

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

PROPUESTA DE MODELO DE GOBERNANZA DE TI PARA EL DESARROLLO DE SOFTWARE PARA EQUIPOS TIPO VSE (ENTIDADES MUY PEQUEÑAS)

PROYECTO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN GESTIÓN DE LAS COMUNICACIONES Y TECNOLOGÍAS DE LA INFORMACIÓN

ALONSO RENÉ ARÉVALO CAMPOS

aarevalo@ups.edu.ec

DIRECTOR: CARLOS MONTENEGRO

carlos.montenegro@epn.edu.ec

Quito, junio del 2017

DECLARACIÓN

Alonso René Arévalo Campos, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Alonso René Arévalo Campos

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Alonso René Arévalo Campos, bajo mi supervisión.

Carlos Montenegro, Msc
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradezco a mi esposa Sonia y mis hijos Víctor y Rosi por ser el soporte para completar este logro durante todo este tiempo.

Agradezco a Carlos Montenegro por su auspicio y la guía provista para el desarrollo de presente trabajo.

Agradezco a colegas y profesionales que me dieron soporte para poder elaborar el presente documento.

DEDICATORIA

A mi esposa Sonia que, con su amor, comprensión e incondicional apoyo, me impulsa a seguir adelante.

A mis hijos Víctor y Rosi quienes con su responsabilidad y sacrificio me dieron la energía necesaria para poder culminar con éxito la presente maestría.

CONTENIDO

DECLARACIÓN	1
CERTIFICACIÓN	2
AGRADECIMIENTOS	3
CONTENIDO	5
RESUMEN	7
ABSTRACT	8
1. INTRODUCCION	1
Pregunta de Investigación:	4
1.1. Objetivos	4
Objetivo General:	4
Objetivos Específicos:	4
1.2. Marco teórico	4
1.2.1 Fundamentos.	4
1.2.2 El marco de trabajo COBIT 5.	6
1.2.3 La práctica de la Ingeniería de Software.	10
1.2.4 El Proceso de Ingeniería de Software / Ingeniería Web (ISW).	11
Proceso Marco de Ingeniería Software o Web (PMI-S/W).	12
1.2.5 Metodologías Ágiles, SCRUM, XP, KAMBAN.	14
1.2.6 ISO 29110.	15
1.3. Trabajos relacionados.....	16
1.3.1 Evaluación de Ecosistemas de Desarrollo de Software desde la perspectiva de COBIT.....	16
1.3.2 Uso de métricas para el desarrollo de software ágil.....	17
1.3.3 Gobernanza para el desarrollo del producto de forma ágil (DP).....	18
1.4. Estado del arte de la gobernanza.	19
2. ASPECTOS METODOLOGICOS	20
2.1. Metodología utilizada	21
2.2. Desarrollo del modelo para la Gobernanza de desarrollo de software para empresas con áreas de TI, tipo VSE.....	22
2.2.1 Determinar la situación actual de la Gobernanza en las empresas públicas y privadas del medio ecuatoriano.....	22
2.2.2 Cálculo del tamaño de la muestra.....	23
3. RESULTADOS	24
3.1 Resultados de la encuesta para determinar el estado actual de Gobernanza de software en empresas que disponen de áreas de TI, tipo VSE.....	24
3.2 Desarrollo del modelo de Gobernanza de TI para el desarrollo de software para equipos tipo VSE. 30	

3.2.1	Diseño del Modelo de Gobernanza de TI para el desarrollo de software para equipos tipo VSE.	32
3.3	Resultados de la aplicación del modelo MGDS-VSE al caso estudio.	44
3.3.1	Aplicación del modelo de gobernanza de software al caso estudio.	44
3.3.2	Recomendaciones para la evaluación del modelo para la Gestión la de Gobernanza del proceso de desarrollo de software.	48
3.3.3	Evaluación del modelo de gobernanza de software.	48
3.3.4	Resultados de la evaluación del modelo de gobernanza de software.	50
3.4	Discusión	51
4	CONCLUSIONES.	52
	REFERENCIAS BIBLIOGRAFICAS	54
	ANEXOS	56
	ANEXO 1	57
	Cuestionario de la Encuesta	57
	ANEXO 2	64
	Resultado de las valoraciones caracterizadas para las preguntas de la encuesta /Sector Público	64
	ANEXO 3	65
	Resultados de las valoraciones caracterizadas para las preguntas de la encuesta /Sector Privado	65
	ANEXO 4	66
	Detalle del esquema estructural del modelo para tarea de COMUNICACIÓN	66
	ANEXO 5	67
	Detalle del esquema estructural del modelo para tarea de PLANIFICACIÓN	67
	ANEXO 6	68
	Aplicación del modelo MGDS-VSE al caso estudio.	68

RESUMEN

La Gobernanza es un tema actual, tiene gran impacto en el mundo empresarial y está relacionado con la gestión del gobierno Corporativo y de TI en las empresas, tiene como objetivos garantizar la consecución de resultados positivos y el crecimiento de la empresa en función del logro de los objetivos estratégicos de la misma. Este trabajo se centra en la construcción de un modelo que posibilite la aplicación de la Gobernanza requerida para el desarrollo de software, en empresas que tienen un área de TI tipo VSE. El modelo se basa en el marco de gobierno de TI propuesto por COBIT 5, y en la aplicación de una metodología de desarrollo ágil para la gestión del desarrollo de software. La determinación de la situación actual, se efectuó a través de una encuesta, que recopila información sobre el tema de la gobernanza, para llenar la encuesta se dotó de un enlace para acceder a la misma. El análisis de los resultados confirma, que existe un desconocimiento de lo que es y el aporte de la gobernanza para TI y para DS. Los porcentajes de empresas que no gestionan apropiadamente la gobernanza es elevado. El 57% no gestiona apropiadamente la gobernanza de TI y un 64% no gestiona apropiadamente la gobernanza del desarrollo de software. La aplicación del modelo al proyecto tipo fue exitosa, permitió gestionar apropiadamente el proceso de desarrollo de la aplicación y los inconvenientes presentados en el desarrollo del proyecto.

Palabras Clave— Gobernanza, COBIT, SCRUM, GDS, TI, VSE, ISO 29110

ABSTRACT

Governance is a current issue, has a great impact in the business world and is related to corporate governance and IT governance in companies, aims to ensure the achievement of positive results and the growth of the company based on the achievement of the strategic objectives of the same. This work focuses on the construction of a model that allows the application of the Governance required for the development of software, in companies that have an area of IT type VSE. The model is based on the IT governance framework proposed by COBIT 5, and on the application of an agile development methodology for the management of software development. The determination of the current situation was made through a survey, which collects information on the topic of governance, to fill the survey was provided with a link to access it. The analysis of the results confirms that there is a lack of knowledge of what is and the contribution of governance for IT and DS. The percentages of companies that do not properly manage governance is high. 57% do not properly manage IT governance and 64% do not properly manage software development governance. The application of the model to the type project was successful, it made it possible to properly manage the application development process and the disadvantages presented in the development of the project.

Keywords— COBIT 5, GOVERNANCE, IT, VSE, ISO 29110.

1. INTRODUCCION

Gobernanza es un término que en la actualidad tiene gran aplicación en el mundo empresarial y está relacionado con la forma cómo las empresas son dirigidas y controladas [1],[2],[3]. La Gobernanza Corporativa y de Tecnología tiene como objetivo garantizar la consecución de resultados positivos y el crecimiento de la empresa en función del logro de los objetivos estratégicos de la misma. Uno de los productos estratégicos de los procesos de TI constituye la construcción o desarrollo de aplicaciones de software, que son vistas como elementos necesarios para lograr obtener ventajas competitivas en el área de los negocios o servicios.

Este trabajo se centra en la gobernanza requerida para el desarrollo o construcción de software, ya sea en la modalidad de desarrollo propio o por medio de empresas que proveen estos servicios. El estudio se centra en organizaciones públicas o privadas que tienen un área de TI tipo *Very Small Entity* VSE. Una VSE es una empresa, organización, departamento o proyecto que cuenta con un número funcionarios que está dentro del rango de 5 a 25 personas [4],[5].

La gobernanza relacionada con el proceso de desarrollo de software, es fundamental para toda organización, ya que se trata de crear herramientas que permiten automatizar tareas o procesos que están orientados a proveer de mejores productos o servicios a sus clientes internos o externos, desde la perspectiva estratégica del negocio.

El desarrollo del producto de software implica dos elementos fundamentales a tomar en cuenta, i) la Gobernanza del Desarrollo de Software (GDS) que se encarga de que los objetivos estratégicos de la organización estén alineados con los del software y que estos sean alcanzados, y como resultado de esto, se obtenga los beneficios empresariales esperados, y la correspondiente creación de valor para sus productos o servicios [6]; y, ii) el desarrollo del producto software en base a los requerimientos acordados entre las partes, los mismos que deben estar alineados con el logro de objetivos estratégicos de la organización demandante del producto.

De acuerdo con [4],[5] (autores) lo expuesto anteriormente, el principal problema que existe es el desconociendo de la gobernanza, tanto por parte de la organización que requiere

el desarrollo del producto software, así como también de la empresa que desarrolla el mismo [2][7].

La causa de este problema es el desconocimiento que se tiene de la gobernanza por parte de los ejecutivos de las organizaciones privadas o estatales y la carencia o la falta de participación que tienen los ejecutivos del área de TI en la planificación de los objetivos estratégicos de la organización. Esto se debe a que un gran número de gerentes de desarrollo de software, desarrolladores y empresas de desarrollo de software no tienen una comprensión adecuada de los conceptos de la gobernanza del desarrollo de software (GDS) y de sus componentes [2].

La gobernanza relacionada al proceso de desarrollo de software es un tema de actualidad, que aún no se ha resuelto y que no sido tratado con la importancia que se requiere; esto se puede comprobar observando la baja producción de artículos que tratan sobre esta temática. Para confirmar esta aseveración se puede ejecutar una consulta con los siguientes parámetros “governance + software + development”, para el período de tiempo comprendido entre los años 2010 y 2017, en la base de datos de Scopus, esta retorna un total de 160 artículos en la que se incluyen las revisiones. De este total, solo el 35.6% corresponden al área de Ciencias de Computación, lo que da un total de 57 artículos para los 7 últimos años, como se muestra en la figura 1.

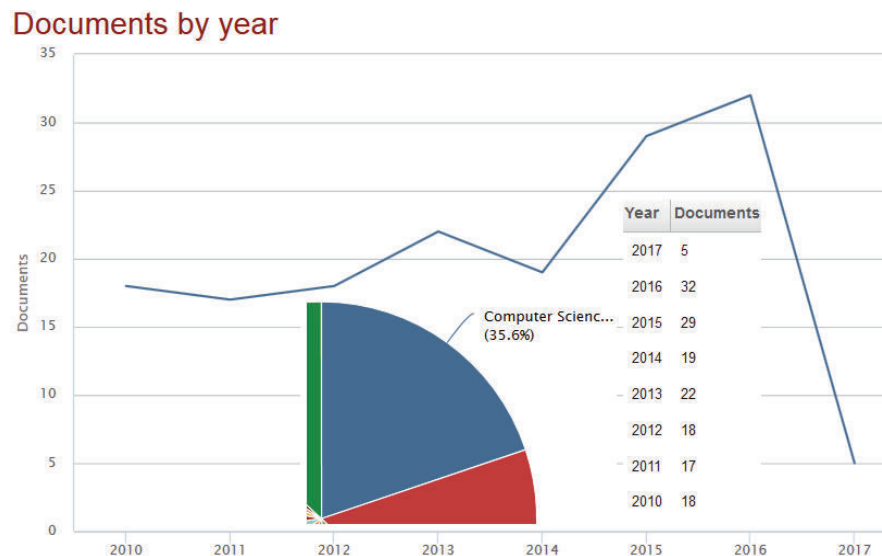


Figura 1. Resultados de la consulta de artículos sobre gobernanza de software en Scopus.

Sobre la base de los argumentos expuestos se puede afirmar que el tema referente a la gobernanza para el desarrollo de productos de software es un aspecto pendiente y requiere de un estudio especializado.

En resumen, tres son los elementos genéricos que justifican la importancia del tema a desarrollarse:

- 1) Existe un desconocimiento del tema por parte de las personas involucradas con el proceso de desarrollo de software desde el nivel ejecutivo hasta los encargados de desarrollar el software y gestionar el proceso [6].
- 2) La adecuada implementación de la gobernanza, en el proceso de desarrollo de software, tiene vital importancia para toda organización, ya que se asegura que los objetivos estratégicos de la misma están alineados con los objetivos del software que se desarrolla [8][9].
- 3) En los últimos años, la gobernanza del desarrollo de software ha sido reconocida como un componente clave para la dirección y alineación de los proyectos de desarrollo de software, posibilitando satisfacer acertadamente las necesidades del negocio [10][8][7].

El presente trabajo busca aportar con una propuesta para solucionar la deficiencia en la aplicación de la gobernanza en el desarrollo de software, planteando un modelo basado en las mejores prácticas de gobierno de TI, que consolide los elementos más importantes de la gobernanza, de tal forma que sea posible la aplicación de la GDS en el desarrollo de aplicaciones de software que requiere la organización, tomando como línea base el marco de referencia para la gobernanza de TI propuesto por COBIT 5.

El modelo propuesto se orienta principalmente a organizaciones que cuentan con áreas de TI con equipos de trabajo tipo VSE, que no han desarrollado un componente estructural para la gobernanza dentro de las mismas y que requieren tener un conocimiento explícito de la gobernanza de TI y su enfoque, para aplicarla a sus procesos de desarrollo de software.

La creación del modelo constituye un aporte metodológico directo para las áreas de TI tipo VSE, puesto que proveerá de un método sencillo y fácil de implementar en la empresa,

su aplicación posibilitará gestionar la gobernanza en los procesos de desarrollo de software, teniendo en cuenta las necesidades y limitaciones de nuestro medio.

Sobre la base de lo que se ha expuesto, es posible realizar la pregunta de investigación y determinar sus objetivos.

Pregunta de Investigación:

¿Un modelo de gobernanza de software (GDS), construido sobre la base de los antecedentes teóricos, prácticos y de los requerimientos empresariales del medio ecuatoriano, permite mejorar el proceso de desarrollo de software?

1.1. Objetivos

Objetivo General:

Desarrollar un modelo de gobernanza de TI para el desarrollo de software en equipos tipo VSE.

Objetivos Específicos:

- i) Establecer el estado del arte de la GDS e identificar los requerimientos relacionados con el medio ecuatoriano.
- ii) Diseñar y desarrollar el modelo de Gobernanza de Desarrollo de Software (GDS), tomando como base los procesos recomendados por la aplicación de las mejores prácticas.
- iii) Validar el modelo propuesto, aplicándolo al proyecto de desarrollo de software en el sitio para el caso de estudio.

1.2. Marco teórico

1.2.1 Fundamentos.

Para responder apropiadamente a los actuales retos que involucra el proceso de desarrollo de software existen herramientas que permiten crear ambientes o entornos de trabajo que favorecen el proceso de desarrollo de software.

El desarrollo de un producto de software debe ser visto desde dos ámbitos, 1) desde la Gobernanza del Desarrollo de Software (GDS), que es el objeto de este trabajo, la GDS

se encarga de que los objetivos estratégicos de la organización sean tomados en cuenta y estén alineados con los del software y que estos sean alcanzados, y como resultado de esto, se obtenga los beneficios esperados, y la correspondiente creación de valor para los productos o servicios de la organización [2]; y, 2) el desarrollo del producto software aplicando los principios de Ingeniería de Software, a partir de los requerimientos acordados entre las partes, los mismos que deben estar alineados con el logro de objetivos estratégicos de la organización demandante del producto.

Todo proceso debe estar sustentado por la aplicación de buenas prácticas en el área relacionada con el proyecto, para el caso del modelo de gobernanza de desarrollo de software, este debe estar sustentado por la aplicación de buenas prácticas de la gobernanza y del desarrollo de software.

Según J. Arnoletto [11], práctica es la repetición de un proceso hasta llegar hacerlo en forma excelente, para satisfacer una necesidad propia o de un tercero, es una capacidad asociada con la persona que la ejecuta. Mientras que buenas prácticas, son un conjunto de principios, medidas, acciones y el manejo de procesos probados para generar valor agregado con calidad y que exceden las expectativas de la organización y garantizan el prestigio de la entidad que las utiliza, son una capacidad institucional de gestión de la calidad [12].

Para la gestión de la gobernanza de TI, se han desarrollado algunos marcos de trabajo, como es el caso de la Norma ISO/ IEC 38500:2008 [13], [10], que proporciona principios para guiar a los directores de las organizaciones (incluyendo, dueños, miembros de la junta, directores, socios, altos ejecutivos o similares) sobre el uso eficaz, eficiente y aceptable de la tecnología de la información (TI) en sus organizaciones. Esta norma también se aplica al gobierno de los procesos de gestión (y las decisiones) relacionados con los servicios de información y comunicación utilizados por la organización, de esta manera los procesos podrían ser controlados por los especialistas de TI de la organización, por proveedores externos del servicio o por unidades de negocios dentro de la organización. Otro marco de gobierno que vale la pena mencionar es el de CALDER-MOIR [14], que en realidad es un meta-modelo que permite coordinar un conjunto de marcos de trabajo (COBIT, BSC, COSO, ITIL, etc.) y organizar el gobierno de TI. Este meta modelo busca obtener el máximo beneficio de los marcos de trabajo y estándares que orquesta, y facilita a las organizaciones la implementación de la norma ISO / IEC 38500, dispone de un conjunto de plantillas de documentos y recursos basados en el estándar internacional ISO38500, en construye en base a los 6 principios de la ISO 38500:

Responsabilidad, Estrategia, Adquisición, Rendimiento, Conformidad, Comportamiento Humano

Para gestionar el gobierno de TI, esta investigación se centra en el marco de trabajo COBIT 5 [2], que es el estándar más utilizado a nivel global sobre el tema de Gobierno de TI.

1.2.2 El marco de trabajo COBIT 5.

COBIT 5 Permite comprender el gobierno y la gestión de las tecnologías de información (TI) de una organización. COBIT 5 [15], es un marco de trabajo integral que ayuda a las empresas a alcanzar sus objetivos para el gobierno y la gestión de las TI corporativas, ayuda a las empresas a crear el valor óptimo desde TI manteniendo el equilibrio entre la generación de beneficios y la optimización de los niveles de riesgo y el uso de recursos. COBIT 5 posibilita que las TI sean gobernadas y gestionadas de un modo holístico, abarcando al negocio completo de principio a fin, incluyendo las áreas funcionales de responsabilidad de TI, considerando los intereses de las partes interesadas internas y externas relacionados con TI. COBIT 5 es genérico y útil para empresas de todos los tamaños, tanto comerciales, como sin ánimo de lucro o del sector público. COBIT 5 se basa en cinco principios claves (mostrados en la figura 2) para el gobierno y la gestión de las TI empresariales.

