o de otra índole) y no obviar alguna que deba también delegarse (The Open Group, 2022).

4. Revisar el catálogo de interfaces.

Permitirá evaluar las dependencias entre aplicaciones, es decir es necesario determinar cómo se comunican las aplicaciones entre sí para escribir, leer, actualizar y eliminar datos, identificando las API, servicios web, archivos por lotes u otras formas de comunicación (The Open Group, 2022).

5. Determinar el Fan in y el Fan out de cada aplicación a externalizar

En el punto 2.3.12 KPI para aplicaciones del marco teórico se explicó cómo obtener el fan-in y el fan-out. Para nuestros fines esto nos da una idea de la complejidad procesal de la aplicación y así poder categorizarla en consecuencia, también el ejercicio de calcularlos nos ayuda a identificar a los elementos que se comunican con una aplicación y así poder identificar interfaces no documentadas.

6. Describir las interfaces identificadas.

Para el caso de las interfaces identificadas se deberán caracterizar indicando a que aplicaciones comunica, la información y el mecanismo de operación.

7. Identificar aplicaciones adquiridas.

El objetivo es determinar que aplicaciones tienen limitaciones contractuales para su mantenimiento, dado que su código fuente es propiedad de un tercero. Identificarlas es importante para definir el tratamiento que se les dará

8. Elaborar un listado con una descripción de cada ítem que debe ser externalizado.

Hablamos de aplicaciones, interfaces, servidores de aplicaciones, incluso servidores físicos o instancias cloud, u otros elementos o componentes que necesitan ser externalizados en conjunto con cada aplicación.

9. Identificar el costo total de una aplicación (TCO- Total cost of ownership).

Esto permitirá identificar en que rubros una aplicación genera costes, ayudando así a identificar elementos que deben considerarse para la externalización. En el punto 2.3.12 KPI para aplicaciones del marco teórico se explicó el concepto de TCO.

10. Elaborar un listado de restricciones técnicas.

Por cada ítem a externalizar identificar y documentar las restricciones técnicas relacionadas, p.ej. el número de licencias disponibles de un framework de desarrollo, que necesita funcionar sobre un sistema operativo que ya no tiene soporte del fabricante, que no existe documentación técnica. En todas las prácticas previas podemos identificar restricciones a considerar.

11. Revisar los procesos de gestión de la configuración

Este proceso ayudará a determinar qué elementos intervienen en el versionado del código fuente.

12. Revisar e identificar componentes en los pipes o tuberías de los flujos DevOps o DevSecOps para los procesos CI y CD.

Revisar los procesos Continuos Integration (CI) y Continuos Delivery (CD) para las aplicaciones que pasan por ellos ayudará a identificar otros componentes y también restricciones técnicas a considerar.

13. Revisar si cada aplicación tiene operativos y homologados sus ambientes previos con los de producción.

Determinar la condición operativa actual de los ambientes previos y productivos, lo mismo que validar si el código fuente de la aplicación que reside en producción está homologado con el de los ambientes previos, ayuda a identificar restricciones técnicas importantes, si es que se tienen hallazgos negativos.

14. Determinar los tipos y el número de entornos de desarrollo (IDE) por cada aplicación.

Estas consideraciones ayudan a establecer requisitos adicionales al respecto.

4.1.4.2 Prácticas sugeridas para el proceso caracterización de externalizables

Para los ítems a externalizar, identificados en el proceso previo, se debe realizar un perfilado con sus características técnicas. Las siguientes prácticas pueden ayudar a este fin.

1. Perfilar las características básicas de una aplicación.

Esto deberá incluir entre otras características áreas usuarias que la usan, funciones de negocio que soporta, descripción de la tecnología sobre la que esta desarrollada, entornos y ubicaciones, aplicaciones con las que se comunica, infraestructura de TI y de software base que la soporta, licencias requeridas, IDE de desarrollo, criticidad para el negocio.

2. Perfilar las características técnicas de los repositorios de código fuente y su gestor de versiones de cada aplicación.
3. Perfilar los elementos que participan en el proceso DevOps o DevSecOps, si es que estos van a ser delegados también.
4. Por cada ítem a externalizar se deberá identificar los roles técnicos involucrados en su mantenimiento y soporte. Esto ayudará a determinar el capacity de recursos necesarios que el proveedor de SWF deberá garantizar.
5. Actualice las restricciones técnicas identificadas en el proceso previo por cada ítem a delegar, si durante el perfilado se tiene nueva información sobre las mismas.

4.1.4.3 Prácticas sugeridas para el proceso desarrollar los requisitos técnicos de la externalización (RTE)

Las siguientes prácticas contribuyen a obtener los RTE:

1. Por cada ítem externalizable (aplicación, interfaz, u otro elemento) desarrollar un documento de definición de alcance (DDA) en el que desarrollará los requisitos técnicos exigidos por el adquiriente al proveedor de SWF con respecto a ese ítem. Estos requisitos abarcan responsabilidades, procedimientos de operación, acuerdos de niveles de servicio, entre otros relacionados con el ítem que se externaliza. Se debe tener en cuenta los perfiles técnicos y restricciones obtenidas del proceso previo.
2. Incluya dentro de los requisitos técnicos las responsabilidades respecto al licenciamiento de componentes de terceros, como el de las API, o las condiciones de servicios de terceros consumidos por las aplicaciones
3. En cada DDA se recomienda crear un apartado con los roles y sus aptitudes necesarias para mantener cada ítem.
4. En el DDA se deben incluir requisitos para gestionar la parte técnica, por ejemplo, obligatoriedad a asistir a reuniones de seguimiento, elaboración de informes sobre el servicio y su frecuencia, entre otros.
5. Aunque proponemos esto en el proceso siguiente, se deben definir desde ya los requisitos de verificación, es decir el mecanismo que usará el adquiriente para constatar el cumplimiento de los otros requisitos.

4.1.4.4 Prácticas sugeridas para el proceso de validación y verificación de los DDA

Estas prácticas ayudan a validar si estamos definiendo los RTE correctos y a establecer requisitos de verificación que permitan comprobar los RTE:

1. El equipo del proyecto debe explicar los DDA (salida del proceso previo, que contiene los RTE de cada ítem a externalizar) a cada líder usuario (LU) según su área de interés, dado que ellos no han participado en los procesos previos.
2. Todos deben validar que cada requerimiento este dentro del ámbito del caso de negocio.

