CREACIÓN DE UNA METODOLOGÍA CON TÉCNICAS ÁGILES PARA EL ÁREA DE PRUEBAS DE SOFTWARE EN TECHREA S.A.S

CHARLES EDILBERTO HOLGUÍN ORDÓÑEZ

Código: 20151099006

Especialización Ingeniería de Software

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

FACULTAD DE INGENIERÍA

ESPECIALIZACIÓN EN INGENIERÍA DE SOFTWARE

BOGOTÁ

2015

CREACIÓN DE UNA METODOLOGÍA CON TÉCNICAS ÁGILES PARA EL ÁREA DE PRUEBAS DE SOFTWARE EN TECHREA S.A.S

CHARLES EDILBERTO HOLGUÍN ORDÓÑEZ

Trabajo de grado presentado como requisito parcial para optar al título de:

Especialista en Ingeniería de Software

Revisor

Ingeniero Jhon Jairo Londoño

Director

Ingeniera Lilian Bejarano

Línea de investigación: Ingeniería de software

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

FACULTAD DE INGENIERÍA

ESPECIALIZACIÓN EN INGENIERÍA DE SOFTWARE

BOGOTÁ

2015

Dedicatoria

Dedicado a todas y cada una de las personas que amo, quienes compartieron conmigo y aportaron experiencias y consejos de vida durante los últimos años, con las cuales sin su apoyo moral y compañía no hubiese sido posible la culminación de una meta más. En especial a mí madre María Teresa Ordóñez, por su comprensión y apoyo incondicional, a mis hermanos Maribel, Jaime y Ronald Holguín, a mi padre, a mi novia Lizeth Ríos, por su amor y apoyo.A mi familia y amigos muchas gracias... Charles E. Holguín O.

Agradecimientos

Quiero agradecer al creador por permitir la culminación de un logro más en mi vida, a todas las personas que aportaron su conocimiento, tiempo, experiencias y consejos en la realización de este trabajo, a la Universidad Distrital Francisco José de Caldas, a los profesores que durante el último año brindaron su conocimiento y experiencia en pro del aprendizaje, a mi Director, la Ingeniera Lilian Bejarano, a mi Revisor, el Ingeniero Jhon Jairo Londoño, los cuales brindaron apoyo, conocimiento para la terminación del proyecto de grado, también hago extensivo este agradecimiento a la empresa Techrea S.A.S, a sus empleados, al área de tecnología, a su gerente Fredy Taffur, por brindarme su apoyo en todo momento y permitirme realizar la investigación en sus instalaciones. A todas y cada una de las personas que estuvieron involucradas en este largo proceso, muchas gracias por su apoyo. Charles E. Holguín O.

GLOSARIO

La siguiente tabla muestra la lista de términos utilizados en este documento con su definición.

Término	Definición
Ciclo de vida	Conjunto organizado de actividades y fases, tanto técnicas como de gestión que es necesario desarrollar a lo largo de la vida del sistema.
Cmmi	Integración de modelos de madurez de capacidades o (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.
Sprint backlog	Lista de tareas a realizar en la iteración de un proyecto.
Ieee	Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y Electrónicos, es una asociación dedicada a la estandarización de procesos.
Sprint	Agrupa actividades con un tiempo específico en el cual se ejecutaran las tareas planificadas.
Sprint review	Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario.
Daily scrum	El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas.
Sprint goal	El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Lista de Producto.
Sprint planning meeting	El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint.
Jira	Es una aplicación basada en web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. Jira también se utiliza en áreas no técnicas para la administración de tareas.
ToDo	Es un tipo de actividad utilizado por la herramienta Jira para demarcar un que hacer.
Bug	Es un tipo de incidencia o reporte de un defecto encontrado en el sistema.
MEND – TEST	Proceso testing aplicable al área de pruebas, proporciona agilidad al proceso de pruebas.

Table 1: Tabla Siglas

Resumen Analítico

El desarrollo de software durante los últimos años ha estado en constante crecimiento y evolución, la exigencia que demanda estar a la vanguardia en esta industria, representa para las empresas tecnológicas un reto, no solo por lo competitivo del mercado, sino también por la exigencia que cada día demandan los usuarios a la hora de adquirir productos de alta calidad.

Dentro del ciclo de vida del desarrollo, la implementación de metodologías y herramientas dentro de las áreas de pruebas de software, representa un porcentaje mínimo, respecto a las demás tareas de software, esto permite que las empresas incurran en costos adicionales, esfuerzos innecesarios, demoras en sus entregas y por ende pérdida de credibilidad ante sus clientes.

El presente trabajo de grado propone satisfacer las necesidades del área de pruebas de la compañía TechRea S.A.S mediante el establecimiento de un proceso testing con técnicas ágiles, la cual aportará dinamismo y mejorará la calidad de los productos software entregados por la compañía.