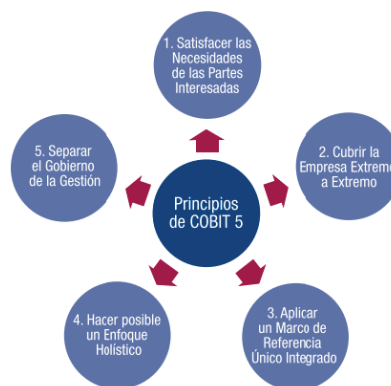


Figura 2. Principios COBIT 5 [15]

Principio 1. Satisfacer las Necesidades de las Partes Interesadas

- Las empresas existen para crear valor para sus partes interesadas manteniendo el equilibrio entre la realización de beneficios y la optimización de los riesgos y el uso de recursos, tal como se visualiza en la figura 3.

COBIT 5 provee todos los procesos necesarios y otros catalizadores para permitir la creación de valor del negocio mediante el uso de TI. Dado que toda empresa tiene objetivos diferentes, una empresa puede personalizar COBIT 5 para adaptarlo a su propio contexto mediante la cascada de metas, traduciendo metas corporativas de alto nivel en otras metas más manejables, específicas, relacionadas con TI y mapeándolas con procesos y prácticas específicos.



Figura 3. Satisfacer las Necesidades de las Partes Interesadas [15].

Principio 2. Cubrir la empresa de extremo a extremo

- COBIT 5 integra el gobierno y la gestión de TI en el gobierno corporativo, tal como visualiza en la figura 4.
- Cubre todas las funciones y procesos dentro de la empresa; COBIT 5 no se enfoca sólo en la “función de TI”, sino que trata la información y las tecnologías relacionadas como activos que deben ser tratados como cualquier otro activo por todos en la empresa.
- Considera que los catalizadores relacionados con TI para el gobierno y la gestión deben ser a nivel de toda la empresa y de principio a fin, es decir, incluyendo a todo y todos
 - internos y externos
 - los que sean relevantes para el gobierno y la gestión de la información de la empresa y TI relacionadas

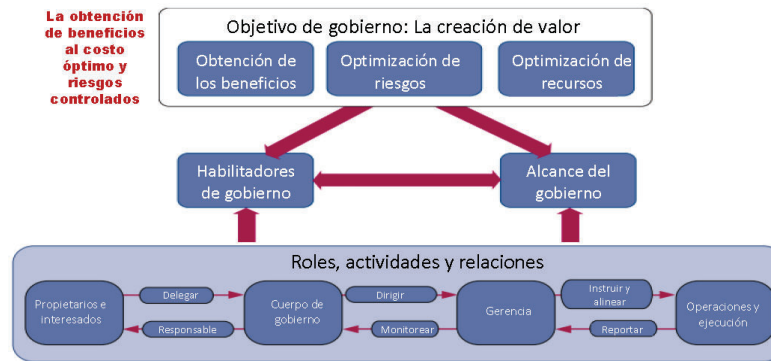


Figura 4. Cubrir la empresa de extremo a extremo

Principio 3: Aplicar un Marco de Referencia Único Integrado - Hay muchos estándares y buenas prácticas relativos a TI, ofreciendo cada uno ayuda para un subgrupo de actividades de TI. COBIT 5 se alinea a alto nivel con otros estándares y marcos de trabajo relevantes, y de este modo puede hacer la función de marco de trabajo principal para el gobierno y la gestión de las TI de la empresa.

Principio 4: Hacer Posible un Enfoque Holístico - Un gobierno y gestión de las TI de la empresa efectivo y eficiente requiere de un enfoque holístico que tenga en cuenta varios componentes interactivos. COBIT 5, define un conjunto de catalizadores (*enablers*) para apoyar la implementación de un sistema de gobierno y gestión global para las TI de la empresa, tal como se visualiza en la figura 5.

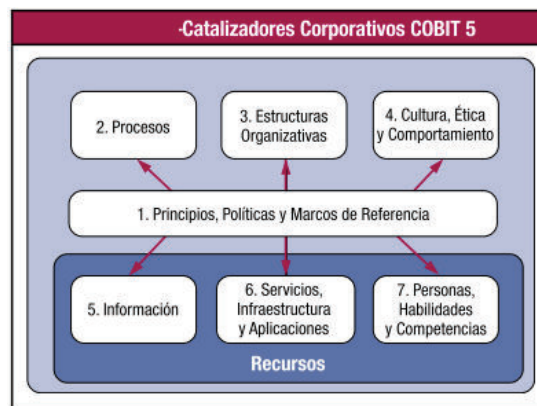


Figura 5 Catalizadores Corporativos [15]

Los catalizadores se definen en líneas generales como cualquier cosa que puede ayudar a conseguir las metas de la empresa. El marco de trabajo COBIT 5 define siete categorías de catalizadores:

- Principios, Políticas y Marcos de Trabajo
- Procesos

- Estructuras Organizativas
- Cultura, Ética y Comportamiento
- Información
- Servicios, Infraestructuras y Aplicaciones
- Personas, Habilidades y Competencias

Principio 5: Separar el Gobierno de la Gestión - El marco de trabajo COBIT 5 establece una clara distinción entre gobierno y gestión. Estas dos disciplinas engloban diferentes tipos de actividades, requieren diferentes estructuras organizativas y sirven a diferentes propósitos [15]. La visión de COBIT 5 en esta distinción clave entre gobierno y gestión es:

- Gobierno. El Gobierno asegura que se evalúan las necesidades, condiciones y opciones de las partes interesadas para determinar que se alcanzan las metas corporativas equilibradas y acordadas; estableciendo la dirección a través de la priorización y la toma de decisiones; y midiendo el rendimiento y el cumplimiento respecto a la dirección y metas acordadas.
- Gestión. La gestión planifica, construye, ejecuta y controla actividades alineadas con la dirección establecida por el cuerpo de gobierno para alcanzar las metas empresariales.

Partiendo de las definiciones de gobierno y gestión, está claro que comprenden diferentes tipos de actividades, con diferentes responsabilidades, esto se visualiza en la figura 6. Sin embargo, dado el papel de gobierno – evaluar, orientar y vigilar – se requiere un conjunto de interacciones entre gobierno y gestión para obtener un sistema de gobierno eficiente y eficaz [15].

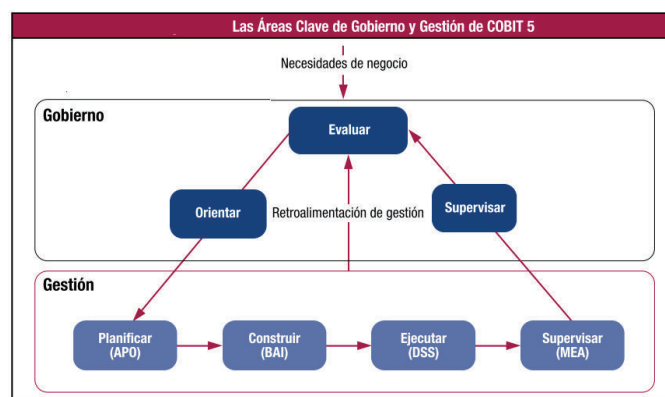


Figura 6. Las Áreas Clave de Gobierno y Gestión de COBIT 5.

1.2.3 La práctica de la Ingeniería de Software.

La esencia de la solución de problemas y, en consecuencia, la esencia de la práctica de la ingeniería de software, fue provista por George Polya, en un libro clásico, *How to Solve It* [16], escrito antes de que existieran las computadoras modernas, en este se plantea que para proveer de la solución a los problemas planteados se deben seguir 4 etapas:

1. Entender el problema (comunicación y análisis).
2. Planear la solución (modelado y diseño del software).
3. Ejecutar el plan (generación del código).
4. Examinar la exactitud del resultado (probar y asegurar la calidad).

Durante la implementación cada una de estas etapas se debe contestar con claridad una serie de cuestionamientos claves para obtener la solución requerida.

- **Entender el problema.** Es difícil de admitir, pero la mayor parte de expertos en el área adopta una actitud inapropiada cuando se le presenta un problema, no se da la importancia necesaria a entender el problema que se está planteando y ya imaginamos la solución, sin considerar si la misma es la que se requiere o la más apropiada. Desafortunadamente, entender no siempre es fácil, es conveniente dedicar el tiempo necesario para responder algunas preguntas sencillas [16]:
 - ¿Quiénes tienen que ver con la solución del problema? Es decir, ¿quiénes son los participantes?
 - ¿Cuáles son las incógnitas? ¿Cuáles datos, funciones y características se requieren para resolver el problema en forma apropiada?
 - ¿Puede fraccionarse el problema? ¿Es posible representarlo con problemas más pequeños que sean más fáciles de entender?
 - ¿Es posible representar gráficamente el problema? ¿Puede crearse un modelo de análisis?
- **Planear la solución.** Una vez que se tiene un entendimiento del problema (o es lo que piensa), no se puede comenzar a escribir el código, antes de hacerlo, se debe hacer un diseño, aunque sea pequeño y contestar los siguientes cuestionamientos [16]:
 - ¿Ha visto antes problemas similares? ¿Hay patrones reconocibles en una solución potencial? ¿Hay algún software existente que implemente los datos, funciones y características que se requieren?

- ¿Ha resuelto un problema similar? Si es así, ¿son reutilizables los elementos de la solución?
- ¿Pueden definirse problemas más pequeños? Si así fuera, ¿hay soluciones evidentes para éstos?
- ¿Es capaz de representar una solución en una forma que lleve a su implementación eficaz? ¿Es posible crear un modelo del diseño?
- **Ejecutar el plan.** Una vez que se cuenta con el diseño para el sistema que quiere construir, el siguiente paso es crear un plan de ejecución. Puede haber desviaciones inesperadas y es posible que descubra un mejor camino a medida que avanza, pero el “plan” le permitirá proceder sin que se pierda [16].
 - ¿Se ajusta la solución al plan? ¿El código fuente puede apegarse al modelo del diseño?
 - ¿Es probable que cada parte componente de la solución sea correcta? ¿El diseño y código se han revisado o, mejor aún, se han hecho pruebas respecto de la corrección del algoritmo?
- **Examinar el resultado.** No es posible asegurar que la solución es perfecta, pero sí de que se ha diseñado un número suficiente de pruebas para descubrir tantos errores como sea posible [16].
 - ¿Puede probarse cada parte componente de la solución? ¿Se ha implementado una estrategia razonable para hacer pruebas?
 - ¿La solución produce resultados que se apegan a los datos, funciones y características que se requieren? ¿El software se ha validado contra todos los requerimientos de los participantes?

“No se debe sorprender de que gran parte de este enfoque tenga que ver con el sentido común. En realidad, es razonable afirmar que un enfoque de sentido común para la ingeniería de software hará que nunca se extravíe” [16].

1.2.4 El Proceso de Ingeniería de Software / Ingeniería Web (ISW).

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de Software o Web. Una acción, es un conjunto de tareas que producen un producto importante, por ejemplo, el diseño de la arquitectura. Una tarea se centra en un objetivo pequeño, pero bien definido, por ejemplo, realizar una prueba unitaria, la misma que produce un resultado tangible.

En el contexto de la ingeniería de Software o Web, un proceso no es una prescripción rígida de cómo elaborar software, por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo.

Proceso Marco de Ingeniería Software o Web (PMI-S/W).

Un proceso marco establece las bases para un proceso de ingeniería S/W completo, mediante la identificación de un número reducido de actividades marco que son aplicables a todos los proyectos de gestión de desarrollo o mantenimiento de aplicaciones web, independientemente de su tamaño o complejidad. Además, el marco de trabajo del proceso abarca un conjunto de actividades generales que son aplicables a través de todo el PMI-S/W [17]. El resumen de las actividades marco del proceso PMI-S/W se detallan en la tabla 1.

Tabla 1 Actividades y acciones del proceso marco de Ingeniería de Software o Web

Actividad	Acciones
1) Comunicación	Formulación, Negociación y Elicitación
2) Planeación	Estimación, Análisis de riesgos, Programación de actividades, Vigilancia
3) Modelado	Análisis, Diseño
4) Construcción	Codificación, Pruebas
5) Despliegue	Entrega, Evaluación

Comunicación. Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software. En esta actividad marco se llevan a efecto 3 acciones que son la Formulación, Negociación y Elicitación. Para crear un modelo de análisis completo para la aplicación que se requiere desarrollar, se debe definir el ámbito total de la misma durante la actividad de formulación de la primera iteración [17].

Planeación. Establece un plan incremental para el desarrollo de la aplicación, en este, se establece las acciones que se van a producir, las tareas técnicas que se llevarán a cabo, los riesgos a los que probablemente se tenga que exponer y mitigar, los recursos que serán necesarios, los productos de trabajo que se producen, y un cronograma de

trabajo. Cualquier proyecto por más complicado que sea, se simplifica si existe un plan con su respectivo cronograma valorado de actividades para el proyecto [17].

Modelado. Abarca la creación de modelos que ayudan a los desarrolladores y al cliente que requiere la aplicación a comprender mejor las necesidades y el diseño que se requiere implementar para poder alcanzar los requisitos previstos [17].

Construcción. Combina tanto la generación de código HTML, XML, Java u otro código similar, más el conjunto de pruebas que se requieren ejecutar, para descubrir errores en el código [17].

Despliegue. Proporciona un incremento parcial de la aplicación que se está desarrollando, el cliente, evalúa y proporciona información sobre los resultados de la evaluación (retroalimentación). La retroalimentación que provee el evaluador, posibilita al proveedor de la aplicación corregir o mejorar características de la misma que requieren ajustes [17].

Las actividades marco del PMI-S/W, son complementadas por actividades de tipo general, que tienen relación con la ejecución del proyecto y se relacionan con la dirección del proyecto, la gestión del servicio, y la mitigación del riesgo. Estas actividades se aplican a lo largo de todo el proyecto de desarrollo del software y ayudan a que el equipo que lo lleva a cabo, pueda administrar el proyecto, controlar el avance, gestionar la calidad, gestionar el cambio y el riesgo. Los resúmenes de las actividades complementarias se visualizan en la tabla 2.

Tabla 2 Actividades complementarias del proceso marco de Ingeniería de Software o Web.

Actividad	Descripción
1) Seguimiento y control del proyecto	Permite que el equipo de software evalúe el progreso comparándolo con el plan del proyecto y tome cualquier acción necesaria para apegarse a la programación de actividades.
2) Administración del riesgo	Evalúa los riesgos que puedan afectar el resultado del proyecto o la calidad del producto.
3) Aseguramiento de la calidad	Define y ejecuta las actividades requeridas para garantizar la calidad del software.
4) Medición	Define y reúne mediciones del proceso, proyecto y producto para ayudar al equipo a entregar software que satisfaga las necesidades de los participantes; puede usarse junto con todas las demás actividades estructurales y complementarias (paraguas o sombrilla).
5) Administración de la configuración	Administra los efectos del cambio a lo largo del proceso del software.

6) Administración de la reutilización	Define criterios para volver a usar el producto del trabajo (incluso los componentes del software) y establece mecanismos para obtener componentes reutilizables.
7) Preparación y producción del producto del trabajo	Agrupa las actividades requeridas para crear productos del trabajo, tales como modelos, documentos, registros, formatos y listas.
8) Revisiones técnicas	Evalúa los productos del trabajo de la ingeniería de software a fin de descubrir y eliminar errores antes de que se propaguen a la siguiente actividad.

1.2.5 Metodologías Ágiles, SCRUM, XP, KAMBAN.

Como resultado de los nuevos retos que se presentan dentro del proceso de desarrollo de software, las metodologías de desarrollo han evolucionado y se han diseñado y construido nuevos enfoques para gestionar el proceso de desarrollo de software con la agilidad suficiente para elaborar respuestas válidas e inmediatas, a estas se les conoce como Metodologías Ágiles [18], las cuales rompen con la rigidez de las metodologías tradicionales, que están caracterizadas por la extensa documentación del proceso de desarrollo, la no flexibilidad ante los cambios y por un alto número de recursos.

Tanto SCRUM, como Programación Extrema (XP) están diseñados para que el producto se construya en procesos iterativos [18][19][20], tal como se aprecia en la figura 8., que van incorporando componentes funcionales al final de cada iteración, esto implica que los equipos desarrollen una parte funcional del producto que potencialmente puede ser liberable al final de cada iteración. Las iteraciones están diseñadas para ser cortas y de duración fija, de esta forma se entrega código funcional cada poco tiempo. Los equipos ágiles (SCRUM, XP, KAMBAN) no tienen tiempo para teorías, generan la documentación esencial para especificar la pila del producto, las historias de usuario (casos de uso) en base a los requisitos fijados para la iteración, sin detenerse a diseñar el modelo UML perfecto en una herramienta CASE, escribir el documento de requisitos perfecto o escribir código que se adapte a todos los cambios futuros imaginables. En vez de eso, se enfocan en que las cosas se hagan, aceptan que pueden equivocarse por el camino, pero también son conscientes de que la mejor manera de encontrar dichos errores es dejar de pensar en el software a un nivel teórico de análisis y diseño y se centran en el desarrollo del mismo, enfocándose en la construcción del producto [14] [15].

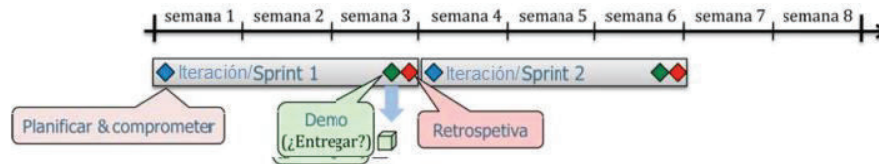


Figura 7. Iteraciones SCRUM / XP

La fuerza y lo duro de estas metodologías es que estamos obligados a adaptarlas a nuestras situaciones específicas, su asimilación es producto de un proceso de aprendizaje continuo. “Henrik demuestra que lo importante no es la herramienta con la que empiezas, sino la forma en la que mejoras constantemente el uso de esa herramienta y expandas tu conjunto de herramientas con el tiempo” [19].