3. Todos deben revisar los enunciados de la especificación de cada requerimiento para validar de que los mismos cumplen con las necesidades requeridas, que son inteligibles, específicos, medibles, alcanzables, verificables.
4. Todos deben revisar que no haya requerimientos contradictorios, es decir que uno contradiga parcial o totalmente a otros, no sólo a nivel de lo que hay que realizar sino de plazos o responsables.
5. Todos deben revisar que no haya requerimientos redundantes.
6. Para los ítems 4 y 5 previos se recomienda realizar un matriz de requisitos por requisitos, para comparar unos con otros y ver que abarcan puntos diferentes.
7. Definir criterios de verificación de cada uno de los requisitos definidos, para poder comprobar el cumplimiento objetivo de los mismos. P.ej. como haríamos para verificar que el proveedor de SWF tomo control efectivo del servidor de aplicaciones, o qué controles y/o informes deberán emitirse para verificar el cumplimiento de ANS.
8. Se deben establecer que roles de la SWF, del proveedor, serán los responsables del cumplimiento de cada requisito, así como de aquellas actividades por parte del adquiriente.
9. Todos los involucrados deben cualificar cualitativamente cada DDA por complejidad y prioridad; deberán hacer lo mismo con cada requisito dentro de cada DDA. Se puede incluir otro tipo de cualificación en función al criterio del adquiriente.

4.2 Macroproceso dos (macropro-2): Definir un marco metodológico de DMS para la externalización.

Busca lograr el objetivo específico 02 de nuestro trabajo:

OE02: Definir procesos del framework que permitan establecer criterios apropiados para seleccionar un marco metodológico de desarrollo de software en un contexto de externalización vía un modelo de SWF.

4.2.1 Diagrama de contexto y descripción del macropro-2.

La figura 19 y la tabla 10 siguientes se corresponden con el diagrama de contexto y la descripción.

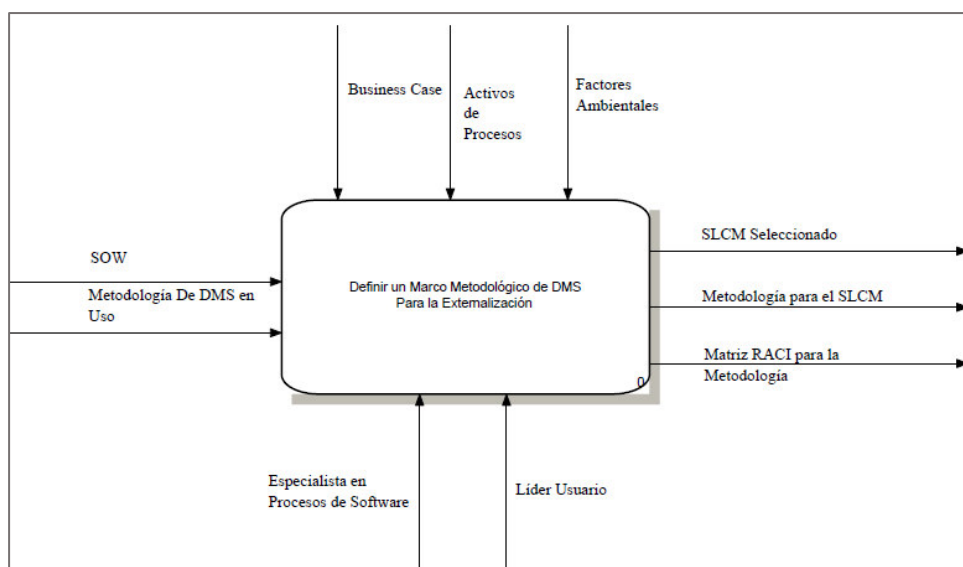


Figura 19. Diagrama de contexto del macropro-2

Fuente. Elaborador por el autor

Tabla 10. Descripción de los elementos del macropro-2 de SWFrame

Nombre del macroproceso	
Definir un Marco Metodológico de DMS Para la Externalización.	
Propósito	
Ayudar a seleccionar el SLCM más adecuado a usarse en el ámbito de la SWF.	
Punto de vista	
Equipo del proyecto del adquiriente conformado por los roles que participan en cada proceso.	
Entradas	
SWO	Ya descrito.
Metodología de DMS en Uso.	Es la metodología de DMS usada actualmente por la organización o lo que se le aproxime, como un proceso; aunque es un activo de proceso, lo consideramos como input porque es un insumo principal para este proceso. No debemos confundirla con el proceso de gestión del proyecto.
Elementos de Control	
Business Case	Ya descrito.
Activos de procesos	Agrupar normas, procedimientos, metodologías de la compañía que se deben tener en cuenta al ejecutar este proceso.
Factores ambientales	Se refiere a los factores del entorno como la cultura, sistemas existentes de administración, estructura organizacional y otros de la organización adquiriente.
Recursos	
Especialista en procesos de software	Rol conocedor de SLCM, procesos de software y del actual proceso de desarrollo del adquiriente. Este rol debe ser parte del equipo del proyecto de externalización.
Líder usuario	Son los representantes de las áreas usuarias propietarias y/o usuarias de las aplicaciones a externalizar.
Salidas	
SLCM seleccionado	Modelo de ciclo de vida seleccionado.
Metodología para el SLCM seleccionado	Es la metodología que implementa el SLCM, definiendo flujos de actividades, entregables, roles, guías entre otros.
Matriz RACI para la metodología.	Cuadro de responsabilidades que deberá indicar quienes asumirán, del lado del proveedor y del adquiriente, los roles de la metodología y la responsabilidad de los entregables de la misma.

Fuente. Elaborado por el autor.

4.2.2 Diagrama de descomposición y procesos de macropro-2.

Para este macroproceso su diagrama de descomposición se indica a continuación en la figura 20, en ella apreciamos que está conformado por tres procesos que describimos en los siguientes ítems.