ABSTRACT

Software development, in recent years, has been growing and evolving constantly; the request of being on the top of this industry, represents a challenge for technology companies, not only for the competitiveness of actual markets, but for the high users expectations at the time of buying products of high quality.

Inside the life cycle of development; implementation of methodologies and tools in the areas of software testing represents a lower percent relative to the other tasks in software it makes companies incurring in additional costs and unnecessary efforts delays in deliveries and of course loss of credibility with their customers.

This paper proposes grade meet the needs of the test area techrea sas company by establishing a management methodology with agile techniques, which provide dynamism and improve the quality of software products delivered by the company.

Tabla de Contenido

Dedicado a **3**

Agradecimientos **4**

1 CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN **12**
1.1 Identificación del problema . . . 12
1.2 Objetivos de la Investigación . . . 12
1.2.1 Objetivo General . . . 12
1.2.2 Objetivos Específicos . . . 12
1.3 Justificación práctica . . . 13
1.4 Hipótesis . . . 13
1.5 Marco de Referencia . . . 13
1.5.1 Marco teórico . . . 13
1.6 Metodología de la Investigación . . . 17
1.6.1 Marco conceptual . . . 17

2 ARQUITECTURA EMPRESARIAL ALTO NIVEL **23**
2.1 Capa de negocio . . . 24
2.1.1 Metamodelo del punto de vista de organización . . . 24
2.1.2 Modelo del punto de vista de organización . . . 24
2.1.3 Metamodelo del punto de vista de cooperación de actor . . . 25
2.1.4 Modelo del punto de vista de cooperación de actor . . . 25
2.1.5 Metamodelo del punto de vista función de negocio . . . 26
2.1.6 Modelo del punto de vista función de negocio . . . 26
2.1.7 Metamodelo del Punto de vista de Proceso de Negocio . . . 27
2.1.8 Módelo del Punto de vista de Proceso de Negocio . . . 27
2.1.9 Metamodelo del Punto de vista de Cooperación de Proceso de Negocio . 28
2.1.10 Módelo del Punto de vista de Cooperación de Proceso de Negocio . . . 28
2.1.11 Metamodelo del Punto de vista de producto . . . 29
2.1.12 Módelo del Punto de vista de producto . . . 29
2.2 Capa Codificación . . . 30
2.2.1 Metamodelo comportamiento de aplicación . . . 30
2.2.2 Módelo comportamiento de aplicación . . . 30
2.2.3 Metamodelo Punto de vista de Cooperación de Aplicación . . . 31
2.2.4 Módelo Punto de vista de Cooperación de Aplicación . . . 31
2.2.5 Metamodelo Punto de vista de Estructura de Aplicación . . . 32
2.2.6 Módelo Punto de vista de Estructura de Aplicación . . . 32
2.2.7 Metamodelo Punto de vista de Uso de Aplicación . . . 33
2.2.8 Módelo punto de vista de Uso de Aplicación . . . 33
2.3 Capa Infraestructura . . . 34
2.3.1 Metamodelo Punto de vista de Infraestructura . . . 34

2.3.2 Módelo Punto de vista de Infraestructura . . . 34
2.3.3 Metamodelo Punto de vista de Uso de Infraestructura . . . 35
2.3.4 Módelo punto de vista de Uso de Infraestructura . . . 35
2.3.5 Metamodelo Punto de vista de Implementación y despliegue . . . 36
2.3.6 Módelo Punto de vista de Implementación y despliegue . . . 36
2.3.7 Metamodelo punto de vista de Estructura de Información . . . 37
2.3.8 Módelo punto de vista de Estructura de Información . . . 37
2.3.9 Metamodelo Punto de vista de Realización del Servicio . . . 38
2.3.10 Módelo Punto de vista de Realización del Servicio . . . 38
2.3.11 Módelo Punto de vista de Capas . . . 39
2.4 Capa Motivacional . . . 40
2.4.1 Metamodelo Punto de vista de Implicados . . . 40
2.4.2 Módelo Punto de vista de Implicados . . . 40
2.4.3 Metamodelo Punto de vista de Realización de Objetivos . . . 41
2.4.4 Módelo Punto de vista de Realización de Objetivos . . . 41
2.4.5 Metamodelo Punto de vista de Contribución de Objetivos . . . 42
2.4.6 Módelo Punto de vista de Contribución de Objetivos . . . 42
2.4.7 Metamodelo Punto de vista de Principios . . . 43
2.4.8 Módelo Punto de vista de Principios . . . 43
2.4.9 Metamodelo punto de vista de Realización de Requerimientos . . . 44
2.4.10 Módelo punto de vista de Realización de Requerimientos . . . 44
2.4.11 Metamodelo Punto de vista de Motivación . . . 45
2.4.12 Módelo Punto de vista de Motivación . . . 45