Ninguna herramienta es completa, ni es perfecta, no especifican o dicen todo lo que se debe hacer, solo proporcionan ciertas restricciones y directrices. “Usar las herramientas adecuadas te ayudará a triunfar, pero no garantizará el éxito. Es fácil confundir el éxito/fracaso del proyecto, con el éxito / fracaso de la herramienta” [18].

1.2.6 ISO 29110.

Constituye un conjunto de normas y guías de acuerdo con un conjunto de características y necesidades de las VSE. Las guías se basan en subconjuntos de elementos de normas apropiadas, denominadas perfiles VSE.

ISO / IEC 29110 está dirigida a productores de normas, vendedores de herramientas y vendedores de metodología. Introduce los conceptos para perfiles normalizados internacionales de software para VSEs. Establece la lógica detrás de la definición y aplicación de perfiles estandarizados. Especifica los elementos comunes a todos los perfiles estandarizados (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO / IEC 29110 [2].

Está orientado al desarrollo de software. El proyecto puede cumplir un contrato externo o interno. El contrato interno entre el equipo del proyecto y su Cliente no necesita ser explícito. La Guía brinda procesos de Gestión de Proyectos e Implementación de Software que integran prácticas basadas en la selección de la ISO / IEC 12207: 2008, Sistemas e ingeniería de software - Procesos de ciclo de vida del software e ISO / IEC 15289: 2006 Software de ciclo de vida proceso de productos de información (Documentación) elementos de normas[3].

Los beneficios de utilizar la Guía, VSE son los siguientes: 1) Disponer de un conjunto acordado de requisitos de proyecto y productos esperados para ser entregados al Cliente. 2) Disponer de un proceso disciplinado de gestión, que proporciona visibilidad del proyecto y acciones correctivas de los problemas y desviaciones del proyecto. 3) Se sigue un proceso sistemático de implementación de software que satisface las necesidades del Cliente y asegura productos de calidad.

1.3. Trabajos relacionados.

El tema de la Gobernanza para el desarrollo de software, probablemente nunca dejará de ser un tema de actualidad, puesto que las empresas, organizaciones, la tecnología, e industria, están en constante cambio, y en todas estas áreas existe componentes de software que están involucrados, por lo tanto, está involucrada la Gobernanza para el desarrollo de software al interior de las empresas u organizaciones que desarrollan software y también las que requieren de este servicio.

Un ejemplo de esto es la aparición de nuevos marcos de trabajo para desarrollo distribuidos como por ejemplo los ecosistemas desarrollo de software [21], marco de gobernanza para el trabajo colaborativo, modelo de desarrollo global GTM [7], [22], renta o contratación de servicios de provisión de software [23], subcontratación de desarrollo de software (SDO). Un SOP es una forma mejorada de la relación de subcontratación convencional, en la que se establece una asociación estratégica de confianza para el desarrollo de software entre las organizaciones (cliente y proveedor), sobre la base de una renegociación de tareas, nuevos compromisos y ajustes a los existentes etc.

Para este trabajo se aborda las temáticas más recientes que tienen relación con el tema que se investiga.

1.3.1 Evaluación de Ecosistemas de Desarrollo de Software desde la perspectiva de COBIT.

En el trabajo de Yosvany Castaño Gil [21], se analiza el hecho de que un proceso de desarrollo de software en la actualidad no es abordado por una sola empresa, sino por un conjunto de dos o más de ellas, cada una de las cuales asume un determinado rol dentro del proceso de desarrollo de software. Publicado en el año 2016, y trata de cómo aplicar y utilizar el marco de trabajo COBIT 5 para evaluar el nivel de uso de las herramientas y buenas prácticas presentes en un ecosistema de desarrollo de software. El término ecosistema representa la unión simbiótica de varios elementos, es un concepto heredado de los ecosistemas biológicos y de negocios.

Este trabajo [21], recomienda y resalta los beneficios que se obtienen al aplicar el marco de gobierno COBIT 5, para gestionar la gobernanza del proceso de desarrollo de software, donde analizan, los niveles, operativo, táctico y estratégico.

Propone aplicar los siguientes dominios de COBIT 5: Alinear, Planificar y Organizar (APO), Construir, Adquirir e Implementar (BAI) y Supervisar, Evaluar y Valorar (MEA) y un sub conjunto de las prácticas de estos dominios que están relacionadas con el desarrollo de software en ecosistemas de desarrollo de software [21], [24]. Se valida el modelo propuesto aplicándolo a tres empresas que desarrollan software en ambientes ECOS y luego en el análisis de los resultados, se concluye que es posible aplicar el marco de trabajo COBIT 5, para lograr obtener los objetivos previstos por la organización en ambientes de desarrollo [20].

1.3.2 Uso de métricas para el desarrollo de software ágil.

El estudio de Sikandar Ali y Siffat Ullah Khan [23], resalta que los métodos ágiles proponen el uso de algunas métricas clave, pero se sabe poco del uso real de estas métricas en los equipos ágiles. El objetivo de este trabajo es aumentar el conocimiento de los beneficios y las consecuencias del uso de métricas ágiles en la industria del software.

Propone el uso de metodologías ágiles como SCRUM y XP, para el desarrollo de productos software de alta calidad y además señala que en la actualidad la industria de software ha optado por usar los métodos ágiles para el desarrollo de software, es publicado en el 2015.

Trata sobre el uso de métricas en el desarrollo ágil de software industrial. Se identificaron 774 trabajos, que se redujeron a 30 estudios primarios a través un proceso de selección. Los resultados indican que las razones y los efectos del uso de métricas se centran en las siguientes áreas: planificación del sprint, seguimiento del progreso, medición de la calidad del software, solución de problemas dentro del proceso de construcción del software y motivación de las personas.

Las métricas más influyentes en función del número de ocurrencias encontradas en los estudios primarios analizados son la estimación de velocidad, el esfuerzo y el número de defectos. El análisis cualitativo de las métricas, muestra que la satisfacción del cliente,

la deuda técnica, el estado de la construcción y el progreso del trabajo son métricas muy importantes.

Se concluye en que el uso de métricas en el desarrollo de software ágil es similar al desarrollo de software tradicional. Los proyectos y los sprints (iteraciones o entregas parciales) necesitan ser planificados y rastreados. La calidad debe medirse. Los problemas necesitan ser identificados, fijados y solucionados.

1.3.3 Gobernanza para el desarrollo del producto de forma ágil (DP).

El estudio de Anita Friis Sommer et al [8], señala que la gestión del desarrollo de productos (DP) está cambiando a través de la aparición e implementación de principios ágiles en los marcos de desarrollo de productos existentes. Se introduce el concepto de la gobernanza de Desarrollo de Productos (PD) de soluciones híbridas ágiles / etapa-puerta a través de un estudio comparativo que incluye cinco casos de empresas, apoyado por una revisión de la literatura existente del tema en estudio, es publicado en el 2014 [8]. Se menciona a la gobernanza como un aspecto emergente de la gestión del DP (Desarrollo del Producto) y también que es un área de creciente atención [8].

Se define la gobernanza de PD como un conjunto de sistemas de gestión, reglas, protocolos, relaciones y estructuras que proporcionan el marco dentro del cual se toman las decisiones para lograr el negocio o la motivación estratégica deseada [15], y requiere de una estructura de gobierno general con activos de gobierno de apoyo (adaptado de Weill y Ross [25]).

Agile PD es una de las últimas ramas de investigación que recientemente surgieron dentro de la gestión industrial PD. Los métodos ágiles se especializan en el manejo de la PD de alta complejidad, incluyendo la participación activa del cliente. Los métodos ágiles se originan en la industria de desarrollo de software y ahora se han adaptado a la nueva investigación de DP en el contexto de la industria manufacturera.

1.4. Estado del arte de la gobernanza.

Tabla 3 Artículos Referenciados Estado del Arte

Ref.	Autor	Artículo
[7]	J. Noll, S. Beecham, I. Richardson, and C. N. Canna	A Global Teaming Model for Global Software Development Governance: A Case Study,
Aporte. Modelo marco de equipo global que especifica las prácticas y recomendaciones para el Desarrollo de Software Global (GSD). Como tal, sirve como un medio para organizar las prácticas pertinentes a la gestión del desarrollo de software.		
[8]	A. F. Sommer, I. Dukovska-popovska, and K. Steger-jensen	Agile Product Development Governance – On Governing the Emerging Scrum / Stage-Gate Hybrids
Aporte. Introduce la gobernanza de desarrollo de productos (PD) para soluciones híbridas ágiles.		
[15]	ISACA	<i>Un Marco de Negocio para el Gobierno y la Gestión de las TI de la Empresa - COBIT 5</i>
Aporte. COBIT 5 contiene el marco para el gobierno y la gestión de las TI de la empresa, se construye sobre cinco principios básicos, que quedan cubiertos en detalle e incluyen una guía exhaustiva para el gobierno y la gestión de las TI de la empresa		
[21]	Yosvany Castaño Gil	Evaluación de Ecosistemas de Desarrollo de Software desde la perspectiva de COBIT
Aporte. Analiza la gobernanza en ecosistemas de desarrollo de software y presenta una propuesta de cómo se podría utilizar el marco de trabajo COBIT 5 para evaluar el nivel de uso de las herramientas y buenas prácticas en un ecosistema de desarrollo de software.		
[22]	C. Manteli, B. Van Den Hooff, and H. Van Vliet	The effect of governance on global software development: An empirical research in transactive memory systems
Aporte. Propone un método analítico para examinar las decisiones de gobernabilidad en el Desarrollo de software global (GSD) y su efecto en los sistemas de memoria transactiva, como una herramienta de "causa y efecto" para mejorar la colaboración de equipos de software globales.		
[23]	Sikandar Ali y Siffat Ullah Khan	Critical Success Factors for Software Outsourcing Partnership (SOP): A Systematic Literature Review
Aporte. Identifica varios factores que son significativos para los proveedores en la conversión de una relación contractual de subcontratación a la asociación. Factores como interdependencia mutua y valores compartidos, confianza mutua, comunicación eficaz y oportuna, proximidad organizacional y producción de calidad.		

[24]	O. M. Gabriel	Análisis de grandes ecosistemas de software abierto
Aporte. Define ecosistema de software y los actores que se encuentra involucrados, ofrece una serie de recomendaciones para la creación de un ecosistema de código abierto.		
[25]	P. Weill and J. W. Ross	How Top Performers Manage IT Decisions Rights for Superior Results
Aporte. Propuesta para tomar decisiones clave en temas de TI, sobre la base de un estudio de 300 empresas en todo el mundo.		

2. ASPECTOS METODOLOGICOS

Los aspectos metodológicos permiten describir la forma en la que se realiza la investigación, según Roberto Hernández Sampieri et al, “La investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno” [26].

Para la realización de este trabajo se utilizó una metodología de investigación híbrida. El enfoque cuantitativo implica el desarrollo de un conjunto de procesos secuenciales y probatorios. Se parte de una idea (modelo de Gobernanza de desarrollo de software), que va acotándose y, una vez delimitada (aplicada e empresas cuya área de TI, sea de tipo VSE), se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica, se define el tamaño de la muestra, se efectúa la recolección de los datos, se analiza los datos, etc. El enfoque cuantitativo de esta investigación está directamente relacionado con el proceso de determinación del estado actual de la Gobernanza en nuestro medio, el mismo que se lleva a cabo a través del diseño y recolección de datos por medio de la encuesta. En relación con el enfoque cualitativo en este estudio se tiene el diseño y validación del modelo de la Gobernanza para el desarrollo de software, en la cual se efectúa una recolección de información sin asignar valores numérico, al igual que en el enfoque cualitativo, se parte de una idea, se realiza una revisión de la literatura referente a la temática del estudio, se requiere elaborar un diseño conceptual, recolección de datos, análisis de los datos, interpretación de los resultados, todas estas fase son desarrolladas para el diseño y creación del modelo de la gobernanza de software, en los enfoques se analizan las mediciones obtenidas, y se establece una serie de conclusiones respecto de la(s) hipótesis [26].

El enfoque descriptivo del estudio está dado por la revisión bibliográfica minuciosa efectuada sobre la base de conocimientos existente, que guarda relación con la temática

del estudio, en este caso el gobierno de TI y las metodologías de desarrollo ágil, con el objeto de resaltar las características fundamentales de los mismos. El enfoque prescriptivo del estudio está provisto por la propuesta metodológica que se presenta como resultado de esta investigación, el método para gestionar la gobernanza del desarrollo de software en empresas que disponen de un área de TI tipo VSE [27].

2.1. Metodología utilizada.

El desarrollo de este estudio se lo hace en cuatro fases:

- 1) Determinación del estado del arte de la Gobernanza para el Desarrollo de software, revisión minuciosa de fuentes bibliográficas referentes a la temática de la gobernanza y el desarrollo de software, relacionados con los marcos de referencia, modelos, métodos y ejemplificaciones sobre el tema.
- 2) Determinación de la situación actual. Investigación de campo, para recabar información sobre el tema de la gobernanza en organizaciones públicas y privadas de la ciudad de Quito, que cuentan con áreas de TI, catalogadas como VSE.
- 3) Diseño y desarrollo del modelo. Crear el modelo para la gobernanza del desarrollo de software en las organizaciones que cuentan con áreas de TI, tipo VSE.
- 4) Validación del modelo, mediante su aplicación controlada, en la organización tomada como caso de estudio y presentación de resultados.

La justificación práctica para el uso de esta metodología, está relacionada con la estructura genérica de una metodología de investigación, que en este caso es híbrida, y como tal se estructura con dos componentes cuantitativo y cualitativo. La estructura de la metodología planteada se alinea completamente con las etapas de un modelo de investigación sea este cuantitativo o cualitativo ya que en los casos se requiere fases que implican una revisión bibliográfica minuciosa de la temática relacionada con la investigación, la concepción del diseño del estudio, la recolección de datos, y finalmente el análisis de resultados.

Para el caso de la aplicación del modelo la secuencia de implementación de los componentes, también resulta lógica, primero el componente estático y luego el dinámico. Una vez que el componente estructural y dinámico de la gobernanza están establecidos en la organización, la aplicación del componente dinámico a través de la ejecución de procesos de medición y control apropiados, le permiten a la organización consolidar la estructura de la misma y aprender a gestionarla [2], [5], [14]. Las implicaciones prácticas descritas se las valida aplicando el modelo desarrollado a un caso de estudio, en el

Hospital de Especialidades Eugenio Espejo, que implementa un software específico para la administración y gestión de partes operatorios en quirófanos.

2.2. Desarrollo del modelo para la Gobernanza de desarrollo de software para empresas con áreas de TI, tipo VSE.

El desarrollo de software es una actividad vital de TI que continuamente es estudiada y analizada con el objeto de desarrollar nuevos modelos o marcos de trabajo, que posibiliten mejorar y hacer más eficiente al proceso de desarrollo de software, de tal forma que sea posible satisfacer las demandas de las aplicaciones de software que se requieren en la actualidad, las mismas que cada vez más extensas y complejas. Uno de los marcos de trabajo al cual no se le ha dado la debida atención es la de aplicar los principios de gobernanza al dominio de desarrollo de software.

2.2.1 Determinar la situación actual de la Gobernanza en las empresas públicas y privadas del medio ecuatoriano.

Para la determinación de la situación actual de la aplicación de la gobernanza en el proceso de desarrollo de software, en organizaciones públicas y privadas de la ciudad de Quito, que cuentan con áreas de TI, catalogadas como VSE, se implementó una encuesta que consta de 20 preguntas distribuidas en 3 bloques: Bloque 1), preguntas [1-4], consulta información relacionada con el tipo de empresa, Bloque 2), preguntas [5-14], consulta información relacionada con la aplicación de la Gobernanza de TI en la organización, y Bloque 3), preguntas [15-20], consulta información relacionada con la aplicación de la Gobernanza en el proceso de desarrollo de software. El detalle de la encuesta se incorpora en el anexo 1.

Para la recopilación de los datos se implementó la encuesta en una herramienta web (survio.com), que permite la creación de encuestas en línea, para solicitar el llenado de la misma se solicitó la ayuda a la Asociación Ecuatoriana de Software AESOFT (aesoft.com.ec), y a la Subsecretaría de Gobierno Electrónico, que forma parte de la Secretaría Nacional de la Administración Pública del Ecuador. (administracionpublica.gob.ec). El enlace para acceder a la encuesta en portal web **survio.com** es: <https://www.survio.com/survey/d/X7C1W9S4D3M2C2Y7E?preview=1>.

2.2.2 Cálculo del tamaño de la muestra.

Para determinar el tamaño de la muestra se realizó una investigación de campo en relación con el número de empresas que desarrollan software en el Ecuador, para lo que se tomó como referencia el número de empresas de la Asociación de Software del Ecuador que desarrollan software. A la fecha de realización de la investigación, 10 de abril del 2017, el número de empresa existentes en el registro de AESOF es de 163, de las cuales se estima que al menos 130 de estas desarrollan software, esta información se obtiene luego de verificar la información que consta en el directorio de empresas asociadas a AESOFT, para el caso del sector público se estima en 20 las empresas que desarrollan software de una manera formal, este cantidad se calcula de la siguiente manera, se estima en que son 8 los ministerios que desarrollan software formalmente, tomando en cuenta su tamaño, de igual manera se estima que son 8 los municipios y 4 empresas estatales las que desarrollan software formalmente, aplicando el mismo criterio, su tamaño. De esta forma se estima que la cantidad de empresas públicas o privadas que desarrollan software está en 150.

La fórmula para calcular el tamaño de muestra es la siguiente:

$$n = \frac{N \times Z_a^2 \times p \times q}{d^2 \times (N - 1) + Z_a^2 \times p \times q}$$

Ecuación 1 Cálculo del tamaño de la muestra

En donde, N = tamaño de la población Z = nivel de confianza, P = probabilidad de éxito, o proporción esperada Q = probabilidad de fracaso D = precisión (Error máximo admisible en términos de proporción).

La temporalidad en la que se fija el presente análisis es el año 2017, entre los meses de abril y mayo que es cuando se realiza la investigación.

En la tabla 4 se presenta el detalle de los resultados del cálculo del tamaño de la muestra.

Tabla 4. Resultados del cálculo de la muestra.