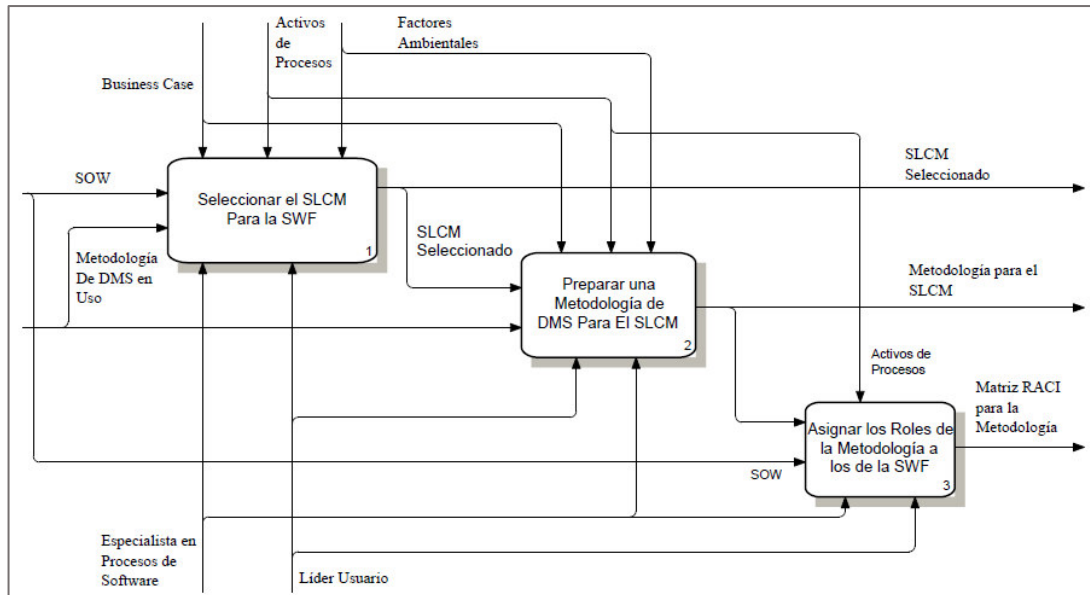


Figura 20. Diagrama de descomposición del macropro-2

Fuente. Elaborado por el autor

4.2.3 Procesos del macroproceso 02

A continuación se describe cada proceso del macroproceso 02 y seguidamente las prácticas recomendadas por cada uno de ellos

4.2.3.1 Seleccionar el SLCM para la SWF.

Tabla 11. Descripción del proceso seleccionar el SLCM para la SWF

Nombre del proceso	
Seleccionar el SLCM para la SWF.	
Propósito	
Seleccionar el SLCM más adecuado para usarse en un contexto de externalización y conveniente para el adquiriente.	
Entradas	
SOW	Ya descrito.
Metodología de DMS en uso	Ya descrito.
Elementos de Control	
Business Case	Ya definidos.
Factores Ambientales	
Activos de Procesos	
Recursos	
Equipo del proyecto	Ya descritos.
Lidere usuario	
Salidas	
SLCM seleccionado	SLCM elegido mejor opción a usarse en el contexto de la externalización. Pudiera elegirse más de un modelo, si el adquiriente tiene la necesidad de usar más de un enfoque, sin embargo, esto incrementa la complejidad del control y de otros procesos operativos de la SWF, como el de gestión de necesidades.

Fuente. Elaborado por el autor.

4.2.3.2 Preparar una metodología de DMS para el SLCM.

Tabla 12. Descripción de preparar una metodología de DMS para el SLCM

Nombre del proceso	
Preparar una metodología de DMS para el SLCM.	
Propósito	
Desarrollar una metodología que implemente el SLCM elegido. El adquiriente deberá incluir actividades e hitos de control que le ayuden a validar y verificar los productos de trabajo de la metodología, entre ellos los entregables.	
Entradas	
SLCM seleccionado	Ya descritos.
Metodología de DMS en uso	
Elementos de Control	
Business Case	Ya descritos.
Factores Ambientales	
Activos de Procesos	
Recursos	
Equipo del proyecto	Ya descritos.
Líder usuario	
Salidas	
Metodología Para el SLCM	La metodología que define fases del DMS, actividades con sus entradas y salidas, productos de trabajo, flujos de actividades, roles, guías, plantillas, entre otros

Fuente. Elaborado por el autor.

4.2.3.3 Asignar los Roles de la Metodología a los de la SWF.

Tabla 13. Descripción de asignar los Roles de la Metodología a los de la SWF

Nombre del proceso	
Asignar los Roles de la Metodología a los de la SWF	
Propósito	
El objetivo es determinar y precisar que roles del adquiriente y del proveedor son responsables de que actividades y productos de trabajo de la metodología definida.	
Entradas	
SOW	Ya descrito.
Metodología para el SLCM.	
Elementos de Control	
Activos de procesos	Ya descrito.
Recursos	
Equipo del proyecto	Ya descritos.
Lidere usuario	
Salidas	
Matriz RACI para la metodología	La metodología definirá roles (p.ej. Modelador, Tester, Developer, etc.) que realizan la tarea, los que en una matriz RACI calificarían como los responsables, pero habrá que precisar si es un rol del proveedor o del adquiriente. También hay que definir el aprobador (Accountable), Consultado (Consulted) y el Informado (Informed). La matriz RACI debe ser parte del AE y por ende del contrato.

Fuente. Elaborado por el autor.

4.2.4 Prácticas sugeridas para cada proceso de macropro02

4.2.4.1 Definir SLCM a usar en la externalización

Como indicamos en el apartado 2.3.8 del marco teórico, los SLCM fluctúan entre los estrictamente predictivos hasta los adaptativos con variantes entre ambos extremos. Se tiene que recordar que los SLCM son formas de organizar los procesos de software (PS), y requieren de una metodología que los implemente. En un contexto de externalización se debe elegir uno, entre los existentes, de acuerdo con las necesidades y preparación del adquiriente, buscando personalizarlo para adecuarlo de la mejor manera al entorno de externalización, a la SWF. Entre las buenas prácticas a considerar tenemos:

1. Revise el SLCM que usa a la fecha en función a su metodología actual o a su proceso de DMS usado y verifique si este tiende a lo predictivo o adaptativo, luego pregúntese si desea mantener el enfoque en su modelo de externalización; aún si decidiese mantenerlo, su metodología va a cambiar porque muchas de esas actividades ahora serán realizadas por la SWF y se requerirán otras, como aquellas que permitirán el monitoreo y control por parte del adquiriente.
2. Considere la cultura organizacional del adquiriente, generalmente las organizaciones muy reguladas por entes internos y externos se orientarán más a enfoques predictivos, sin que signifique que se puedan balancear con enfoques adaptativos. Revise los modelos comentados en el marco teórico.
3. Evaluar si la frecuencia de las solicitudes de cambios en proyectos pasados es alta, lo que podría orientar la selección del modelo hacia enfoques adaptativos, como alguno de los descritos en el marco teórico.
4. Considere el nivel de experiencia de los recursos que exigirá al proveedor de SWF, si este nivel es alto, probablemente podría tener éxito un enfoque adaptativo, como alguno de los descritos en el marco teórico.
5. Si el adquiriente puede empoderar un representante suyo a los proyectos, para que pueda definir y aprobar alcances, dar conformidad a las implementaciones, autorizar cambios, absolver consultas con celeridad entonces un enfoque adaptativo podría ser la elección, pero esto dependerá de los procesos normativos de la organización del adquiriente.
6. Si la criticidad de los sistemas a desarrollar o mantener es alta, es decir el coste por fallas puede ser muy gravoso, con impactos importantes, entonces un enfoque predictivo que planifique bien cada fase y la documente podría convenir más.
7. Si el adquiriente planea montar sistemas de software por etapas cortas un enfoque adaptativo sería lo más conveniente.
8. Recuerde que está en un contexto de externalización, y lo que su SLCM plantee debe ser viable técnica y financieramente, así por ejemplo los enfoques adaptativos, intensos en prácticas ágiles, deberían poder realizarse en menos de seis meses, para ello debe poder definir un producto viable de realizarse en ese periodo, debe poder sustraerse el desarrollo de documentación innecesaria y de procesos engorrosos; sin embargo si su entorno ambiental es regulado por normas que impiden esta flexibilidad, entonces su organización no calificaría para un modelo puramente adaptativo.

9. No sea purista, en el sentido de elegir entre los extremos, puede personalizar su SCLM, para ello nuevamente recomendamos revisar lo expuesto al respecto en nuestro marco teórico y ampliar esta revisión a otras fuentes.

4.2.4.2 Prácticas sugeridas para el proceso preparar una metodología de DMS para el SLCM.

El SLCM elegido necesita ser aterrizado en una metodología, en el sentido estricto de lo que esto significa (ver la definición de metodología que damos en el apartado 2.3.6), para esto recomendamos las siguientes prácticas.

1. Este proceso deberá ser definido por uno o más especialistas en procesos de software, conocedores del ciclo de desarrollo actual del adquiriente, también de sus activos de procesos relacionados y de la cultura organizacional.
2. Define primero las fases de la metodología, pero no recomendamos que estas sean el equivalente al orden natural de los procesos de software (análisis, diseño, etc.), porque si usa esta secuencia estará asumiendo un modelo estrictamente predictivo.
3. Por cada fase podría definir los PS que en ella usará, inclusive el grado de intensidad de c/u de ellos. Luego por cada PS defina las actividades correspondientes que ese proceso realizará, p.ej. para el PS de diseño, podría definir que debe abarcar actividades como diseño de clases, diseño de GUI, entre otras. Luego por cada actividad defina las tareas que la implementan e inclusive los pasos de cada tarea.
4. Determine los roles responsables de cada tarea y de cada actividad, estos pueden ser testers, managers, developers y otros. Estos roles son los que realizan la tarea.
5. Identifique las entradas y salidas de cada actividad. Muchas salidas serán productos de trabajo y algunos de estos entregables que requerirán la verificación por parte del adquiriente en cada proyecto.
6. Habiendo definido sus PS, debe definir cuáles desea incluir por cada fase que definió, lo que dependerá de su SCLM elegido, así podrá decidir ser más lineal (acercándose a un waterfall) o más iterativo e incremental (acercándose a un enfoque puramente adaptativo). Asegúrese de que no haya una incongruencia entre el modelo y su implementación metodológica.
7. Para los pasos previos puede apoyarse en la propuesta de UMA, descrita en el ítem 2.3.11 del marco teórico, para tal efecto tenga en cuenta lo siguiente:

- a. Cree un contenido central del método basado en el concepto de tareas, roles, productos de trabajo y guías.
 - b. Las tareas deben dividirse en pasos, estos deben ser de reflexión, de ejecución y de revisión; los primeros para examinar los productos de trabajo de entrada y para formular un resultado, los segundos para crear o actualizar los productos de trabajo, finalmente los terceros donde el rol que ejecuta la tarea verifica los resultados.
 - c. Agrupe las tareas en actividades y estas a su vez en procesos que formaran parte de una determinada fase de lo que UMA llama proceso de entrega (metodología). Se recomienda crear un proceso de entrega principal, que abarque todo lo que se desearía realizar en un proyecto promedio.
 - d. Reutilice y personalice este proceso principal para definir variantes del mismo según el tipo y tamaño de proyecto.
 - e. El adquiriente podría de antemano definir estos procesos de entrega y tener por ejemplo un ciclo metodológico para proyectos pequeños, medianos y grandes.
8. El objetivo no es definir una metodología única e inflexible para todos los proyectos de DMS que la SWF ejecutará, sino una que se pueda personalizar según la índole del proyecto y en concordancia con las partes durante el inicio de cada proyecto e inclusive antes, pero a partir de los patrones de entrega previamente definidos.
 9. Explique la metodología tanto a usuarios de negocio como al equipo de TI, y sométala a revisión por parte de ellos, para recoger sus observaciones.
 10. Se recomienda con énfasis crear la metodología usando alguna herramienta para creación de procesos como Rational Method Composer o Eclipse Process Framework Composer, esto permitirá realizar las personalizaciones a la misma de manera sistemática y publicarla vía web al interior de la SWF y en el entorno del adquiriente, para que sea accesible para conocimiento, revisión y aplicación. Además, estas herramientas permiten crear con relativa facilidad variantes de un proceso de entrega definido.
 11. Haga la metodología definida parte del AE.

4.2.4.3 Prácticas sugeridas para el proceso Asignar los Roles de la Metodología a los de la SWF.

Las actividades de la metodología tienen roles participantes, por lo que es necesario definir en el AE quienes asumirán estos, tanto por el lado del adquiriente como por el lado del proveedor de la SWF.

1. Se recomienda elaborar una matriz RACI por cada PS, actividades del PS versus roles de la SWF, asegure la correctitud del rol responsable. Para facilitar esta tarea, se recomienda previamente hacer una correlación entre los roles de la metodología definida y los recursos y roles existentes en la SWF.
2. Las actividades administrativas relacionadas, que no hubieran sido consideradas como parte de la metodología deberían tener su propia matriz RACI, como por ejemplo aquellas relacionadas con gestión de necesidades o de la demanda, que es donde se autorizan los requisitos que luego se convierten en proyectos.
3. El conjunto de matrices RACI deben ser parte del AE.