3 DESARROLLO DE LA INVESTIGACIÓN 46
3.1 Situación Actual del proceso del área de pruebas . . . 46
3.1.1 Procedimiento al ingresar un proyecto de pruebas . . . 47
3.2 Metodología Mend-Test . . . 48
3.3 Mend-Test . . . 48
3.3.1 Fases planteadas (Planeación- Diseño - Ejecución - Cierre) . . . 51
3.3.2 Características . . . 52
3.3.3 Proceso guiado por ciclos “Sprint” . . . 52
3.3.4 Roles . . . 54
3.3.5 Ventajas de la metodología mend-test . . . 54
3.3.6 Desventajas de la metodología mend-test . . . 54

4 CIERRE DE LA INVESTIGACIÓN 55
4.1 Conclusiones . . . 55
4.2 Recomendaciones . . . 55
4.3 Trabajos Futuros . . . 56
4.4 Bibliografía . . . 57
4.5 Infografía . . . 58

Lista de Figuras

1.1 Grupos de procesos de la gestión de proyectos (Swebok 2012) 14
1.2 Predictiva Vs Adaptable (Juan Palacio, 2008) 15
1.3 Criterios de idoneidad para un proyecto (Juan Palacio, 2008)) 16
1.4 Relación de proyectos exitosos de la Standish Group International (2015)) 18
1.5 Conceptos relacionados con las pruebas de software. (Majchrzak, 2012)) 19
1.6 Procedo de depuración (Roger S. Pressman, 2005) 20
1.7 Ciclo de trabajo de Scrum (Juan Palacio, 2008)) 22

2.1 Capas de la arquitectura empresarial - SJBC, Coloso Archimate 23
2.2 Metamodelo del punto de vista de organización 24
2.3 Módelo del punto de vista de organización - SJBC, Coloso Archimate 24
2.4 Metamodelo punto de vista de cooperación de actor - SJBC, Coloso Archimate . 25
2.5 Modelo punto de vista de cooperación de actor- SJBC, Coloso Archimate 25
2.6 Metamodelo del punto de vista función de negocio- SJBC, Coloso Archimate . . 26
2.7 Modelo del punto de vista función de negocio - SJBC, Coloso Archimate 26
2.8 Metamodelo del Punto de vista de Proceso de Negocio - SJBC, Coloso Archimate 27
2.9 Módelo del Punto de vista de Proceso de Negocio - SJBC, Coloso Archimate . . 27
2.10 Metamodelo del Punto de vista de Cooperación de Proceso de Negocio - SJBC, Coloso Archimate 28
2.11 Módelo del Punto de vista de Cooperación de Proceso de Negocio - SJBC, Coloso Archimate 28
2.12 Metamodelo del Punto de vista de producto - SJBC, Coloso Archimate 29
2.13 Módelo del Punto de vista de producto - SJBC, Coloso Archimate 29
2.14 Metamodelo comportamiento de aplicación - SJBC, Coloso Archimate 30
2.15 Módelo comportamiento de aplicación 30
2.16 Metamodelo Punto de vista de Cooperación de Aplicación - SJBC, Coloso Archimate 31
2.17 Módelo Punto de vista de Cooperación de Aplicación 31
2.18 Metamodelo Punto de vista de Estructura de Aplicación - SJBC, Coloso Archimate 32
2.19 Módelo Punto de vista de Estructura de Aplicación - SJBC, Coloso Archimate . 32
2.20 Metamodelo Punto de vista de Uso de Aplicación - SJBC, Coloso Archimate . . 33
2.21 Módelo punto de vista de Uso de Aplicación - SJBC, Coloso Archimate 33
2.22 Metamodelo Punto de vista de Infraestructura - SJBC, Coloso Archimate 34
2.23 Módelo Punto de vista de Infraestructura - SJBC, Coloso Archimate 34
2.24 Metamodelo Punto de vista de Uso de Infraestructura - SJBC, Coloso Archimate 35
2.25 Módelo punto de vista de Uso de Infraestructura - SJBC, Coloso Archimate . . . 35
2.26 Metamodelo Punto de vista de Implementación y despliegue - SJBC, Coloso Archimate 36
2.27 Módelo punto de vista de Implementación y despliegue - SJBC, Coloso Archimate 36
2.28 Metamodelo punto de vista de Estructura de Información - SJBC, Coloso Archimate 37