Estimar el tamaño de la muestra	
Total, de la población (N) (Si la población es infinita, dejar la casilla en blanco)	150
Nivel de confianza o seguridad (1- α)	95%
Precisión (d)	4%
Proporción (valor aproximado del parámetro que queremos medir) (Si no tenemos dicha información $p=0.5$ que maximiza el tamaño muestral)	5%
Tamaño Muestral (n)	65
El tamaño muestral ajustado a pérdidas	
Proporción esperada de pérdidas (R)	15%
Muestra Ajustada a las Pérdidas	77

3. RESULTADOS

Los resultados del presente trabajo se analizan desde dos perspectivas, la primera en base a los resultados de la encuesta, que presentan la información del estado actual de la Gobernanza en empresas privadas o públicas que cuentan con áreas de TI tipo VSE, y la segunda presenta los resultados de la aplicación del modelo propuesto, a un proyecto de desarrollo de software para el Hospital de Especialidades Eugenio Espejo, en el cual se implementó una aplicación web para la administración y gestión de partes operatorios en quirófano.

3.1 Resultados de la encuesta para determinar el estado actual de Gobernanza de software en empresas que disponen de áreas de TI, tipo VSE.

El número total de empresas que atienden la petición de llenado es de 74, el acceso a la aplicación WEB que contiene la encuesta se realizó mediante provisión de un enlace. Para la solicitud de llenado se contó con la colaboración la AESOF y la Subsecretaría de Gobierno Electrónico de la Secretaría Nacional de la Administración Pública.

Los resultados de la aceptación a la petición de llenado de las encuestas se presentan en siguiente cuadro resumen, en el cual se detallan la cantidad de las encuestas recibidas por tipo de empresa y número de empleados en su área de TI:

Tabla 5 Resumen de las encuestas por Tipo de Empresa

No. Empleados	Total, Encuestas	Pública	Privada
[1 - 4]	20	3	17
[5 - 25] VSE	33	15	18
[+ 25]	21	9	12
Total, encuestas	74	27	47

Prueba de la valides de encuesta.

La prueba para validar la calidad de los resultados y la escala de medida utilizada en la encuesta, se la realiza mediante la determinación del coeficiente Alfa de Cronbach [28][29], que permite estimar la fiabilidad de un instrumento de medida.

$$\alpha = \frac{k}{k-1} \left[1 - \frac{\sum_i^2}{S_r^2} \right] \quad \alpha = \frac{16}{15} \left[1 - \frac{12,980}{102,21} \right] = 0,931$$

P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	VarP	α
0,6671	0,76	0,79	0,79	0,82	0,694	0,678	0,937	0,757	0,809	0,81	0,696	1,029	0,952	0,887	0,902	102,215	0,931213

Donde α = coeficiente de Cronbach, K = número de preguntas, Pi = Varianza P (VarP) para cada uno de los ítems de la encuesta en forma individual, Varianza P (VarP) = Varianza promedio de todas las valoraciones efectuadas a cada uno de los ítems de la encuesta.

Como criterio general, se sugieren las siguientes recomendaciones para evaluar los coeficientes de alfa de Cronbach (tabla 6):

Tabla 6 Criterios de valoración para α de Cronbach

>0.9 es excelente	>0.8 es bueno	>0.7 es aceptable	>0.6 es cuestionable
-------------------	---------------	-------------------	----------------------

El valor obtenido para el coeficiente α de Cronbach de 0.93, lo cual avala la calidad de la encuesta y sus resultados.

Análisis de resultados de la encuesta.

En concordancia con el objeto de este estudio, el análisis de los resultados se centra en las encuestas de las empresas cuya área de TI está clasificada como tipo VSE. Para el análisis de los mismos, se consideraron valores de tendencia central, los mismos que dan la posibilidad de representar con un número al conjunto de datos. El valor en el que

se basa este análisis es la Media Aritmética, la misma que parte del principio de la esperanza matemática o valor esperado.

De los resultados obtenidos se puede ver que el comportamiento de las empresas del sector público y privado son similares, en relación a como gestionan la Gobernanza de TI y la Gobernanza de los procesos de desarrollo de software.

Del análisis individual de la Gobernanza de TI y la de DS, se puede apreciar que en el sector público se gestiona de mejor manera la Gobernanza de TI, ya que se tiene mejores valoraciones para 7 de las 10 preguntas relacionadas con la Gobernanza de TI, por el contrario, la Gobernanza para el Desarrollo de software se lo gestiona mejor en el sector privado, 4 de las 6 preguntas sobre la Gobernanza para el desarrollo de software están mejor valoradas.

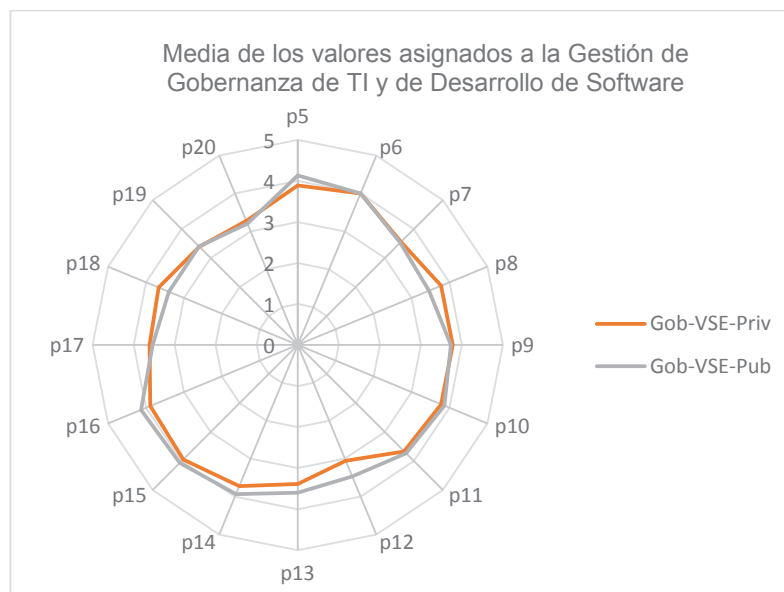


Figura 8 Continuidad de Media de los valores asignados a la Gestión de Gobernanza de TI y de Desarrollo de Software

En la figura 19, se puede apreciar de forma continua, cómo es el comportamiento de la gestión de la Gobernanza de TI y del desarrollo de software, tanto para las empresas privadas y públicas.

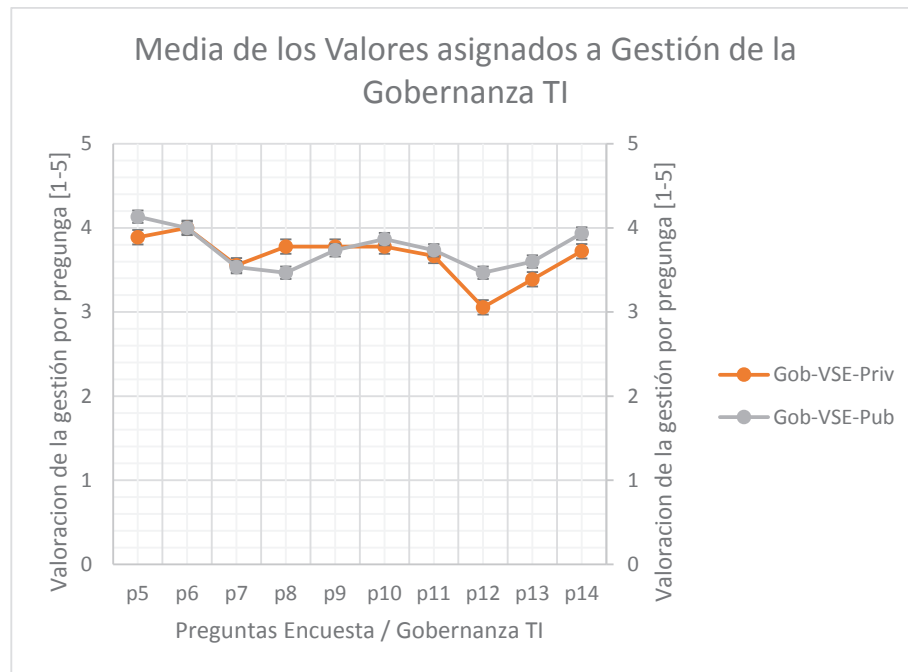


Figura 9. Comportamiento de la Media de los valores asignados a la Gestión de Gobernanza de TI

Un aspecto relevante que se puede apreciar en la figura 20, para la Gobernanza de TI en el sector Privado, es la baja valoración alcanzada por la pregunta p12, es la más baja de todas, y mide si la organización es más eficiente en el uso de sus recursos financieros, tecnológicos y humanos con el soporte de TI. Al parecer no se tiene claro el aporte que provee a la Organización en su conjunto el área de TI.

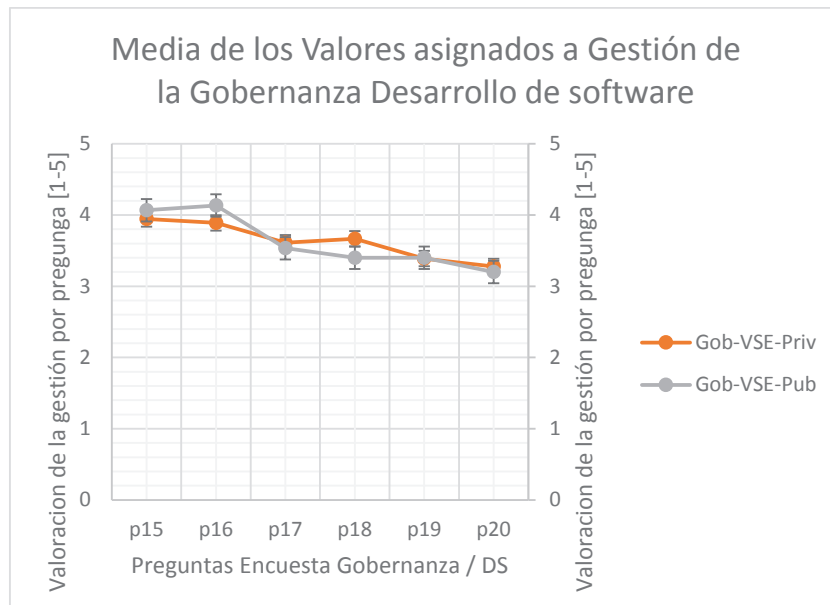


Figura 10 Comportamiento de la Media de los valores asignados a la Gestión de Gobernanza de DS

De la revisión del gráfico de la figura 21, correspondiente a la valoración de la Gobernanza del DS, se puede apreciar que las valoraciones obtenidas para las preguntas p19 y p20, son las más bajas, lo que se puede interpretar como que las organizaciones públicas y privadas tienen deficiencias en la implementación de los mecanismos de control requeridos para la gestión del riesgo, así como para supervisar, evaluar y valorar la conformidad de los productos.

Análisis de resultados observados en la encuesta realizada en la empresa donde se ejecuta el proyecto piloto.

Los resultados de la encuesta sobre el tema de la Gobernanza de TI y de DS en la empresa en la que se aplica el modelo se muestran en el siguiente cuadro. La fila correspondiente a la valoración contiene los valores asignados a cada una de las preguntas.

Tabla 7. Resultados de la encuesta de la empresa en la que se evalúa el modelo.

Pregunta	Gobernanza de TI										Gobernanza de DS					Promedio	
	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19		p20
Valoración	4	4	4	5	4	4	4	4	3	5	3	4	4	3	3	3	
Promedio	4,1333	4	3,5333	3,4667	3,7333	3,8667	3,7333	3,4667	3,6	3,9333	4,0667	4,1333	3,5333	3,4	3,4	3,2	3,7
Moda	4	4	3	3	3	4	4	3	3	4	4	4	4	3	3	3	

Se puede visualizar que la parte correspondiente a la gestión de la Gobernanza de TI, se la está efectuando acertadamente, pero existen problemas con el tema de la Gobernanza de desarrollo de software, lo cual coincide con lo expresado anteriormente, que la empresa pública gestiona apropiadamente la Gobernanza de TI, e inadecuadamente la Gobernanza de DS.

Análisis de los resultados para determinar el número de empresas que gestionan apropiadamente la Gobernanza de TI y de DS

Tomando como base la caracterización de las valoraciones a cada una de las preguntas de la encuesta, los valores 4 y 5, certifican que la gestión es correcta (4 satisfactoriamente y 5 totalmente). Por lo tanto, se puede asegurar que si el promedio de valoraciones provistas para la Gobernanza de TI y de DS, son iguales o superiores a 4, en la empresa se está gestionando satisfactoriamente la Gobernanza. Si el promedio es mayor o igual a 3 pero menor que 4, se estará gestionando medianamente la Gobernanza. Si el promedio es menor que 3 la gestión será deficiente.

Tabla 8. Cuadro resumen de la gobernanza en empresas con áreas de TI tipo VSE

33	Empresas	GTI defic	GTI median	GTI aprop	GDS defic	GDS median	GDS aprop	G-TI/DS aprop
15	Públicas	0	8	7	2	8	5	3
18	Privadas	2	9	7	4	7	7	6
Total x tipo de Gestión		2	17	14	6	15	12	9
%		6,06	51,52	42,42	18,18	45,45	36,36	27,27

En la tabla 8 se puede apreciar los siguientes resultados globales del análisis de universo de empresas (33), cuya área de TI está catalogada como tipo VSE, que atienden la petición de llenado de encuesta.

- El 6% gestiona deficientemente la Gobernanza de TI y pertenecen al sector privado. El 18% gestiona deficientemente la gobernanza del DS, de las cuales el 67% del total de empresas que gestionan deficientemente pertenece al sector privado.
- El 51% gestiona medianamente la Gobernanza de TI, de las cuales el 53% pertenecen al sector privado. El 45% gestiona medianamente la Gobernanza de DS, de las cuales 53% pertenece al sector público.
- El 42% gestiona apropiadamente la Gobernanza de TI, de las cuales 50% pertenece al sector privado. El 36% gestiona apropiadamente la Gobernanza de DS, de las cuales 59% pertenece al sector privado.

- El 27% gestiona apropiadamente la Gobernanza de TI y DS, de las cuales 67% pertenece al sector privado.

La encuesta se implementó en la herramienta de software **Survio**, que posibilita la creación de encuestas online (survio.com). El enlace para acceder a la encuesta es el siguiente:

<https://www.survio.com/survey/d/X7C1W9S4D3M2C2Y7E?preview=1>

3.2 Desarrollo del modelo de Gobernanza de TI para el desarrollo de software para equipos tipo VSE.

La Gobernanza es un concepto multidimensional, que abarca un conjunto de arreglos y prácticas que una organización pone en marcha para asegurar que sus actividades sean administradas adecuadamente y apropiadamente como son: la protección, la rendición de cuentas, la gestión de riesgos, el cumplimiento de los compromisos, el control, la corrección, la supervisión funcional, la asignación de recursos y la gestión de la capacidad. La gobernanza no es la operación directa y el control de las actividades empresariales de la organización, sino la infraestructura necesaria para asegurar su gestión a satisfacción de las partes interesadas directas e indirectas [30].

El diseño del modelo de gobernanza de TI para el desarrollo de software que se propone en este estudio, se basa en la aplicación del concepto de célula de gobierno propuesto por Paul L. Bannerman en su trabajo titulado Gobernanza del desarrollo de software: una perspectiva de meta-gestión [30] y por la aplicación del marco de trabajo para el gobierno y la gestión de las TI en la empresa provista por COBIT 5. Parte de este conjunto de recomendaciones tienen que ver con la gestión de la gobernanza de desarrollo de software.

El modelo conceptual de la célula de gobierno que Bannerman propone, está conformado por los siguientes componentes:

1. Propósito.

Contiene la descripción de los resultados que se espera alcanzar con la aplicación de la célula de gobierno y motiva las acciones de la misma.

2. Estructuras.

Contiene un conjunto de una o más entidades organizativas a través de las cuales se promulga el trabajo de la célula, tales como un consejo, un comité directivo, consejo o grupo de trabajo, un rol o perfil específico.

3. Procesos.

Contiene una colección de tareas interrelacionadas o procesos que se promulgan para lograr el propósito de la célula.

4. Mecanismos relacionales.

Arreglos o mecanismos que se deben implementar para apoyar y habilitar el propósito de la célula, facilitando la interacción y la comunicación con las partes interesadas dentro y fuera del dominio.

La representación gráfica del modelo conceptual de una célula de gobierno se visualiza en la figura 13.



Figura 11. Esquema de la Célula de Gobernanza Conceptual

Para poder responder a los retos que implica la Gobernanza para el desarrollo de software (GDS), en este estudio se propone analizarlos desde dos puntos de vista:

- 1) El componente estructural o estático del gobierno, con sus correspondientes estructuras jerárquicas y cadenas de responsabilidad, autoridad y comunicación para empoderar a las personas dentro de la organización donde se desarrolla el producto de software [5][7][8].
- 2) El componente dinámico del gobierno, con sus correspondientes estructuras y mecanismos de medición y control, para verificar que los desarrolladores de software, directores de proyectos y otras personas dentro de la organización donde se desarrolla el producto de software, realicen apropiadamente sus funciones y responsabilidades [5][7][8].

3.2.1 Diseño del Modelo de Gobernanza de TI para el desarrollo de software para equipos tipo VSE.

De acuerdo con el análisis expuesto para la gestión de la gobernanza para el desarrollo de software, en este trabajo se propone abordar este macro proceso desde dos aristas, un componente para la parte estructural y otro para la dinámica, cada componente es concebido como una célula de gobierno [30], de esta forma el modelo queda compuesto por dos células de gobierno que interactúan entre ellas y se complementan.

Cada célula tiene por objeto facilitar la implementación de uno de los componentes de la gobernanza visualizados en el análisis de la GDS, esto es una célula se encarga de crear y gestionar el componente estructural, y la otra de crear y gestionar el componente dinámico. La representación gráfica del modelo se la visualiza en la figura 14.

El alcance planteado para este trabajo está centrado en el estudio de la gobernanza requerida para el desarrollo de software, en organizaciones públicas o privadas que tienen un área de TI tipo VSE.