5 Validación y Resultados

5.1 Validez y fiabilidad de los datos recopilados en la encuesta

Dado que nuestros constructos miden percepciones y están obviamente no son observables ni medibles directamente, hemos utilizado una encuesta como instrumento de medida, pero antes de sumar los puntajes obtenidos de cada ítems de esta, fue necesario demostrar que hay consistencia interna entre estos, y por ende es correcto totalizarlos y así obtener una medida de calidad de los constructos. Para ello se tuvo que demostrar que las puntuaciones obtenidas en la muestra trabajada son fiables y validas. La primera propiedad se refiere al grado o posibilidad de reproducir los resultados aplicando el mismo instrumento varias veces al mismo sujeto u objeto; la segunda es el grado en que el instrumento mide aquello que es su objetivo medir (Frías-Navarro, 2022) (Hernández Sampieri y otros, 2010).

5.1.1 Validez

Para evaluar la validez se determinó la validez convergente y divergente entre ítems (Los PEOU y PU de la encuesta), donde el valor de la validez convergente promedio de un ítem debe ser mayor que su validez divergente promedio, según (Campbell y Fiske) citado por (Abrahamo Gonzales y otros, 2011). En la tabla 14 se muestran los resultados del análisis de correlación inter-ítems, donde se comprueba la validez de los datos recolectados dado que todos los promedios cumplen la regla de Campbell y Fiske, excepto los ítems PEOU3 y PU1, cuyos promedios de validez divergente fueron mayores que sus correspondientes de validez convergente por lo que estos ítems fueron excluidos del análisis. La validez convergente promedio fue de 0,484 para PEOU y 0,499 para PU; la divergente fue de 0.348 para PEOU y 0.317 para PU.

5.1.2 Fiabilidad

Para determinar la fiabilidad del instrumento, se usó el alfa de Cronbach, los resultados de este análisis de fiabilidad, luego de remover la pregunta PEOU3 y PU1, fueron los siguientes para el ítem PEOU =0.776 y para PU=0.700, y el total para el cuestionario fue de 0.782, los que se muestran en la tabla 15. Según George, D., & Mallery, P. (2003) citado por (Frías-Navarro, 2022, pág. 11) el mínimo valor de fiabilidad aceptable es 0.7, por lo que se concluye que los elementos de la encuesta son fiables.

Tabla 14. Correlación inter-ítems a los datos recopilados

	Perceived Usefulness (PU)				General							CV	DV	VALID?
	PEOU1	PEOU2	PEOU3	PEOU4	PU1	PU2	PU3	PU4	PU5	PU6	PU7			
PEOU1	1.000	0.500	0.158	0.791	0.866	0.158	0.493	0.000	0.000	0.158	0.000	0.483	0.239	YES
PEOU2	0.500	1.000	0.632	0.316	0.577	0.632	0.493	0.500	0.500	0.158	0.000	0.483	0.409	YES
PEOU3	0.158	0.632	1.000	0.350	0.274	0.550	0.639	0.316	0.632	0.100	0.316	0.380	0.404	NO
PEOU4	0.791	0.316	0.350	1.000	0.822	0.350	0.623	0.158	0.316	0.350	0.158	0.486	0.397	YES
PU1	0.866	0.577	0.274	0.822	1.000	0.548	0.768	0.000	0.289	0.548	0.289	0.407	0.635	NO
PU2	0.158	0.632	0.550	0.350	0.548	1.000	0.639	0.316	0.632	0.550	0.316	0.500	0.423	YES
PU3	0.493	0.493	0.639	0.623	0.768	0.639	1.000	-0.197	0.493	0.639	0.690	0.571	0.562	YES
PU4	0.000	0.500	0.316	0.158	0.000	0.316	-0.197	1.000	0.500	-0.158	-0.500	0.279	0.244	YES
PU5	0.000	0.500	0.632	0.316	0.289	0.632	0.493	0.500	1.000	0.632	0.500	0.508	0.362	YES
PU6	0.158	0.158	0.100	0.350	0.548	0.550	0.639	-0.158	0.632	1.000	0.791	0.553	0.192	YES
PU7	0.000	0.000	0.316	0.158	0.289	0.316	0.690	-0.500	0.500	0.791	1.000	0.584	0.119	YES

Fuente: Propia.

Nota: Datos generados con SPSS.

Tabla 15. Análisis de fiabilidad para los ítems de la encuesta

	Alfa de Chronbach	N de elementos
PEOU	0.776	3
PU	0.700	6
PEOU+PU	0.782	9

Fuente: Propia.

Nota. Datos generados usando SPSS.

5.2 Pruebas de hipótesis

En este apartado probaremos las hipótesis de MEM H1,H2 y H6 que son de interés para nuestro trabajo de investigación y que probarían la percepción de utilidad y de facilidad de uso de nuestro framework.

5.2.1 Evaluando H1 y H2 analizando los resultados para las variables de percepción y de desempeño.

Luego de calcular las variables de desempeño según lo explicado con anterioridad en este documento, se obtiene para A_EFE (Actual Effectiness) un valor de 0.810, lo que quiere decir que la efectividad para responder las preguntas de la tarea de evaluación fue alta, superior al 80%. Para A_EFI (Actual Efficiency) se obtuvo el valor de 0.990, que implica un muy buen uso del tiempo en responder correctamente las preguntas. En la tabla 16 se indican los datos paramétricos de estas mediciones. Ambos indicadores son muy buenos y apoyan las hipótesis alternas de H1 y H2.

Tabla 16. Datos paramétricos de las variables basadas en desempeño

Variable	N	Min.	Max.	Mean	Std. Dev.
A_EFE	9	0.571	1.000	0.809	0.123
A_EFI	9	0.500	1.500	0.990	0.365

Fuente. Propia.

Para probar si el framework se percibe como fácil de usar (PEOU) y útil (PU) se midieron las variables basadas en percepciones con los datos arrojados por la encuesta, los resultados indican que los sujetos percibían el framework como fácil de usar ($\neg H1_0$) y útil ($\neg H2_0$), dado que el valor promedio de estas variables es mayor que el valor tres, promedio de la población, ver resultados en las tablas 17 a la 19.

Tabla 17. Estadísticos descriptivos de PEOU y PU

Variable	N	Min.	Max.	Mean	Std. Dev.
PEOU	36	4	5	4.42	0.500
PU	63	1	5	4.38	0.851

Fuente. Propia.

Tabla 18. Estadísticos descriptivos por cada ítem de PEOU

	N	Min.	Max.	Mean	Std. Dev.
PEOU1	9	4	5	4.33	0.500
PEOU2	9	4	5	4.33	0.500
PEOU3	9	4	5	4.56	0.527
PEOU4	9	4	5	4.44	0.527

Fuente. Propia.

Tabla 19 Estadísticos descriptivos por cada ítem de PU

	N	Min.	Max.	Mean	Std. Dev.
PU1	9	3	5	4.00	0.866
PU2	9	4	5	4.56	0.527
PU3	9	1	5	3.89	1.691
PU4	9	4	5	4.67	0.500
PU5	9	4	5	4.33	0.500
PU6	9	4	5	4.56	0.527
PU7	9	4	5	4.67	0.500

Fuente. Propia.

Toca probar el supuesto de que hay una diferencia significativa entre las medias de PEOU y PU con respecto a tres, valor promedio de la población; para tal efecto, sobre los datos correspondientes a PEOU y PU se realizó una prueba de normalidad, cuyos resultados se muestran en la tabla 20, así con un alfa de 0.05 se obtuvo valores de significancia superiores a alfa, demostrando que los datos tienen una distribución normal. En esta tabla TOTAL_PEOU y TOTAL PU representan las sumatorias de los ítems PEOU y PU respectivamente del instrumento de medición y se usó la prueba shapiro-wilk, pues nuestra muestra es menor a veinte individuos. Verificada la normalidad aplicamos la prueba t-student, cuyos resultados demuestran lo supuesto. Estos se muestran en la tabla 21.

Tabla 20. Prueba de normalidad shapiro-Wilk para PEOU y PU

Variable	Estadístico	gl	Sig.
Total_PEOU	0.858	9	0.092
Total_PU	0.878	9	0.148

Fuente. Propia.

Tabla 21. Estadísticas descriptivas y prueba t de 1 cola con valor de prueba tres

Variable	Min.	Max.	Mean	Std. Dev.	Std. Error mean	t	gl	Sig. (bilateral)
PEOU	4.25	4.59	4.42	0.500	0.083	17	35	<0.001
PU	4.17	4.60	4.38	0.851	0.107	12.886	62	<0.001

Fuente. Propia.

5.2.2 Evaluando H6, la relación causales entre PEOU y PU.

La hipótesis H6 se derivan del Modelo de Aceptación de Tecnología (TAM), predecesor de MEM, en el que se encontró una influencia directa de la PEOU sobre la PU. Esto se comprobó determinando el grado de correlación que hay entre estos dos constructos. Dado que ya hemos corroborado que ambos set de datos son normales, hemos calculado el índice R de correlación de Pearson, el resultado se indica en la tabla 22, que con un valor de 0.747 denota una correlación significativa, que la percepción de utilidad del framework está determinada por la de facilidad de uso. De esta tabla ya hemos explicado TOTAL_PEOU y TOTAL_PU con anterioridad.

Tabla 22. Correlación de Pearson entre PU y PEOU

	TOTAL_PEOU
TOTAL_PU	Correlación de Pearson 0,747*
	Sig. (bilateral) 0,021
	N 9

Fuente. Propia

5.3 Presentación de resultados

Logramos verificar que el framework se percibe como útil y fácil de usar al verificarse las hipótesis alternas de H1 y H2, esto se respalda por las medias obtenidas para las variables de percepción y por los grados de eficiencia y eficacia obtenidos de la data de las variables de desempeño. Se demostró también que existe una relación entre la percepción de la facilidad de uso sobre la percepción de la utilidad, tal como la validación de H6 lo muestra. Por supuesto que cuando un framework como el nuestro se lleve a uso en un contexto de un proyecto real de externalización, otros elementos o causas surgirán que impacten sobre las percepciones que hemos buscado medir mediante MEM vía su propuesta, p. ej., entre otros la complejidad de la arquitectura empresarial, las urgencias del negocio, la normativa vigente, los factores ambientales, la experiencia del equipo de externalización. Sin embargo con MEM hemos logrado evaluar metódicamente estas variable de percepción, que por ser del tipo latentes no se pueden medir directamente.

6 Conclusiones y trabajos futuros

6.1 Conclusiones

En esta tesis se propuso SWFrame que presenta procesos y prácticas que bien usados contribuirán a una correcta definición del AE para adoptar un modelo de SWF. Este fue validado utilizando los mecanismos sugeridos por MEM, para evaluar los grados de utilidad y de facilidad de uso percibidos mediante los instrumentos de medición que propone y hemos explicado, lo que permitió corroborar las hipótesis planteadas, y así lograr el objetivo principal de este trabajo de proponer un framework como un instrumento útil y de fácil uso para ayudar acometer con éxito el reto de definir un esmerado AE, cuestión vital en estos siempre tan complejos proyectos de externalizaciones de DMS y de ITO en general, tan intensivos y recurrentes en el ámbito empresarial.

En relación con los objetivos específicos, estos se lograron porque el framework propuesto incluye procesos para los fines planteados en estos, todos representados en la notación IDEF0 la que nos permitió representar el aspecto funcional y seguir un enfoque top-down para facilitar el entendimiento de cada proceso, para los cuales además hemos incluido prácticas recomendadas para su implementación.

Una conclusión aparte merece el estado del arte, por el cual se determinó que a la fecha existen pocas propuestas en la literatura que ayuden a desarrollar el AE para el problema particular que hemos planteado, lo encontrado son modelos prescriptivos y genéricos, agnósticos a cualquier tipo de ITO, lo que respalda nuestra propuesta.

6.2 Trabajos futuros

En este trabajo hemos sido específicos en proponer procesos y prácticas para la externalización del tipo DMS mediante SWFrame, el cual puede ampliarse para incorporar otros aspectos de la externalización de este tipo de función organizacional, como la de precisar cómo sería el tratamiento de la gestión de la demanda o necesidades, como se manejarían los mantenimientos adaptativos, o los procesos de desarrollo seguro en entornos de SWF, entre otros. Además el mismo enfoque seguido en este trabajo se podría usar para proponer otros framework que ayuden a definir el

AE correspondiente para otro tipo de externalizaciones, como las de administración de la infraestructura de TI o la del monitoreo y gestión de la seguridad informática por los servicios SOC de terceros. En general las ITO en la actualidad abarcan muchas áreas funcionales de una compañía, todas deben licitarse y para esto se requerirá siempre un adecuado AE, crucial para cualquier tipo de outsourcing.