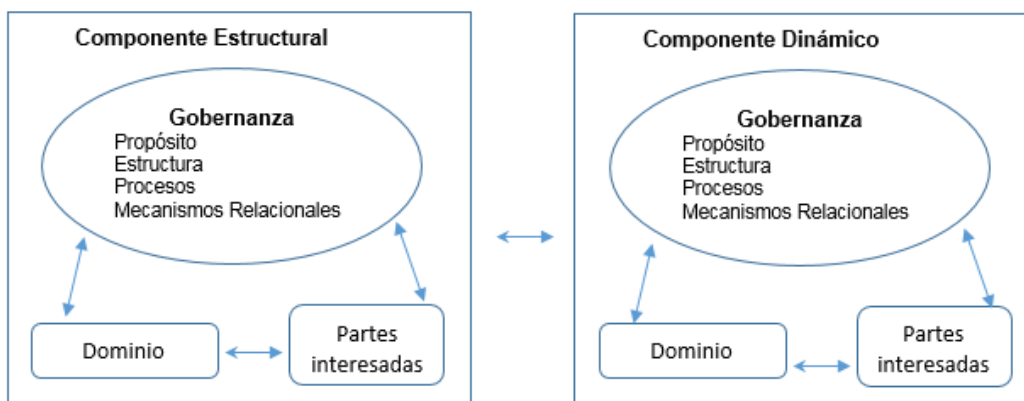


Figura 12 Modelo de Gobernanza de Desarrollo de Software para áreas de TI tipo VSE

Al modelo propuesto, se le denomina como **MGDS-VSE**, Modelo de Gobernanza para el Desarrollo de Software para áreas de TI tipo VSE (ISO 29110).

Fuentes que aportan al modelo.

Para la construcción del modelo se consideran aportes desde 4 diferentes fuentes que las mencionamos a continuación:

- El trabajo de Paul L. Bannerman, “Gobernanza del desarrollo de software: una perspectiva de meta-gestión” [30], del cual se extrae el concepto de célula de

gobernanza. Una celda de gobernanza es una estructura que se aplica a las actividades de un dominio de administración particular en este caso, la creación y la gestión de la infraestructura de gobierno para el desarrollo de software, y la creación de estructuras y procesos de control para la gestión de proyectos de desarrollo de software. Está asociada con un conjunto de partes interesadas tales como proponentes, usuarios de la aplicación, beneficiarios, y funcionarios a nivel estratégico que gestionan y administran la organización. Cada celda se interconecta con la otra para formar una infraestructura de gobierno integrada, que permite establecer un patrón o marco de gobierno que se adapte a las necesidades de la organización.

- El marco de trabajo para el gobierno y la gestión de las TI en la empresa COBIT 5, contribuye con un conjunto de recomendaciones para la creación de las estructuras de gobierno y de control, así como también con un conjunto de procesos probados y validados para: Alinear, Planificar y Organizar (APO), Gestionar el marco de gestión de TI (APO01), Gestionar la innovación (APO04), Gestionar los recursos humanos (APO07), Gestionar la calidad (APO11), Gestionar la definición de requerimientos (BAI02), Gestionar los problemas (DSS03), Asegurar la transparencia a las partes interesadas (EDM05), Supervisar, Evaluar y Valorar Rendimiento y Conformidad (MEA01), etc.
- Las metodologías ágiles para gestionar los proyectos de desarrollo de software. De SCRUM, se extrae la estructura jerárquica y los roles requeridos para la gestión del proceso de desarrollo, las estructuras o contenedores de los requerimientos, tanto a nivel del proyecto como para cada iteración, esto es la pila de requerimientos del producto (*Product Backlog*) y la pila de requerimientos de la iteración (Sprint Backlog), El incremento es la parte de producto producida en una iteración (Sprint), y tiene como característica el incorporar una o más funcionalidades completamente terminadas y operativas, en condiciones de ser entregada al cliente [20].
De Programación Extrema XP, se extrae las historias de usuarios y el Proceso de exploración o lluvia de ideas donde colabora todo el equipo partiendo de la visión del dueño del producto y las partes interesadas.
- Del marco de trabajo de Ingeniería de Software / Web. El proceso marco de Comunicación, tiene importancia crítica comunicarse y colaborar con el cliente (y con los interesados y otros participantes), se debe entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software. Para crear un modelo de análisis completo para la aplicación que se requiere desarrollar, se debe definir el ámbito total de la misma durante la actividad de formulación de la primera iteración [17], y

el proceso marco de Planeación, en el cual se establece un plan incremental para el desarrollo de la aplicación, en este, se introducen las acciones que se van a producir, las tareas técnicas que se llevarán a cabo, los riesgos a los que probablemente se tenga que exponer y mitigar, los recursos que serán necesarios, los productos de trabajo que se producen, y un cronograma de trabajo.

Descripción del Modelo MGDS-VSE.

El modelo se compone de dos células de gobernanza, cada célula tiene su propósito definido, 1) crea el componente estructural y 2) crea el componente dinámico:

Célula de Gobernanza Componente Estructural

- **Propósito.** Establecer el componente estructural de la gobernanza requerida para el proceso de desarrollo de software.
- **Estructuras.** Crear la estructura de gobierno para el proceso de desarrollo de software.
 - ✓ Administrador del proyecto, Responsable del desarrollo, Experto del negocio, Comité para la recepción, Parte interesada directa e indirecta
- **Procesos.** Procesos de COBIT 5 relacionados con Desarrollo de software.

Alinear, Planificar y Organizar (APO)

01 Definir la estructura organizativa.

- ✓ Definir los roles para las funciones internas y externas, el alcance de las funciones internas y externas, las capacidades y los derechos de decisión necesarios.
- ✓ Identificar las decisiones necesarias para el logro de los resultados empresariales, requeridos en relación con el desarrollo del software y la estrategia de TI.
- ✓ Establecer la participación de las partes interesadas que son fundamentales para la toma de decisiones (responsabilidad de dar cuenta, responsabilidad de hacer o desarrollar y quienes deben ser consultados o informados).
- ✓ Proporcionar directrices para cada estructura de gestión (incluida la gestión, los objetivos, los asistentes a la reunión, el calendario, el seguimiento, la supervisión y la supervisión), así como los insumos necesarios y los resultados esperados de las reuniones.

02 Establecer roles y responsabilidades.

- ✓ Designar a los funcionarios que ocupan que ocupan los roles o estructura jerárquicas creadas: Administrador del proyecto, Experto del negocio (funcionario que conoce el área o áreas de la organización para la cual se

desarrolla la aplicación), Responsable del desarrollo, Responsable de las pruebas. Identificar a las partes interesadas y sus registrar sus intereses, Estructurar un grupo responsable de las pruebas con elementos de las partes interesadas, Designar al Comité para la recepción del desarrollo de software.

- ✓ Definir las decisiones que la estructura está autorizada a tomar, los Niveles de autorización y derechos de decisión, establecer una guía para la toma de decisiones en el proyecto.

04 Comunicar la gestión de los objetivos y la dirección.

- ✓ Comunicar el conocimiento y comprensión de los objetivos de TI y la dirección a los interesados y usuarios apropiados en toda la empresa.

AP02 Gestionar la Estrategia

01 Considerar el entorno empresarial actual y los procesos de negocio, así como la estrategia de la empresa y los objetivos futuros.

- ✓ Alineación de TI con la estrategia de la empresa, establecer el compromiso de la dirección para tomar decisiones relacionadas con TI y gestionar los riesgos del negocio relacionados con TI.

AP08 Gestionar las Relaciones

01 Entender las expectativas del negocio.

- ✓ Entender los problemas y objetivos empresariales actuales y las expectativas de negocio para TI. Asegurar que los requerimientos son entendidos, manejados y comunicados, y su estado final es acordado y aprobado.

04 Coordinar y comunicar.

- ✓ Trabajar con las partes interesadas y coordinar la entrega final de los servicios y soluciones de TI proporcionadas al negocio.

Supervisar, Evaluar y Valorar (MEA)

MEA01 Supervisar, Evaluar y Valorar Rendimiento y Conformidad.

01 Establecer un enfoque de monitoreo.

- ✓ Colaborar con las partes interesadas para establecer y mantener un enfoque de monitoreo para definir si los objetivos, el alcance y la prestación de servicios es la esperada.

- **Mecanismos relacionales**

- ✓ Promover la participación activa de los interesados.
- ✓ Colaboración entre las principales partes interesadas.
- ✓ Diálogo estratégico y participación activa de las principales partes interesadas.
- ✓ Recompensas e incentivos de la asociación.
- ✓ Comprensión compartida de los objetivos de negocio / TI.
- ✓ Rotación de puestos de trabajo multifuncionales / IT.

Célula de Gobernanza Componente Dinámico

- **Propósito.** Establecer el componente dinámico de la gobernanza requerida para el desarrollo de software, esto es establecer los mecanismos de medición y control, para verificar que los desarrolladores de software, directores de proyectos y otras personas realicen apropiadamente sus funciones y responsabilidades.
- **Estructuras**
 - ✓ Documento de especificación de requisitos funcionales y no funcionales que debe satisfacer la aplicación.
 - ✓ Documento de especificación de requisitos funcionales y no funcionales que deben ser implementados en cada iteración.
 - ✓ Cronograma valorado de actividades para la implementación de los requerimientos previstos en la iteración.
 - ✓ Lista de los estándares a utilizar en el desarrollo.
 - ✓ Lista de las métricas para evaluar el avance y la calidad del producto (elementos implementados por iteración, características implementadas, velocidad, estándares utilizados, usabilidad).
 - ✓ Marco de trabajo SCRUM para la gestión del desarrollo
 - ✓ Roles SCRUM (Maestro SCRUM, miembro del equipo SCRUM Dueño del Producto, Comité de pruebas).
 - ✓ Estructura de la pila del producto
- **Procesos.** Procesos de COBIT 5 relacionados con Desarrollo de software.

Alinear, Planificar y Organizar (APO)

AP02 Gestionar la Estrategia

- ✓ Levantar el documento de especificación de requerimientos entre las partes interesadas y el dueño del producto, maestro SCRUM y el administrador del proyecto
- ✓ Negociar el alcance a partir del documento de especificación de requerimientos entre las partes interesadas y el dueño del producto, maestro SCRUM y el administrador del proyecto
- ✓ Estimación de tiempos para determinar la duración de la iteración y del producto
- ✓ Lista de las iteraciones estimadas en base al alcance del producto

AP11 Gestionar la Calidad

- ✓ Especificar las listas de los estándares a utilizar en el desarrollo y de las métricas que se utilizarán para evaluar la calidad del producto.
- ✓ Crear la pila del producto con todos los requisitos solicitados debidamente priorizados.

- ✓ Establecer mecanismos y técnicas para medir el avance del proyecto.

AP12 Gestionar el Riesgo

- ✓ Identificar, evaluar y reducir continuamente los riesgos relacionados con la TI dentro de los niveles de tolerancia establecidos por la administración ejecutiva de la empresa.

Construir, Adquirir e Implementar (BAI)

BAI01 Gestionar los Programas y Proyectos

- ✓ Gestionar de forma coordinada la planificación, control, ejecución, programas, proyectos, y cierre con una revisión posterior a la implementación.
- ✓ Gestionar el proceso de construcción del software utilizando SCRUM.
- ✓ Estimación de tiempos para determinar la duración de la iteración y del producto.

BAI02 Gestionar la Definición de Requisitos

- ✓ Levantar el documento de especificación de requerimientos entre las partes interesadas y el dueño del producto, maestro SCRUM y el administrador del proyecto.
- ✓ Negociar el alcance a partir del documento de especificación de requerimientos entre las partes interesadas y el dueño del producto, maestro SCRUM y el administrador del proyecto.
- ✓ Lista de las iteraciones estimadas en base al alcance del producto.

BAI04 Gestionar la Disponibilidad y la Capacidad

- ✓ Negociar el alcance a partir del documento de especificación de requerimientos entre las partes interesadas y el dueño del producto, maestro SCRUM y el administrador del proyecto.

BAI08 Gestionar el Conocimiento

- ✓ Levantar la documentación técnica necesaria para garantizar la mantenibilidad del software, manual de usuario, manual técnico, instalación.

Supervisar, Evaluar y Valorar (MEA)

MEA01 Supervisar, Evaluar y Valorar Rendimiento y Conformidad.

- ✓ Recopilar, validar y evaluar las metas del negocio, las metas de TI, los procesos y las métricas.
- ✓ Monitorear que los procesos se están ejecutando de acuerdo con las metas de cumplimiento y métricas acordadas.
- ✓ Proveer de reportes sistemáticos y oportunos.

01 Establecer un enfoque de monitoreo.

- ✓ Colaborar con las partes interesadas para establecer y mantener un enfoque de monitoreo para definir los objetivos, el alcance y el método para la solución de negocios y la prestación de servicios y la contribución a los objetivos de la empresa.

02 Establezca objetivos de rendimiento y cumplimiento.

- ✓ Trabajar con las partes interesadas para definir, revisar periódicamente, actualizar y aprobar los objetivos de desempeño y conformidad en el sistema de desempeño.

03 Recopilar y procesar datos de rendimiento y conformidad.

- ✓ Trabajar con las partes interesadas para recopilar y procesar periódicamente la información sobre el avance del desarrollo y evaluar los reportes de aceptación.

04 Recopilar y procesar datos oportunos y precisos alineados con los enfoques empresariales planteados.






- ✓ Trabajar con las partes interesadas y con la administración de la organización para recopilar y procesar periódicamente la información sobre el cumplimiento de las metas estratégicas asociadas al proyecto.

- **Mecanismos relacionales**

- ✓ Principio de la participación activa de los interesados.
- ✓ Colaboración entre las principales partes interesadas.
- ✓ Asociación recompensas e incentivos.
- ✓ Diálogo estratégico y aprendizaje compartido.
- ✓ Resolución activa de conflictos ("no evitación").
- ✓ Capacitación en negocios y TI.

Para complementar al modelo de gobernanza **MGDS-VSE** propuesto, se sugiere la aplicación de un modelo de gestión SCRUM combinado con marcos de trabajo de Ingeniería de Software o Web [26][27]. El modelo se presenta en la figura 15. En la tabla 8 se presenta las equivalencias para los íconos utilizados en el modelo.

Tabla 9 Equivalencias para los íconos del modelo de desarrollo de software

	Fase Conjunto de tareas asociadas con un objetivo
	Tarea. Conjunto de actividades que permiten realizar una tarea
	Rol. Permisos que debe disponer un usuario para realizar un trabajo.
	Producto. Recurso de entrada o salida al ejecutar una tarea o actividad.
	Guía / métrica. Recomendación para desarrollar o medir una tarea, proyecto, o producto.

	Hito. Elemento de descomposición del trabajo que es un evento importante para la tarea o proyecto
	Paso. Componente que posibilita la ejecución de una tarea
	Flujo. Dirección del flujo de una actividad, tarea, o paso

Modelo SCRUM modificado para gestionar el desarrollo de software aplicando el modelo de gobernanza de TI **MGDS-VSE**.

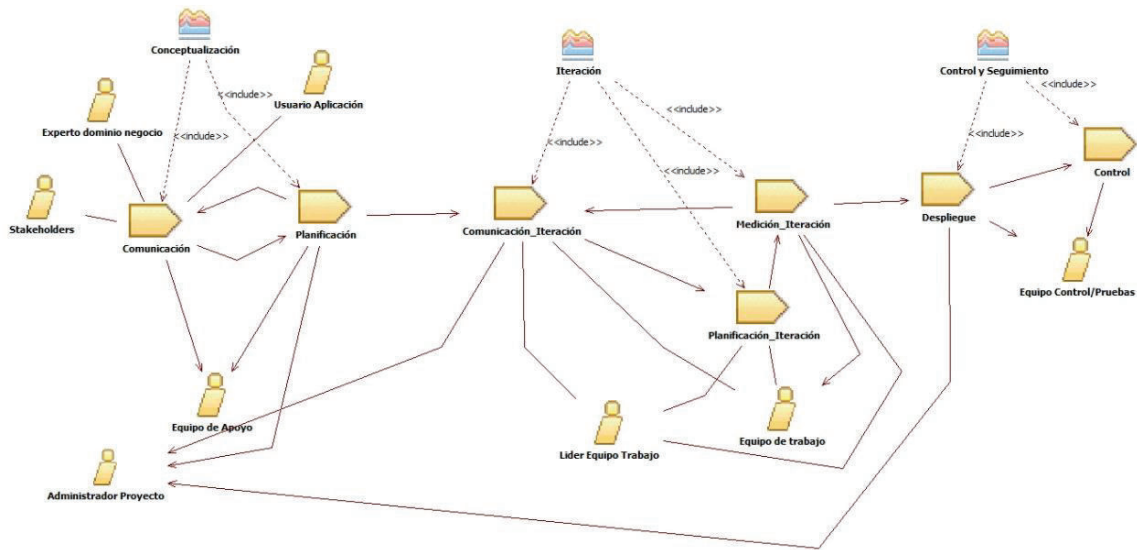


Figura 13 Modelo de comportamiento para SCRUM - MGDS-VSE.

Descripción del modelo SCRUM - MGDS- VSE-ISO 29110 propuesto para la gestión de un proyecto de desarrollo de software.

El modelo se sustenta en la ejecución de tres fases. La primera se refiere a la conceptualización inicial del proceso de desarrollo del software en el que la meta es entender el problema que se está planteando, proponer la solución en forma conjunta con el dueño del negocio, con base a un entendimiento claro de los requerimientos del problema y las métricas acordadas para la evaluación del producto. La segunda fase tiene que ver con el proceso iterativo que posibilita la construcción del software mediante incrementos o entregas parciales del software funcional. La tercera fase se enfoca en el control y seguimiento del proyecto de desarrollo de software [20], [18].

Fase de Conceptualización (Iteración de partida o *Pregame* de SCRUM).

Tarea Comunicación

Actividad

1. Entender el problema que se está planteando, proponer la solución en forma conjunta con el dueño del negocio, con base a un entendimiento claro de los requerimientos del problema y las métricas que se establecen para la evaluación del producto a entregar.
2. Consensuar, difundir y socializar los requerimientos, con los diferentes actores que están involucrados en el proyecto.
3. Determinar el tiempo que debe utilizarse para la iteración.
4. Priorización de las iteraciones, incluyendo el conjunto de requerimientos que se deben implementar para proveer de una funcionalidad operativa al final de cada iteración.

Entregables:

Formulación del problema, lista de requisitos, documento de especificación de requisitos, Documento con la priorización de las iteraciones, Lista de interesados, usuarios, expertos en el negocio, dueños de la aplicación, lista de quienes financian el proyecto, documento con el alcance de los requerimientos negociado y acordado.

Tarea Planificación**Actividad.**

1. Planificar las actividades necesarias para la implementación del proyecto. Incluir la estimación de tiempos para cada una de los requerimientos que forman parte de cada iteración y la estimación de tiempo que toma una iteración.
2. Estimar los riesgos asociados tanto para las actividades previstas.
3. Determinación de hitos para el proyecto.
4. Determinación de estándares y métricas a seguir, para validar y aprobar cada uno de los incrementos entregados.
5. Especificar las guías para de medición o determinación de la calidad del producto.

Entregables:

Cronograma de actividades para implementar los requerimientos acordados, priorización de las iteraciones, métricas acordadas para la medición del control de calidad, estimación de tiempos y riesgos, documento de negociación del alcance y métricas de medición, guías para la medición y control de calidad, definición de hitos.

Fase de Iteración (Entregas parciales de SCRUM).**Tarea Comunicación****Actividad**

1. Entender el problema que se plantea para la iteración
2. Difusión de los requerimientos, las métricas establecidas para la evaluación del incremento, guías para medición, cronograma de actividades de la iteración.
3. Ajustar requerimientos de ser necesario, difundir las métricas de control de calidad.
4. Formulación del problema asociado a la iteración.

Entregables:

Documento de especificación de requisitos de la iteración negociado y acordado, lista de usuarios, y expertos en el negocio que participaran en las pruebas de aceptación, métricas para la evaluación, guías para la evaluación.

Tarea Planificación Iteración.**Actividad**

1. Planificación de actividades y tiempos para las actividades previstas para la iteración.
2. Estimar los riesgos asociados con las actividades previstas para la iteración.
3. Determinación de hitos para la iteración.

Entregables:

Cronograma de actividades con los requerimientos detallados en la especificación de requisitos, estimación de riesgos, documento de negociación del alcance.

Tarea Implementación y Medición**Actividad**

1. Implementar el incremento planificado.
2. Evaluar el incremento entregado, en función de las guías de medición de control de calidad.
3. Elaborar el reporte de resultados.

Entregables:

Reporte de resultados

Fase de Control y Seguimiento.**Tarea Despliegue****Actividad**

1. Pruebas de integración.
2. Revisión de métricas de control de calidad de la iteración, documentación de soporte del incremento.
3. Elaborar: reporte de aceptación de los usuarios, reporte de cumplimiento de las métricas de calidad, reporte de pruebas de integración.

Entregables:

Reporte de pruebas de integración, reporte de aceptación y cumplimiento de métricas.

Tarea Seguimiento y control del proyecto**Actividad**

1. Evaluar el progreso alcanzado en relación con el plan del proyecto.

2. Realizar revisiones técnicas formales e informales al proyecto.
3. Tomar acciones necesarias para apegarse a la programación de actividades cuando no se cumplan los tiempos establecidos para los incrementos, no se cumplan las métricas establecidas.

Entregables:

Informe de control y seguimiento.

Informe de problemas presentados y mitigación de los mismos.

En la figura 16 se presenta el modelo estructural del modelo SCRUM - MGDS-VSE propuesto para la gestión de un proyecto de desarrollo de software, en este se puede visualizar los productos que se generan y sirven de insumos entre las diferentes fases y sus tareas asociadas, así como también el uso de las métricas y las guías acordadas para el proyecto.

En los anexos 4 y 5 se presentan los esquemas estructurales para las tareas de comunicación y planificación de la fase inicial.

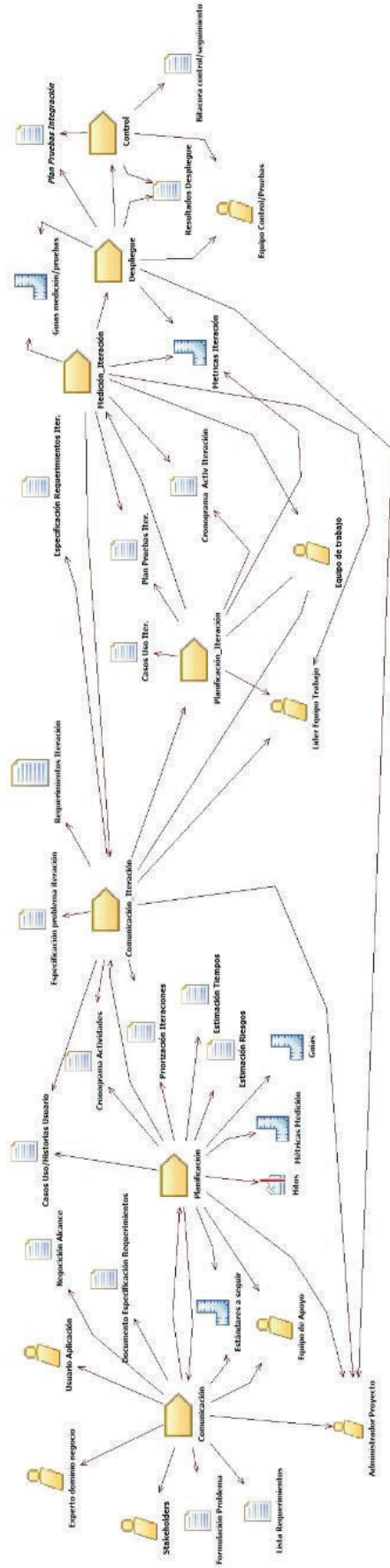


Figura 14 Esquema estructural del modelo SCRUM - MGDS-VSE.

Para completar el proceso de gestión del desarrollo de software, es necesario implementar un conjunto de actividades complementarias, que tienen que ver con el proceso control y seguimiento, la gestión y administración del riesgo, el aseguramiento de la calidad. Estas actividades son independientes de las actividades de desarrollo del software y se deben de realizar durante todo el proceso de desarrollo del software. En la tabla 9 se presenta un resumen de estas actividades.

Tabla 10. Actividades Complementarias del proceso general de Ingeniería de software / web [17], [31].

Actividad	Descripción
1) Seguimiento y control del proyecto.	Permite al equipo evaluar el progreso, comparándolo con el plan del proyecto y tomar cualquier acción necesaria para apearse a la programación de actividades.
2) Administración del riesgo.	Evalúa los riesgos que puedan afectar el resultado del proyecto o la calidad del producto.
3) Aseguramiento de la calidad.	Define y ejecuta las actividades requeridas para garantizar la calidad del software.
4) Medición.	Define y reúne mediciones del proceso, proyecto y producto para ayudar al equipo a entregar software que satisfaga las necesidades de los participantes; puede usarse junto con todas las demás actividades.
5) Administración de la configuración.	Administra los efectos del cambio a lo largo del proceso del software.
6) Administración de la reutilización	Define criterios para volver a usar el producto del trabajo (incluso los componentes del software) y establece mecanismos para obtener componentes reutilizables.
7) Preparación y producción del producto del trabajo	Agrupar las actividades requeridas para crear productos del trabajo, tales como modelos, documentos, registros, formatos y listas.
8) Revisiones técnicas	Evalúa los productos del trabajo de la ingeniería de software a fin de descubrir y eliminar errores antes de que se propaguen a la siguiente actividad.

3.3 Resultados de la aplicación del modelo MGDS-VSE al caso estudio.

A partir de un convenio de cooperación existente entre Universidad Politécnica Salesiana (UPS) y el Ministerio de Salud Pública, se acuerda colaborar en un proyecto de construcción de una aplicación web para el Hospital de Especialidades Eugenio Espejo, en el cual participan tres integrantes de la UPS, un investigador y dos ayudantes de investigación, el investigador es designado como líder del proyecto y los ayudantes de investigación como integrantes del equipo de desarrollo del producto de software.

3.3.1 Aplicación del modelo de gobernanza de software al caso estudio.

Para la aplicación del modelo de gobernanza, se consensó con la parte interesada en el desarrollo de la aplicación, en este caso El Área de Docencia e Investigación del

HEE, en la necesidad de aplicar un modelo de gobierno para garantizar que el proyecto de cooperación genere los mejores resultados posibles y que se garantice que los objetivos estratégicos del HEE, estén en concordancia con los objetivos planteados para la construcción de software a ser construido como producto del proyecto de colaboración entre las partes (Academia - Sector Público).

Los resultados específicos de la aplicación del modelo al proyecto piloto son los siguientes:

- 1) Estructura de gobierno claramente definida, con roles y tareas específicas claras.
 - Administrador del proyecto (Director del Área de Docencia e Investigación HEE)
 - Líder del proyecto (investigador UPS)
 - Experto en el negocio (Funcionario de Especialidades Clínicas o Quirúrgicas)
 - Dueño del Producto (Funcionario designado que conoce de la gestión de Quirófanos)
 - Maestro SCRUM (investigador UPS)
 - Integrante de equipo SCRUM (ayudante investigador UPS)
 - Comité de pruebas y recepción. 1 integrante por área: Docencia e Investigación, TICS, Especialidades clínicas y/o Quirúrgicas
 - Parte interesada directa e indirecta: Docencia e Investigación, Especialidades clínicas y/o Quirúrgicas, Hospital del Día, Estadística, TICS
- 2) Procesos asociados a la gestión de la estructura de gobierno.

Por norma legal todo proyecto que tenga el aval del Área de Docencia, debe ser administrado por el Administrador del área.

- Se establece que el Administrador del Proyecto es el responsable de los siguientes procesos:
 - Definir la estructura organizativa.
 - Designa a los funcionarios que colaborarán en el proyecto, de acuerdo a sus destrezas o conocimientos relacionados con la gestión de quirófanos.
 - Establecer roles y responsabilidades.
 - Establece que las áreas de Docencia e Investigación, Especialidades clínicas y/o Quirúrgicas, Hospital del Día y estadística provean de toda la información requerida para la gestión de los quirófanos.
 - Establece que las responsabilidades sobre las funcionalidades a ser implementadas provengan de un acuerdo entre las partes interesadas, cuando exista un conflicto el responsable de desarrollo tomará la decisión final.

- Se establece que el Dueño del producto es el responsable del desarrollo y de cumplir con las siguientes tareas.
 - Gestionar la plataforma de desarrollo. Con el asesoramiento de un funcionario de TICs, acordaran con el líder del proyecto UPS, el alcance, los requerimientos a ser implementados, la plataforma de desarrollo y las métricas de medición del avance y gestión de la calidad.
 - Gestionar el cambio. Todo requerimiento nuevo o ajuste del mismo, debe ser canalizado a través del responsable del desarrollo.
 - Comunicar la gestión de los objetivos. Comunicar el conocimiento y comprensión de los objetivos, alcance y los requerimientos acordados para el desarrollo a la dirección, y a las áreas interesadas y usuarios.
 - Gestionar la capacitación. Capacitar a los responsables de las pruebas, para que utilicen las métricas acordadas y ejecuten las pruebas utilizando procesos estandarizados.
 - Trabajar con las partes interesadas y coordinar la entrega final de la aplicación.
- 3) Estructura para la gestión de la gobernanza clara y específica
- El formato para levantar los requerimientos funcionales y no funcionales que debe cumplir la aplicación, es el IEEE 830.
 - Para controlar la calidad del producto, se establece que se utilizará la metodología UWE, misma que posibilita gestionar todos los pasos del ciclo de vida del desarrollo de la aplicación web y su documentación técnica.
- 4) Procesos asociados a la gestión del control del desarrollo del software.
- Capacitación y provisión de guías y métricas para las revisiones parciales y para las de integración.
 - Capacitación y formato para informes de revisiones parciales en función de los requerimientos acordados para la iteración.
 - Guías para la revisión y validación de la documentación técnica (manuales de (usuario, instalación y técnico)

Como producto de este proyecto de colaboración entre las dos instituciones se generó la aplicación web para administrar Partes Operatorios, denominada AppPartesOperatorios.

Los resultados obtenidos fueron satisfactorios.

- Se estableció un ambiente de estrecha colaboración entre las partes interesadas, los usuarios, y el grupo SCRUM.

- Los procesos de validación de los resultados, en su mayoría fueron éxitos, y la retroalimentación de estos procesos hacia el grupo de desarrollo fueron beneficiosos para lograr un producto final de calidad.
- Se generó la documentación técnica requerida para el producto de acuerdo con el uso de las buenas prácticas recomendadas por la aplicación de la ingeniería de software o web.
- Se firmó un acta de entrega recepción del producto con la conformidad de las partes.

Para el efecto:

- Se estableció un ambiente de estrecha colaboración entre las partes interesadas, los usuarios, y el grupo SCRUM.
- Los procesos de validación de los resultados, en su mayoría fueron éxitos, y la retroalimentación de estos procesos hacia el grupo de desarrollo fueron beneficiosos para lograr un producto final de calidad.
- Se generó la documentación técnica requerida para el producto de acuerdo con el uso de las buenas prácticas recomendadas por la aplicación de la ingeniería de software o web.
- Se firmó un acta de entrega recepción del producto con la conformidad de las partes.

La aplicación del modelo demostró ser de mucha utilidad, debido a que se presentaron perturbaciones en el proceso de gestión del proyecto.

La perturbación se produjo por el cambio de director de TICS. Este cambio introdujo un ambiente de inestabilidad entre los participantes en el proyecto piloto, puesto que el nuevo líder de TICS desconoció algunos elementos del acuerdo, así como también las designaciones de algunos funcionarios. La oportuna intervención del administrador del proyecto en base a sus funciones y atribuciones reunió al equipo del proyecto junto con el director de TICS, al final de la reunión todo quedó claro y se continuó con el proyecto con los términos acordados.

En el anexo 6 se presenta un cuadro en el que se especifica la aplicación del modelo al proyecto piloto.

3.3.2 Recomendaciones para la evaluación del modelo para la Gestión la de Gobernanza del proceso de desarrollo de software.

Para proceder a evaluar el modelo se define los siguientes criterios, los mismos que buscan visualizar si el modelo cumple con los objetivos y expectativas previstas para su construcción.

1. Posibilita gestionar la gobernanza al proceso de desarrollo de software.
2. Elimina prácticas deficientes como falta de métricas para la medición, no utilizar estándares para desarrollo del software, no se cumplen los plazos establecidos, no se implementan todos los requerimientos acordados, existe alta variabilidad en la ejecución de sus procesos de desarrollo de software, no se dispone de planes de pruebas para validar la aplicación, no se gestiona los riesgos inherentes al desarrollo de la aplicación de manera formal, etc.
3. Recomienda la aplicación de las mejores prácticas de ingeniería de software y es repetible.
4. Su aplicación facilita una base formal para el análisis, diseño, construcción y pruebas.
5. Gestiona el conocimiento, incorporando artefactos, procesos y herramientas para almacenar el conocimiento de la organización.

3.3.3 Evaluación del modelo de gobernanza de software.

Para recopilar la información de la aplicación del modelo en el proyecto de desarrollo de software en la entidad piloto HCAM, se implementó dos encuestas que constan de 10 preguntas cada una. La primera está orientada verificar si la aplicación del modelo aportó de manera positiva a la gestión del proyecto, vista desde la perspectiva del administrador del proyecto y la segunda está orientada verificar si la aplicación del modelo aportó de manera positiva al proceso de desarrollo del producto software, vista desde la perspectiva del equipo de desarrollo que se encargó de su construcción. El detalle de las encuestas se incorpora en el anexo 1.

Tomando como base la caracterización de las valoraciones a cada una de las preguntas de la encuesta, los valores 4 y 5, certifican que el aporte del modelo es positivo (4 satisfactoriamente y 5 totalmente), 3 medianamente y 2 y 1, indican que la aporte es deficiente o no existe.

Los resultados obtenidos se presentan en los siguientes cuadros de resumen, la tabla 10, presenta las valoraciones efectuadas por el administrador del proyecto y la tabla 11, las valoraciones efectuadas por el equipo de desarrollo.

Tabla 11 Aporte para la gestión

Aporte del modelo para la gestión de proyecto, vista por la Dirección del proyecto.

Pregunta	#	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	V Prom
Valoración	1	4	5	4	4	4	4	4	5	4	5	4,3

Las preguntas efectuadas para valorar el aporte del modelo para la gestión de la gobernanza en el proceso de desarrollo de software, se orientan a validar si la definición de las estructuras organizacionales, roles, procesos y actividades, efectivamente aportan en la gestión del proyecto.

Los resultados de la encuesta efectuada al administrador del proyecto, reporta una percepción positiva del aporte del modelo, pues el promedio de las valoraciones es de 4.3 lo que refleja que la percepción es más que satisfactoria (4 satisfactoria).

Tabla 12 Aporte para construcción del producto software

Aporte del modelo para la construcción del producto software, vista por el equipo de desarrollo.

Pregunta	#	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	V Prom	
Valoración	1	4	5	5	4	4	5	5	4	5	5	4,6	I E D
Valoración	2	5	5	5	4	4	5	5	4	5	5	4,7	M S
Promedio		4,5	5	5	4	4	5	5	4	5	5	4,65	

I E D. Integrante del equipo de desarrollo

M S. Maestro SCRUM

Las preguntas efectuadas para valorar el aporte del modelo al proceso de desarrollo del producto software, se orientan a validar si la definición de las estructuras organizacionales, roles, procesos y actividades, aportan para la construcción del software, facilitando la comunicación entre los actores involucrados en el proceso de desarrollo de software, el levantamiento de los requerimientos, la gestión de la calidad, el cumplimiento de los plazos establecidos, y la gestión de los riesgos.

Los resultados de la encuesta efectuada al equipo de desarrollo del proyecto, reportan una percepción positiva del aporte del modelo, pues el promedio de las valoraciones es de 4.65, lo que refleja que la percepción es más que satisfactoria (4 satisfactoria).

3.3.4 Resultados de la evaluación del modelo de gobernanza de software.

En base a la valoración promedio obtenida para cada una de las preguntas de las encuestas efectuadas para verificar si el modelo constituye un aporte positivo para construcción del software, y para la gestión del proyecto. Los resultados obtenidos son más que satisfactorios, 4.65 y 4.3, lo que avala que el aporte del modelo es positivo para la gestión de la gobernanza en el proceso de desarrollo de software.