7 Bibliografía

- Abrahamo Gonzales, S., Insfrán Pelozo, C., Carsí Cubel, J., & Genero Bocco, M. (2011). Evaluating Requirements Methods based on User Perceptions: A Family of Experiments. *Information Sciences*, 181(16), 3356-3378. <https://doi.org/oi:10.1016/j.ins.2011.04.005>
- Ahmad, F., Ali, R., Haini, S., & Maarop, N. (2017). Maturity models of IT outsourcing: A systematic literature review. *International Journal of Advanced and Applied Sciences*, 4(12), 52-56. <https://doi.org/10.21833/ijaas.2017.012.011>
- Azabache, A. (2022). Cómo definir el alcance de una externalización de desarrollo de aplicaciones mediante un modelo de software factory : Una revisión. *Revista Peruana de Computación y Sistemas*, 4(2), 12. <https://doi.org/https://doi.org/10.15381/rpcs.v4i2.24856>
- Barney, S., Mohankumar, V., Chatzipetrou, P., Aurum, A., Wohlin, C., & Angelis, L. (2014). Software quality across borders: Three case studies on company internal alignment. *Information and Software Technology*, 56(1), 20-38. Retrieved 2022, from <https://www.webofscience.com/>
- Bente, S., Bombosch, U., & Langade, S. (2012). *Collaborative Enterprise Architecture*. Waltham, Massachusetts, USA: Elsevier.
- Birk, A., Surmann, D., & Althoff, K.-D. (1999). Applications of Knowledge Acquisition in Experimental Software Engineering. *Lecture Notes in Computer Science book series*, 1621, 67-84. https://doi.org/10.1007/3-540-48775-1_5
- Bjørnson, F., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, 50, 1055-1068. <https://doi.org/10.1016/j.infsof.2008.03.006>
- Brownsword, L., Albert, C., Carney, D., & Place, P. (Octubre de 2014). *A Method for Aligning Acquisition Strategies and Software Architectures*. Retrieved Noviembre de 2022, from <https://resources.sei.cmu.edu/library/>

- Cho, D. (2020). A Contingent Approach to Facilitating Conflict Resolution in Software Development Outsourcing Projects. *JOURNAL OF ORGANIZATIONAL AND END USER COMPUTING*, 32(2), 20-41. <https://www.igi-global.com/gateway/article/245997>
- CMMI for Acquisition, Version 1.3. (2010). *Software Engineering Process Management Program*.
- Comer, E. (1991). Alternative Software Life Cycle Models. <https://silo.tips/download/alternative-software-life-cycle-models>
- Conrad, E. (2011). *Eleventh Hour CISSP*. Retrieved 2022, from ScienceDirect: <https://www.sciencedirect.com/science/article/pii/B9781597495660000084>
- Cruz-Benito, J. (7 de Noviembre de 2016). *zenodo*. <https://doi.org/10.5281/zenodo.165773>
- Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340. <https://doi.org/https://doi.org/10.2307/249008>
- Dias, J. W., & Oliveira Jr, E. (2016). Modeling Variability in Software Process with EPF Composer and SMartySPeM: An Empirical Qualitative Study. *18th International Conference on Enterprise Information Systems*, 1, 283-293. http://eprints.staffs.ac.uk/2770/1/ICEIS_2016_Volume_1.pdf#page=304
- Frías-Navarro, D. (2022). Apuntes de estimación de la fiabilidad de consistencia interna de los ítems de un instrumento de medida. Valencia, España. Retrieved 2023, from <https://www.uv.es/friasnav/AlfaCronbach.pdf>
- Hefley, W. E., & Loesche, E. A. (2006). *The eSCM-CL v1.1: Model Overview Part 1*. Carnegie Mellon University, Information Technology Services Qualifications Center (ITSqc), Pittsburgh, PA. Retrieved Octubre de 2022, from <http://www.qostic.org/Qostic/wp-content/uploads/Qostic6/AHQ-19-08-eSCM-CL-Part1-V1-1.pdf>
- Hefley, W., & Loesche, E. (2006). *The eSCM-CL v1.1: Practice Details Part 2*. Carnegie Mellon University, Information Technology Services Qualification Center (ITSqc), Pittsburgh, PA. Retrieved 2022, from

<http://www.qostic.org/Qostic/wp-content/uploads/Qostic6/AHQ-19-09-eSCM-CL-Part2-V1-1.pdf>

- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2010). *Metodología de la investigación* (Quinta ed.). McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V. Retrieved 2022.
- Hyder, E., Heston, K., & Paulk, M. (2006). *The eSCM-SP v2.01: Model Overview. The eSourcing Capability Model for Service Providers (eSCM-SP) v2.01*. Pittsburgh, , Pennsylvania, USA. Retrieved 2022, from https://www.researchgate.net/publication/220689790_eSourcing_Capability_Model_for_Service_Providers_eSCM-SP
- IBM Corp. (2006). *IBM(R) Rational Unified Process(R)*. Retrieved enero de 2023, from SWI Group Wise development: https://swi.cs.vsb.cz/RUPSmall/#core.base_concepts/customcategories/method_architecture_fundamentals_5F73467B.html
- IEEE. (2004). Software Engineering Body Of Knowledge. IEEE Computer Society Products and Services.
- Ikram, A., Riaz, H., & Khan, A. S. (2018). Eliciting Theory of Software Maintenance Outsourcing Process: A Systematic Literature Review. *IJCSNS International Journal of Computer Science and Network Security*., VOL.18(4), 12. Retrieved 2022, from https://www.academia.edu/download/65739908/SMOPProcessModel_2_.pdf
- International Organization for Standardization [ISO]. (2017). *ISO/IEC/IEEE 12207:2017(en) Systems and software engineering — Software life cycle processes*. <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:12207:ed-1:v1:en>
- International Organization for Standardization. (2014). *Online Browsing Platform (OBP)*. Retrieved 10 de 2022, from ISO/IEC 24744:2014(en) Software engineering — Metamodel for development methodologies: <https://www.iso.org/obp/ui#iso:std:iso-iec:24744:ed-2:v1:en:term:3.2>
- IQBAL, J., AHMAD, R., KHAN, M., NIZAM, M., & AKHUNZADA, A. (2022). Model to Cope With Requirements Engineering Issues for Software

- Development Outsourcing. *IEEE ACCSSES*, 63199-63229. Retrieved 2022, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794738>
- Ketler, K., & Walstrom, J. (1993). The Outsourcing Decision. *International Journal of Information Management*, 13, 449-459. [https://doi.org/10.1016/0268-4012\(93\)90061-8](https://doi.org/10.1016/0268-4012(93)90061-8)
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. 57. Retrieved 08 de 2022, from https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering
- Kronawitter, K., Wentzel, C., & Papadaki, M. (2013). IT Application Outsourcing in Europe: Long-term Outcomes, Success Factors and Implications for ITO Maturity. *46th Hawaii International Conference on System Sciences*, 4456-4465. <https://doi.org/10.1109/HICSS.2013.369>
- Martínez Sánchez, Á., Vela Jiménez, M., de Luis Carnicer, P., & Pérez Pérez, M. (2007). Externalización, flexibilidad del trabajo y resultados empresariales. *Cuadernos de Economía y Dirección de la Empresa*(33), 14. <https://doi.org/DOI: 10.29019/enfoque.v11n2.621>
- Montequín, V., Pérez, C., Fernández, F., & Balsera, J. (2013). Scorecard and KPIs for monitoring software factories effectiveness in the financial sector. *International Journal of Information Systems and Project Management*, 1(3), 15. <https://doi.org/0.12821/ijispm010302>
- Moodys, D. L., Sindre, G., Brasethvik, T., & Sølvsberg, A. (2002). Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework. *Conceptual Modeling — ER 2002*, 380–396. https://doi.org/10.1007/3-540-45816-6_36
- Naciri, S. (2014). Third-Party Application Maintenance Management. *International Journal of Computer Applications*, 103(6). Retrieved 11 de 2022, from https://www.researchgate.net/profile/Samia-Naciri/publication/273345678_Third-Party_Application_Maintenance_Management/links/5b841f52a6fdcc5f8b6b497a/Third-Party-Application-Maintenance-Management.pdf

- National Institute of Standards and Technology. (1993). INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEFO). *Federal Information Processing Standards Publication 183*, 183, 100. Retrieved 2023, from <https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub183.pdf>
- Núñez-Sánchez, Y., & González-Torres, A. (2020). Third-party management in software development: proposal of a methodology. *Enfoque UTE*, 11(2), 71-84. <https://doi.org/https://doi.org/10.29019/enfoque.v11n2.621>
- Petersen, B., Benito, G. R., Dovgan, O., & Welch, L. (2013). Offshore Outsourcing A Dynamic, Operation Mode Perspective. *Institut for Strategic Management and Globalization*, 42(2), 211-222. <https://www.webofscience.com/>
- Phillips, M. (Marzo de 2011). *CMMI® for Acquisition (CMMI-ACQ) Primer, Version 1.3 (CMU/SEI-2011-TR-010)*. Retrieved 25 de Octubre de 2022, from Software Engineering Institute, Carnegie Mellon University: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9977>
- POLO, M., PIATTINI, M., & RUIZ, F. (2002). Integrating Outsourcing in the Maintenance Process. *Information Technology and Management*, 3(3), 23. Retrieved 2022, from https://www.researchgate.net/publication/225136879_Integrating_Outsourcin_g_in_the_Maintenance_Process
- Project Management Institute. (2017). *Guía Práctica de AGIL*. Newtown square, Pennsylvania, EE.UU: Project Management Institute, Inc.
- RAHMAN, H. U., RAZA, M., AFSAR, P., & KHAN, H. U. (2021). Empirical Investigation of Influencing Factors Regarding Offshore Outsourcing Decision of Application Maintenance. *IEEE ACCSSES*, 9, 58589-58608. Retrieved 2022, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9404156>
- Salman, R., Daim, T., Raffo, D., & Dabic, M. (2017). Exploring capability maturity models and relevant practices as solutions addressing information technology service offshoring project issues. *International Journal of Management Science and Engineering Management*, 13(3), 147-157. <https://doi.org/10.1080/17509653.2017.1381052>

SHAHZAD, B., ABDULLATIF, A. M., IKRAM, N., & MASHKOOR, A. (2017). Build Software or Buy: A Study on Developing. *IEEE ACCESS*, 24262-24274. <https://doi.org/https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8085093>

The Open Group. (2011). TOGAF® Standard, Version 9.1. Retrieved 2023, from <https://pubs.opengroup.org/architecture/togaf91-doc/arch/index.html>

The Open Group. (2022). *TOGAF® Standard, Version 10*. Retrieved Octobre de 2022, from <https://pubs.opengroup.org/togaf-standard/>

8 Anexos

8.1 Anexo 01. Principales problemas de la externalización de servicios de TI.

Tabla 23. Listado de 23 problemas de la externalización de servicios de TI

N.º	Problemas
1	Establecer y mantener la confianza con las partes interesadas.
2	Gestión de las expectativas de las partes interesadas.
3	Traducir necesidades implícitas y explícitas en requisitos definidos con niveles de calidad acordados.
4	Establecer contratos bien definidos con las partes interesadas, incluidos clientes, proveedores de servicios, proveedores y socios.
5	Revisar el diseño y la implementación del servicio para garantizar una cobertura adecuada de los requisitos.
6	Asegurar la efectividad de las interacciones con las partes interesadas.
7	Gestionar las relaciones entre clientes y proveedores de servicios, así como las relaciones con proveedores y socios, para asegurar que se cumplan los compromisos.
8	Asegurar el cumplimiento de los requisitos legales y reglamentarios.
9	Gestionar la seguridad de los clientes.
10	Gestión de las diferencias culturales entre las partes interesadas.
11	Actividades de seguimiento y control para el cumplimiento constante de los compromisos de prestación de servicios.
12	Seguimiento y gestión de la satisfacción de clientes y usuarios finales.
13	Desarrollar y mantener las competencias que permitan al personal desempeñar eficazmente sus funciones y responsabilidades.
14	Gestión de la satisfacción, motivación y retención de los empleados.
15	Establecer y mantener un ambiente de trabajo efectivo.
16	Mantener una ventaja competitiva.
17	Innovar, crear flexibilidad y aumentar la capacidad de respuesta para cumplir con los requisitos únicos y cambiantes de los clientes.
18	Gestionar cambios tecnológicos rápidos y mantener la disponibilidad, confiabilidad, accesibilidad y seguridad de la tecnología.
19	Captar y utilizar el conocimiento.
20	Transferencia fluida de servicios y recursos.
21	Mantener la continuidad de la prestación del servicio.
22	Captar y transferir el conocimiento adquirido al cliente durante la realización.
23	Medir y analizar las causas de extinción, para evitar que se repitan.

Fuente. (Hefley & Loesche, 2006)