1. La aplicación del modelo al caso de estudio, permitió definir la estructura organizacional requerida para la aplicación de la gobernanza y definir los diferentes procesos y estructuras de control para la gestión del desarrollo de software con la calidad necesaria, debido a que se aplicaron las prácticas recomendadas por la aplicación de la ingeniería de software. Adicionalmente en base a la aplicación estricta de la jerarquía definida para la estructura organizacional, se pudo eliminar las perturbaciones que se presentaron.
2. El modelo provee de un conjunto de recomendaciones para el desarrollo de software, tales como la aplicación de estándares de desarrollo, métricas para la medición de los productos y avance del proyecto, la utilización de planes de pruebas estandarizados, y además posibilita la generación de la documentación del proceso de desarrollo de software.
3. El modelo recomienda la aplicación de una metodología ágil para el desarrollo de software, propone una metodología basada fundamentalmente en SCRUM, en la que se incorpora elementos recomendados por los marcos de desarrollo de ingeniería de software, en las que se recomiendan la utilización de estándares para el desarrollo del software, adopción de métricas para la medición, planificación de pruebas estandarizadas en base a las métricas adoptadas, implementación de una bitácora de control y seguimiento, etc.
4. La aplicación del modelo genera productos tales como la pila de requerimientos a ser implementados en el proyecto, los requerimientos que deben ser implementados por cada iteración o incremento, los cronogramas de trabajo, recomienda la aplicación de métricas y estándares de desarrollo, recomienda generar planes de pruebas basados en estándares en función de las métricas acordadas. En base a estos elementos se puede afirmar que facilita una base formal para el análisis, diseño, construcción y pruebas.
5. La gestión del conocimiento, esta soportada por la documentación de los procesos recomendados por el modelo, que al ser almacenada genera una base de conocimientos para la organización. El aprovechamiento de esta base de

conocimientos posibilitará la implementación de mejoras al proceso de construcción de software, posibilitando la retroalimentación del modelo propuesto.

3.4 Discusión

Una de las razones para argumentar la necesidad de plantear el desarrollo de un modelo que facilite la implementación de la Gobernanza para el desarrollo de software era el desconocimiento que existe en las empresas acerca del tema de la Gobernanza para el proceso del desarrollo de software. Los resultados obtenidos a través de las encuestas confirman esa hipótesis. Los resultados muestran que un 18% de las empresas no gestiona la Gobernanza para el desarrollo de software o lo hacen de forma deficiente y un 45% lo gestiona medianamente.

El porcentaje de empresas que gestionan apropiadamente la Gobernanza de TI y la de DS es muy baja, solo el 27%, esto pone de manifiesto la falta de conocimiento acerca de la importancia de la gestión de Gobernanza para los procesos de desarrollo de software.

En el país no existen registros de investigaciones anteriores sobre la Gestión de la Gobernanza para el desarrollo de software en las empresas que desarrollan software, por esta razón no es posible comparar los resultados de esta investigación con otros periodos anteriores. Por esta razón se recomienda que se realicen futuras investigaciones sobre el tema, en ellas se podrá visualizar como ha cambiado la percepción actual que tienen las empresas que desarrollan software sobre la Gobernanza para el desarrollo de software.

En base al análisis del marco teórico y a la determinación del estado del arte sobre el tema de la gestión de gobernanza para el desarrollo de software, es posible anticipar el dilema que tienen las empresas cuya área de TI es de tipo VSE, para aplicar la gobernanza a sus procesos de desarrollo de software. Una opción será aplicar marcos de gobierno de TI como COBIT 5, o la norma ISO/IEC 38500, los mismos que requieren que la empresa disponga de una cantidad considerable de recursos, lo cual implica altos costos, que pocas organizaciones los pueden solventar, una segunda opción será utilizar la norma ISO 29110 VSE, la misma que constituye una guía para la gestión del desarrollo de software en organizaciones tipo VSE y finalmente tercera opción será utilizar modelos de gobernanza que se han diseñado para otros entornos de trabajo, como por ejemplo, el Global Teaming Model for Global Software Development Governance GTM, o el modelo para Ecosistemas de Desarrollo, que son modelos orientados para trabajar en entornos distribuidos, los

mismos que deberían ser adaptados ya que no se ajustan a un entorno de trabajo de una VSE.

El modelo propuesto para la gestión de la Gobernanza del proceso de desarrollo de software, es un aporte para las empresas que tienen áreas de TI tipo VSE, puesto que provee de un conjunto de recomendaciones para crear la estructura de gobierno con sus respectivas cadenas de responsabilidad, autoridad y comunicación, que visualizan el nivel de involucramiento de los diferentes roles, así como también para la creación de las estructuras, procesos y mecanismos de control requeridos para medir y controlar que las personas involucradas en proyecto de desarrollo de software cumplan a cabalidad sus funciones.

4 CONCLUSIONES

La propuesta metodológica permite solventar el requerimiento que tienen las empresas que desarrollan software y disponen de un área de TI tipo VSE de contar con un modelo ajustado a sus necesidades, que les posibilite aplicar la Gobernanza a los procesos de desarrollo de software.

El presente trabajo busca aportar con una propuesta para solucionar la deficiencia en la aplicación de la gobernanza en el desarrollo de software, tomando como línea base el marco de referencia para la gobernanza propuesto por COBIT 5, la propuesta se orienta a organizaciones que cuentan con áreas de TI con equipos de trabajo tipo VSE.

La aplicación del modelo posibilita desarrollar software que se ajusta y alinea con el modelo y la estrategia del negocio y que además cumpla con los parámetros y estándares requeridos para el desarrollo de software de calidad.

Las empresas públicas y privadas requieren capacitar a sus ejecutivos y funcionarios de TI en la Gobernanza de Tecnologías de la Información y de Desarrollo de Software, para solventar el desconocimiento existente sobre este tema y que se ve reflejado en los resultados de las encuestas.

Recomendaciones.

En relación con la investigación realizada, sería recomendable hacerla al menos una vez cada dos años, puesto que la gobernanza para el desarrollo de software siempre está vigente, debido a que es un área en constante cambio y desarrollo y es fundamental para el éxito de la estrategia empresarial.

Proponer talleres de capacitación sobre la temática de la gobernanza en las empresas públicas y privadas de nuestro país, a través de la AESOFT, Universidades, Colegio de Ingenieros en Informática, sistemas y Computación.

REFERENCIAS BIBLIOGRAFICAS

- [1] ICG LATAM, "Como aporta COBIT 5 y gobernanza de TI a la gobernanza empresarial," p. 39, 2014.
- [2] ISO/IEC, "ISO / IEC TR -1 Software engineering — Lifecycle profiles for Very Small Entities (VSEs)," vol. 2011, 2011.
- [3] ISO/IEC, "ISO/IEC TR 29110 5-1-2 Software engineering — Lifecycle profiles for Very Small Entities (VSEs)," *Security*, vol. 2000, p. 54, 2011.
- [4] C. Y. Laporte, "ISO / IEC 29110 : Normas y guías de ingeniería de software y sistemas para entidades muy pequeñas," no. March, p. 56, 2016.
- [5] C. Y. Laporte, "Introducción a la norma ISO / IEC 29110 estándares para el desarrollo de software en PYMES," 2016, p. 4.
- [6] S. Chulani, C. Williams, and A. Yaeli, "Software development governance and its concerns," *Proc. 1st Int. Work. Softw. Dev. Gov.*, pp. 3–6, 2008.
- [7] J. Noll, S. Beecham, I. Richardson, and C. N. Canna, "A Global Teaming Model for Global Software Development Governance: A Case Study," *2016 IEEE 11th Int. Conf. Glob. Softw. Eng.*, pp. 179–188, 2016.
- [8] A. F. Sommer, I. Dukovska-popovska, and K. Steger-jensen, "Agile Product Development Governance – On Governing the Emerging Scrum / Stage-Gate Hybrids," *Adv. Prod. Manag. Syst. Innov. Knowledge-Based Prod. Manag. a Glob. World*, p. 8, 2014.
- [9] A. Baars and S. Jansen, "A framework for software ecosystem governance," *Lect. Notes Bus. Inf. Process.*, vol. 114 LNBIP, pp. 168–180, 2012.
- [10] H. Garbarino, "Marco de Gobernanza de TI para empresas PyMEs - SMEsITGF," p. 227, 2014.
- [11] J. Arnoletto, "Impacto de la tecnología en la transformación del mundo.," 2007.
- [12] FAO, "Plantilla de buenas prácticas," p. 5, 2015.
- [13] ISO/IEC, "NTE INEN-ISO/IEC 38500 Gobierno Corporativo de la Tecnología de la Información," p. 21, 2014.
- [14] J. Saavedra and A. Torres Olaya, "Modelo de Gobierno de TI como apoyo al proceso de transformación digital en empresas de la industria editorial," 2012.
- [15] ISACA, *Un Marco de Negocio para el Gobierno y la Gestión de las TI de la Empresa - COBIT 5*. 2012.
- [16] R. S. Pressman, *Ingeniería de software un enfoque práctico.*, Séptima. 2010.
- [17] R. S. Pressman and D. Lowe, *Web Engineering- Practitioner's Approach*, Primera. 2009.
- [18] M. Skarin and H. Kniberg, *Kanban y Scrum – obteniendo lo mejor de ambos*. 2010.
- [19] H. Kniberg, *Scrum and XP desde las trincheras*. 2007.
- [20] P. Juan and R. Claudia, *Scrum Manager Gestión de proyectos*. 2011.
- [21] Y. C. Gil, "Evaluación de Ecosistemas de Desarrollo de Software desde la perspectiva de COBIT," *Des. Eng.*, p. 10, 2016.
- [22] C. Manteli, B. Van Den Hooff, and H. Van Vliet, "The effect of governance on global software development: An empirical research in transactive memory systems," *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1309–1321, 2014.
- [23] S. Ali and S. U. Khan, "Critical Success Factors for Software Outsourcing Partnership (SOP): A Systematic Literature Review," *2014 IEEE 9th Int. Conf. Glob. Softw. Eng.*, pp. 153–162, 2014.
- [24] O. M. Gabriel, "Análisis de grandes ecosistemas de software abierto," p. 78, 2013.
- [25] P. Weill and J. W. Ross, "How Top Performers Manage IT Decisions Rights for Superior Results," *IT Gov.*, no. Harvard Business School Press Boston, Massachusetts, pp. 1–10, 2004.
- [26] M. del P. Hernández Sampieri, Roberto ; Fernández Collado, Carlos ; Baptista Lucio, *Metodología de la investigación*, Quinta edi. 2010, 2010.

- [27] R. M. T. Odar, "El alcance de las investigaciones jurídicas," p. 22, 2015.
- [28] D. Frias-Navarro, "Alfa de Cronbach y consistencia interna de los ítems de un instrumento de medida," *Univ. Val.*, no. 2003, pp. 1–3, 2006.
- [29] S. F. Soler Cárdenas and L. Soler Pons, "Usos del coeficiente alfa de Cronbach en el análisis de instrumentos escritos," *Rev. Médica Electrónica*, vol. 34, no. 1, pp. 01–06, 2012.
- [30] P. L. Bannerman, "Software development governance: A meta-management perspective," *Proc. 2009 ICSE Work. Softw. Dev. Governance, SDG 2009*, no. October, pp. 3–8, 2009.
- [31] R. S. Pressman, *Ingeniería del software*, Séptima. México: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V, 2010.

ANEXOS

ANEXO 1

Cuestionario de la Encuesta

<https://www.surveio.com/survey/d/X7C1W9S4D3M2C2Y7E?preview=1>

Cuestionario

1. ¿Cuál es su cargo en la empresa?
 1. Gerente o Director General
 2. Director de Sistemas de Información/Informática (CIO/CTO)
 3. Director Financiero (CFO)
 4. Gerente de Recursos Humanos
 5. Gerente de Producción
 6. Personal de TI
 7. Otro
2. ¿Qué formación académica posee? (* Marcar el nivel más alto)
 - 1 Posgrado
 - 2 Título universitario
 - 3 Técnico
 - 4 Secundaria/Bachiller
3. ¿Cuántos funcionarios posee el área de TI de su organización?
 1. entre 1 y 4
 2. entre 5 y 25
 3. más de 25
4. ¿En qué tipo de organización se ubica su empresa?
 1. Privada
 2. Pública
5. El área de TI, conoce y entiende el negocio de la organización
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
6. El área de TI, alinea sus objetivos estratégicos con los del negocio u organización
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
7. El área de TI, alinea y adapta sus objetivos estratégicos de manera conjunta con las otras áreas de la organización
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
8. La organización dispone de una adecuada priorización de inversiones para TI.
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente

9. La organización conoce lo que el área de TI, puede aportar a los objetivos estratégicos de la misma
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
10. Están alineadas las capacidades operativas (infraestructura, servicios) de TI con la estrategia del negocio de la organización
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
11. La organización mejora su competitividad con el soporte de TI
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
12. La organización es más eficiente en el uso de sus recursos financieros, tecnológicos y humanos con el soporte de TI
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
13. La percepción del usuario en relación con los servicios que TI provee a la organización es positiva
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
14. ¿Existen procesos de control para supervisar, evaluar y valorar el desempeño y la conformidad de los servicios que provee TI a la organización?
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
15. ¿Existe una estructura de gobierno bien definida, con roles específicos para desempeñar cada función requerida por la gobernanza de TI?
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
16. ¿Los roles definidos cuentan con las atribuciones necesarias para la toma de decisiones que requiere el desempeño de su función?
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente

17. Cuando se desarrollan nuevas aplicaciones de software para la organización, las prioridades son las demandas del negocio
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
18. Cuando se desarrollan nuevas aplicaciones de TI, para la organización se toman en cuenta las demandas de los usuarios
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
19. ¿Existen procesos y estructuras de control para la gestión del riesgo en los proyectos de desarrollo de software?
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente
20. ¿Existen procesos de control para supervisar, evaluar y valorar el desempeño y la conformidad de los productos en los proyectos de desarrollo de software?
 - 1 No
 - 2 Poco
 - 3 Medianamente
 - 4 Satisfactoriamente
 - 5 Totalmente

Encuesta para la validación de la aplicación del modelo: Preparada para el equipo de desarrollo.		
1	Facilita la comunicación entre los integrantes del equipo de desarrollo y el dueño del producto	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
2	Facilita la comunicación entre el dueño del producto y las partes interesadas	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
3	Facilita la implementen de los requerimientos acordados entre las partes	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
4	Facilita que se cumplan con los plazos establecidos para el desarrollo de la aplicación.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
5	Facilita la gestión de los riesgos inherentes al desarrollo de la aplicación de manera formal.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
6	Recomienda la utilización de estándares para los planes de pruebas y posibilita mejorar el proceso de validación de la aplicación.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
7	Elimina la variabilidad en la ejecución de los procesos de desarrollo de software.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
8	Elimina prácticas deficientes como falta de métricas para la medición, no utilizar estándares para desarrollo del software.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente

9	Ayuda en la integración y comunicación entre las partes involucradas en el desarrollo del proyecto.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
10	Ayuda en la solución de los conflictos entre las partes involucradas en el desarrollo del proyecto.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente

Involucrados en el proyecto: dueño del producto, las partes interesadas, equipo de desarrollo, equipo de pruebas, usuarios finales.

	Encuesta para la validación de la aplicación del modelo: Preparada para el equipo el administrador del proyecto	
1	Los procesos recomendados por el modelo para definir la estructura organizativa para el proyecto de desarrollo facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
2	La estructura jerárquica y los roles recomendados por el modelo para gestionar el proyecto de desarrollo facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
3	Las estructuras y procesos recomendados por el modelo para gestionar el avance del proyecto facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
4	Las estructuras y procesos recomendados por el modelo para gestionar el cumplimiento de los requerimientos del producto software facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
5	Las estructuras y procesos recomendados por el modelo para gestionar el control de la calidad del producto software facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
6	Las actividades recomendadas para la fase de comunicación para la obtención de los requerimientos del sistema facilitan esta tarea.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
7	Las actividades recomendadas para la fase de comunicación posibilitan llegar a un consenso sobre los requerimientos levantados.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
8	La aplicación del modelo, promueve, incentiva y asegura la participación de los diferentes elementos involucrados en el proyecto.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente

9	La percepción de las partes interesadas con relación a la aplicación del modelo al proyecto de desarrollo de software es positiva.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente
10	La aplicación del modelo, facilita la gestión del proyecto de construcción del producto software.	1 No 2 Poco 3 Medianamente 4 Satisfactoriamente 5 Totalmente

ANEXO 3

Resultados de las valoraciones caracterizadas para las preguntas de la encuesta /Sector Privado

Tabla 14 Resultados de las valoraciones caracterizadas para las preguntas de la encuesta /Sector Privado

p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20
2	1	3	4	4	4	4	5	5	4	4	5	4	4	4	4	4	5
2	1	3	3	3	3	3	3	4	2	3	4	4	2	2	3	2	2
2	1	4	4	4	4	4	3	4	4	5	5	5	5	4	4	4	3
2	1	5	4	4	3	3	3	3	3	3	4	4	4	4	4	3	3
2	1	4	5	4	5	5	4	5	4	4	5	4	4	5	4	4	4
2	1	5	4	3	4	3	4	5	4	3	5	4	5	3	3	3	3
2	1	3	3	3	3	3	4	3	3	3	3	3	3	2	3	3	3
2	1	5	5	5	5	5	5	4	3	4	3	5	5	5	5	5	5
2	1	4	4	4	4	4	4	4	3	3	4	5	5	4	4	4	3
2	1	3	3	3	3	3	2	3	3	2	3	3	4	3	3	3	3
2	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	1	3	3	3	3	3	4	4	3	3	4	4	4	2	3	3	2
2	1	5	5	4	5	5	5	3	2	4	5	5	4	4	4	4	5
2	1	4	4	4	4	4	3	3	3	4	3	3	3	5	5	4	3
2	1	3	5	4	4	4	4	3	2	3	3	4	4	4	4	2	2
2	1	4	4	2	3	3	3	3	2	3	3	3	4	3	3	2	2
2	1	5	4	3	4	4	4	3	4	4	2	3	3	4	4	4	4
2	1	3	3	3	2	3	3	3	2	2	2	3	3	3	2	3	3
# Enc	18																
Promedio		3,8889	4	3,5556	3,7778	3,7778	3,7778	3,6667	3,0556	3,3889	3,7222	3,9444	3,8889	3,6111	3,6667	3,3889	3,2778
DesvProm		0,6914	0,5556	0,6049	0,6296	0,6049	0,716	0,6667	0,6296	0,6543	0,8642	0,6296	0,6049	0,8086	0,6296	0,7222	0,8148
Moda		3	4	4	4	4	4	3	3	3	3	4	4	4	4	4	3
1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	0	1	1	0	1	0	5	2	2	0	1	3	1	3	4
3		7	5	7	5	7	6	9	7	8	6	6	4	4	6	6	8
4		6	8	9	9	8	7	6	6	7	5	7	9	8	9	8	3
5		5	5	1	3	3	4	3	0	1	5	5	4	3	2	1	3
Suma		18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18

3,64930556
0,67669753

ANEXO 4

Detalle del esquema estructural del modelo para tarea de COMUNICACIÓN.

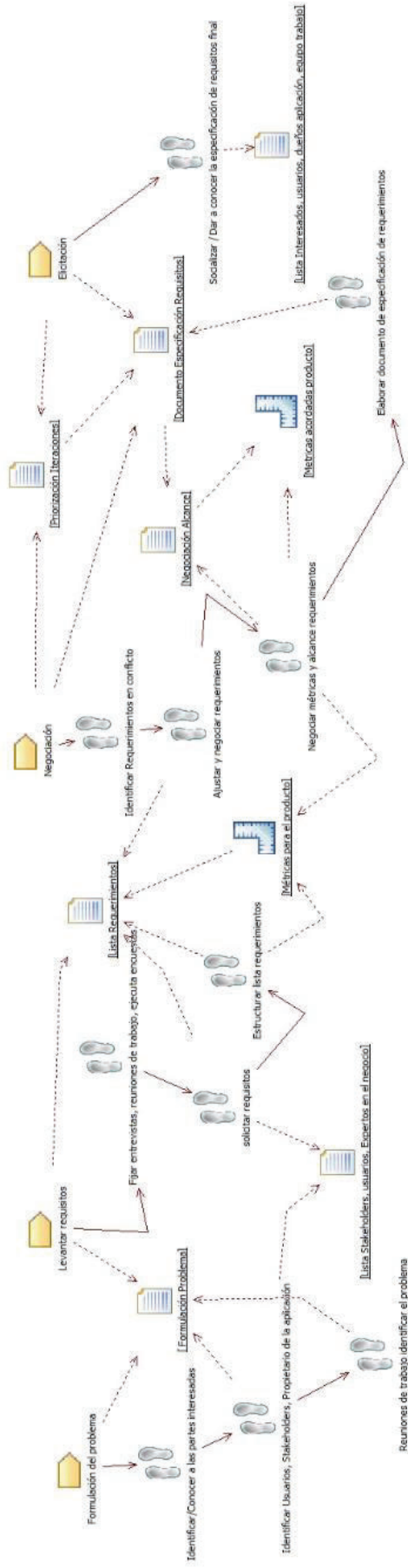


Figura 15 Detalle del modelo para la tarea de Comunicación

ANEXO 5

Detalle del esquema estructural del modelo para tarea de PLANIFICACIÓN

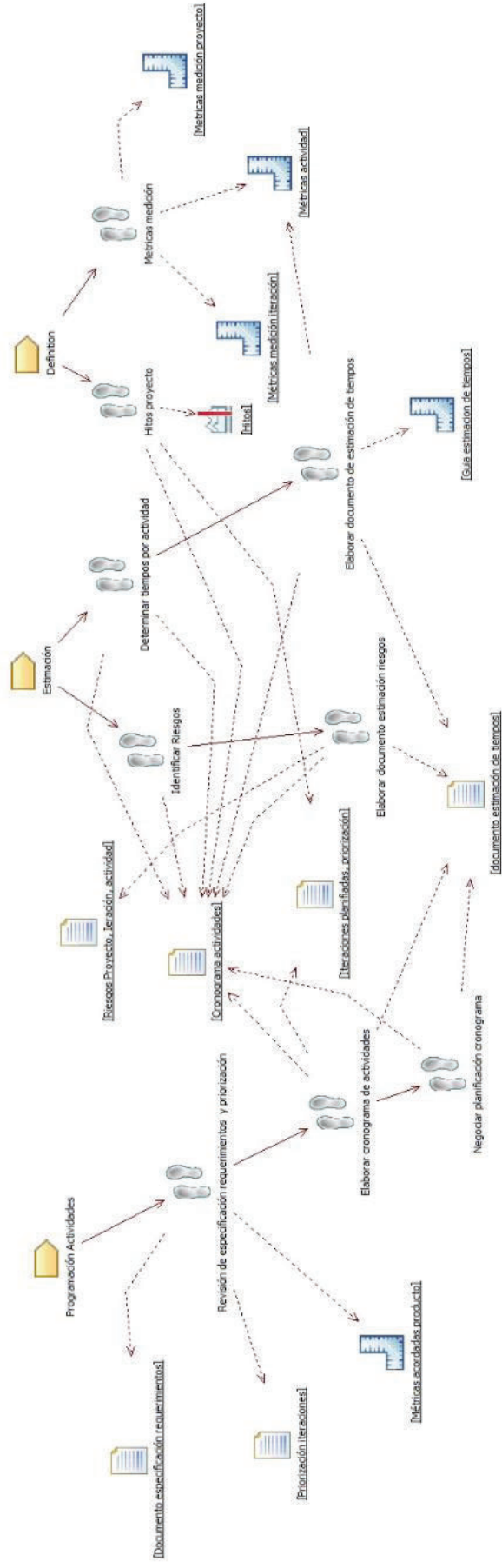


Figura 16. Detalle del modelo para la tarea de Planificación

ANEXO 6

Aplicación del modelo MGDS-VSE al caso estudio.

Aplicación Componente Estructural de la Gobernanza
Estructura de Gobierno
<p>Administrador del proyecto: Director del Área de Docencia e Investigación - HEE. La Unidad de Gestión de Docencia e Investigación tiene como misión, fomentar las actividades de formación de los profesionales de la salud e investigación y gestionar los convenios de colaboración con instituciones académicas.</p> <p>Responsable del desarrollo: 1 Experto del negocio - HEE</p> <p>Maestro SCRUM: 1 Investigador UPS, líder del proyecto</p> <p>Miembro del equipo SCRUM: 2 Ayudantes de investigador UPS</p> <p>Dueño del producto - Experto del negocio: 1 Funcionario designado que conoce de la gestión de Quirófanos</p> <p>Comité para la recepción: 1 integrante por área Docencia e Investigación, TICS, Especialidades clínicas y/o Quirúrgicas</p> <p>Parte interesada directa e indirecta: Docencia e Investigación, Especialidades clínicas y/o Quirúrgicas, Hospital del Día, Estadística, TICS.</p>
Procesos
Alinear, Planificar y Organizar (APO)
AP01 Gestionar el Marco de Gestión de TI
<p>El Administrador del Proyecto es responsable de:</p> <p>01 Definir la estructura organizativa.</p> <ul style="list-style-type: none"> - Designa a los funcionarios que colaborarán en el proyecto, de acuerdo a sus destrezas o conocimientos relacionados con la gestión de quirófanos. <p>02 Establecer roles y responsabilidades.</p> <ul style="list-style-type: none"> - Establece que las áreas de Docencia e Investigación, Especialidades clínicas y/o Quirúrgicas, Hospital del Día y estadística provean de toda la información requerida para la gestión de los quirófanos. - Establece que las responsabilidades sobre las funcionalidades a ser implementadas provengan de un acuerdo entre las partes interesadas, cuando exista un conflicto el responsable de desarrollo tomará la decisión final. - El responsable del desarrollo con asesoramiento de un funcionario de TICs, acordaran con el líder del proyecto UPS, el alcance, los requerimientos a ser implementados, la plataforma de desarrollo y las métricas de medición del avance y gestión de la calidad. - Todo requerimiento nuevo o ajuste del mismo, debe ser canalizado a través del responsable del desarrollo. - El responsable del desarrollo deberá capacitar a los responsables de las pruebas, para que utilicen las métricas acordadas y se ejecuten las pruebas utilizando procesos estandarizados. <p>04 Comunicar la gestión de los objetivos y la dirección. El responsable del desarrollo es encargado de:</p> <ul style="list-style-type: none"> - Comunicar el conocimiento y comprensión de los objetivos, alcance y los requerimientos acordados para el desarrollo a la dirección, a las áreas interesadas y usuarios.
P08 Gestionar las Relaciones
<p>El Responsable de desarrollo, es el responsable de:</p> <p>04 Coordinar y comunicar.</p> <ul style="list-style-type: none"> - Trabajar con las partes interesadas y coordinar la entrega final de la aplicación.
Supervisar, Evaluar y Valorar (MEA)
MEA01 Supervisar, Evaluar y Valorar Rendimiento y Conformidad.

<p>El Responsable del desarrollo y el funcionario designado de TICS, se encargan de:</p> <ul style="list-style-type: none"> - Establecer un proceso colaborativo entre las partes interesadas para establecer y mantener un enfoque de monitoreo y control del alcance. - Las partes interesadas emitirán reportes individuales y conjuntos de su apreciación en relación con el cumplimiento de los requerimientos acordados.
<p>Mecanismos relacionales</p> <ul style="list-style-type: none"> - Principio de la participación activa de los interesados - Colaboración entre las principales partes interesadas - Diálogo estratégico y participación activa de las principales partes interesadas - Recompensas e incentivos de la asociación

<p>Aplicación Célula de Gobernanza Componente Dinámico</p>
<p>Estructuras</p> <ul style="list-style-type: none"> - Formato estándar IEEE830 para la Especificación de requisitos funcionales y no funcionales que debe satisfacer la aplicación. - Estructura de la pila del producto: (id, prioridad, descripción, e_e, n_s, observación, c_v, p_a, cu_hu) (Clave única del registro de la pila, Registro de la prioridad del requerimiento, Descripción del requerimiento, Estimación del esfuerzo, N° de Sprint en el que se realiza, Observaciones adicionales al requerimiento, Criterio de validación, Persona asignada como responsable, Caso de uso o historia de usuario)
<p>Procesos Relacionados con Desarrollo de software</p>
<p>Alinear, Planificar y Organizar (APO)</p>
<p>AP01 Gestionar el Marco de Gestión de TI</p> <p>El responsable de desarrollo, líder proyecto, y funcionario de TICS, son los encargados de:</p> <ul style="list-style-type: none"> - Adoptar formato estándar IEEE830 para la Especificación de requisitos funcionales y no funcionales. - Adoptar SCRUM para la gestión del proyecto de desarrollo de la aplicación. - Adoptar UWE, para el desarrollo de la aplicación utilizando las mejores prácticas de Ingeniería Web, elaborar los diagramas de casos de uso, diagrama del modelo entidad relación físico de la base de datos, diagrama clases, de secuencia, de navegación, de presentación, de procesos. - Establecer los criterios de verificación y aceptación para: Manuales de usuario, manual de instalación, manual técnico.
<p>BAI02 Gestionar la Definición de Requisitos</p> <p>El dueño del producto, maestro SCRUM y el administrador del proyecto son los responsables de:</p> <ul style="list-style-type: none"> - Levantar el documento de especificación de requerimientos entre las partes interesadas, utilizando el estándar IEEE830. - Negociar el alcance a partir del documento de especificación de requerimientos entre las partes interesadas, el dueño del producto, maestro SCRUM y el administrador del proyecto. - Determinar el número de iteraciones para construir el producto. El número de iteraciones para la construcción del proyecto se estimó en 4. - Determinar el tiempo de la duración de la iteración. El tiempo estimado para la iteración se estimó en 1 mes (30 días) - Determinar el tiempo requeridos para entregar la documentación técnica requerida y la realización de las pruebas de aceptación y conformidad del producto final.
<p>AP11 Gestionar la Calidad</p> <p>El dueño del producto, más los 3 funcionarios designados como integrantes del comité para la recepción son los responsables de:</p> <ul style="list-style-type: none"> - Verificar y comunicar el cumplimiento de los requisitos acordados entre las partes para cada iteración. - Reportar la funcionabilidad, usabilidad y navegabilidad de cada componente entregado al término de cada iteración.

<p>- Aplicar las recomendaciones reportadas en las pruebas de aceptación de cada componente al término de la iteración.</p> <p>AP12 Gestionar el Riesgo El dueño del producto, maestro SCRUM, y funcionario designado TICS son los responsables de:</p> <ul style="list-style-type: none"> - Verificar que los componentes y el producto final cumplan las normas y requerimientos de TI, en relación con las seguridades y plataforma de desarrollo. - Verificar que el software cumple las leyes y regulaciones de la institución.
<p>BAI08 Gestionar el Conocimiento El dueño del producto, el investigador líder del proyecto y los ayudantes del investigador son los responsables de:</p> <ul style="list-style-type: none"> - Levantar la documentación técnica necesaria para garantizar la mantenibilidad del software, manual de usuario, manual técnico, instalación.
<p>MEA01 Supervisar, Evaluar y Valorar Rendimiento y Conformidad. El responsable de desarrollo y el dueño del producto son los responsables de:</p> <ul style="list-style-type: none"> - Establecer un mecanismo para colaborar entre las partes interesadas para establecer mecanismos de verificación de cumplimiento de requisitos, rendimiento y de conformidad con el producto recibido en forma conjunta.
<p>Mecanismos relacionales</p> <ul style="list-style-type: none"> - Principio de la participación activa de los interesados - Colaboración entre las principales partes interesadas - Resolución activa de conflictos. Se produjo un cambio de director en TI, lo cual trajo inestabilidad al proyecto, debido a que el nuevo funcionario inicialmente desconoce la autoridad del funcionario asignado como dueño del producto, lo que posteriormente es corregido por el administrador del proyecto.

Fase de Conceptualización de la AppPartesOperatorios
<p>Actividad: Comunicación Iteración partida</p> <p>Tareas</p> <p>1) Entender el problema que se está planteando, proponer la solución en forma conjunta entre el administrador de la aplicación, el dueño del producto, el experto del negocio y el líder del proyecto. Producto: Documento de especificación de requerimientos inicial, con base a un entendimiento claro de los requerimientos del problema y las métricas que se establecen para la evaluación del producto a entregar.</p> <p>2) Consensuar, difundir y socializar los requerimientos, con los diferentes actores que están involucrados en el proyecto. Producto: Documento de especificación de requerimientos definitivo acordado y firmado por las partes que colaboran en el proyecto.</p> <p>3) Determinar el tiempo que debe utilizarse para la iteración. Producto: Estimación del tiempo por iteración en 30 días.</p> <p>4) Determinación del número de iteraciones y priorización de las iteraciones. Producto: estimación en 4 iteraciones para entregar el producto, sin contar la planificación inicial o fase de comunicación. Especificación y priorización de las iteraciones: Iteración 1. Gestión de la seguridad: <ul style="list-style-type: none"> - Control de Acceso - Definición de Roles - Registro de Profesionales Iteración 2. Gestión del servicio: <ul style="list-style-type: none"> - Pedido de Cirugía </p>

- Aprobación de Solicitud
 - Registro Parte Operatorio
 Iteración 3. Difusión información:
 - Publicaciones en las pantallas previstas para difundir la planificación de las cirugías a los usuarios
 Iteración 4. Pruebas de aceptación, recepción del producto:
 Producto: Plan de pruebas de aceptación, documentación técnica del producto
 La plataforma de desarrollo: Java JEE, JSF, Framework PrimeFaces, gestor de base de datos MySQL.

Planificación y ejecución de las Iteraciones (SCRUM)

Actividad: Comunicación de la Iteración.

Especificación del requerimiento: automatizar la gestión del proceso del parte operatorio, registro de la solicitud de cirugía, registro de la información de los resultados de la cirugía, y facilitar el acceso a los datos.

Descripción del proceso:

El proceso general para ingreso a cirugía se estructura de la siguiente manera:

Subproceso 1:

El paciente ingresa por consulta externa, para chequeo de un especialista por alguna enfermedad, activa el subproceso dos.

Subproceso 2:

El especialista realiza un diagnóstico del paciente, decide si necesita cirugía o no.

- Sí, ingreso de paciente a hospitalización, activa el subproceso 3.
- No, envía medicación y nueva consulta.

Subproceso 3:

El departamento de estadística procede al registro de datos, y notifican en hospitalización el ingreso a piso del paciente, dependiendo de la especialidad, activa el subproceso 4.

Subproceso 4:

Una vez hospitalizado es nuevamente valorado por especialistas, activa el subproceso 5.

Subproceso 5:

El paciente es sometido a exámenes médicos, solicitados en el subproceso 4, y decide si el paciente intervenido quirúrgicamente:

- Sí, el paciente ingresa al proceso de tratamiento posquirúrgico, activa el subproceso 8.
- No, verifica si cumple con los requisitos pre-quirúrgicos y decide lo siguiente:
 - Sí, se registra en el parte operatorio la solicitud, activa el subproceso 6.
 - No, el paciente ingresa al proceso de tratamiento pre-quirúrgico hasta estabilizarlo y poder planificar el parte operatorio, regresa al subproceso 4.

Subproceso 6:

Se verifica si necesita medicamentos y/o implementos médicos, se decide lo siguiente:

- Sí, descargo de medicamentos y/o insumos médicos, se procede a la cirugía.
- No, Se procede a la cirugía con implementos básicos.

Subproceso 7:

Terminada la cirugía, se registra en el parte operatorio, la información del procedimiento y observaciones, si fue satisfactorio o no:

- Sí, paciente a recuperación, regresa a subproceso 4.
- No, cadáver a la morgue

Subproceso 8:

El paciente sale del hospital con el alta médica.

Actividad: Planificación del incremento, entrega del incremento.

1) Planificación de actividades y tiempos para las actividades prevista para cada iteración.

Iteración 1. Gestión de la seguridad:

- Control de Acceso
- Definición de Roles

<ul style="list-style-type: none"> - Registro de Profesionales <p>Iteración 2. Gestión del servicio:</p> <ul style="list-style-type: none"> - Pedido de Cirugía - Aprobación de Solicitud - Registro Parte Operatorio <p>Iteración 3. Difusión información:</p> <ul style="list-style-type: none"> - Publicaciones en las pantallas previstas para difundir la planificación de las cirugías a los usuarios <p>Iteración 4. Pruebas de aceptación, recepción del producto:</p> <p>Producto: Plan de pruebas de aceptación, documentación técnica del producto</p> <p>Entregables:</p> <p>Cronograma de actividades con los requerimientos detallados en la especificación de requisitos, estimación de riesgos, documento de negociación del alcance.</p>
<p>5) Medición de la Iteración.</p> <ol style="list-style-type: none"> 1) Evaluar el incremento entregado, en función de las guías de medición de control de calidad. 2) Elaborar el reporte de resultados. <p>Entregable: Reporte de resultados</p>
<p>Fase de Control y Seguimiento</p>
<p>6) Despliegue</p> <ol style="list-style-type: none"> 1) Pruebas de integración. 2) Revisión de métricas de control de calidad de la iteración, documentación de soporte del incremento. 3) Elaborar: reporte de aceptación de los usuarios, reporte de cumplimiento de las métricas de calidad, reporte de pruebas de integración. <p>Entregables:</p> <p>Reporte de pruebas de integración, reporte de aceptación y cumplimiento de métricas.</p>