2.29 Módelo punto de vista de Estructura de Información - SJBC, Coloso Archimate 37
2.30 Metamodelo Punto de vista de Realización del Servicio - SJBC, Coloso Archimate 38
2.31 Módelo Punto de vista de Realización del Servicio - SJBC, Coloso Archimate . . 38
2.32 Módelo Punto de vista de Capas 39
2.33 Metamodelo Punto de vista de Implicados - SJBC, Coloso Archimate 40
2.34 Módelo Punto de vista de Implicados - SJBC, Coloso Archimate 40
2.35 Metamodelo Punto de vista de Realización de Objetivos - SJBC, Coloso Archimate 41
2.36 Módelo Punto de vista de Realización de Objetivos - SJBC, Coloso Archimate . 41
2.37 Metamodelo Punto de vista de Contribución de Objetivos - SJBC, Coloso Archimate 42
2.38 Módelo Punto de vista de Contribución de Objetivos - SJBC, Coloso Archimate 42
2.39 Metamodelo Punto de vista de Principios - SJBC, Coloso Archimate 43
2.40 Módelo Punto de vista de Principios - SJBC, Coloso Archimate 43
2.41 Metamodelo punto de vista de Realización de Requerimientos - SJBC, Coloso Archimate 44
2.42 Módelo punto de vista de Realización de Requerimientos - SJBC, Coloso Archimate 44
2.43 Metamodelo Punto de vista de Motivación - SJBC, Coloso Archimate 45
2.44 Módelo Punto de vista de Motivación - SJBC, Coloso Archimate 45

3.1 Proceso actual Compañía - CEHO - Metodología MEND – TEST 46
3.2 Espina de pescado - CEHO - Metodología MEND – TEST 47
3.3 Criterios de selección - CEHO - Metodología MEND – TEST 48
3.4 Implementación con la metodología MEND – TEST - CEHO 49
3.5 Implementación con la metodología MEND – TEST - CEHO 50
3.6 Metodología MEND – TEST - CEHO 50
3.7 Fases de la Metodología MEND – TEST - CEHO 51
3.8 Proceso de la nueva Metodología MEND – TEST - CEHO 53

INTRODUCCIÓN

Tan solo con observar nuestro entorno, nos damos cuenta que la tecnología está implícita en todas y cada una de las áreas de conocimiento, dentro del auge de la tecnología aplicada en los últimos años. Según SWEBOK (2012), la ISO/IEC 25000 presenta la siguiente definición de calidad de software "la totalidad de la funcionalidad y prestaciones de un producto software que contribuyen con su capacidad de satisfacer necesidades explicitas o implícitas". Es por esta razón que grupos especializados implementan nuevas formas y metodologías para mitigar este riesgo.

Las compañías tecnológicas en los últimos años en Colombia han crecido de forma acelerada, según un informe realizado por Fedesoft (2014). El cual muestra que el sector de la industria del software durante los últimos cinco años aumentó su participación en el producto interno bruto en un 1.7% para 2012. Este aumento se debe a la gran demanda tecnológica generada en nuestro país. Esta industria tiene como objetivo primordial brindar servicios o productos software que satisfagan las necesidades de sus clientes, para ello se apoyan en diferentes herramientas, metodologías, procesos y talento humano, con el fin de lograr satisfacer los altos estándares de calidad exigidos por la industria en general, al igual que para facilitar su labor diaria.

Las pruebas de software son un eslabón fundamental dentro del proceso del ciclo de vida del desarrollo, ya que éstas permiten determinar si el software cumple o no con las necesidades planteadas por el cliente al inicio de los proyectos; la aplicación de buenas prácticas dentro del área de pruebas es sumamente compleja, y conlleva a la utilización de herramientas especializadas para tal fin, al igual estos grupos de pruebas deben apoyarse en metodologías y estándares establecidos por la industria, como lo es el estándar según The International Software Testing Standard (2013), para pruebas de software ISO/IEC 29119.

Desafortunadamente muchas de las compañías involucradas en el mundo de la tecnología le prestan un mínimo de atención a la fase de pruebas de software, esto se debe a diversos factores, ya sea por el desconocimiento de la importancia que tiene esta fase dentro del ciclo de vida del desarrollo, por cuestiones económicas, y muchos factores más. Es bien sabido en la industria que esta fase incrementa los costos dentro de los proyectos, según Glenford J. Myers (2004) los costos de las pruebas pueden llegar a representar entre el 30% y 50% del costo total del desarrollo del software.

Desde la perspectiva de los clientes un producto software de calidad, es aquel que satisfaga sus necesidades o requerimientos iniciales, los cuales fueron planteados al concebir el proyecto, y que garanticen que el riesgo dentro de la operación del negocio, sea mínimo.

La construcción de la metodología para pruebas de software se apoyará en los lineamientos de calidad contenidos en la norma ISO 9001 (2008). Junto con la de CMMI. El desarrollo de la metodología se guiará bajo lo indicado por PALACIO, J. (2008). Con la metodología ágil Scrum, con el objetivo de simplificar el proceso de pruebas y lograr así una disminución sustancial en los tiempos de ejecución de pruebas y en los costos del proyecto.

Por lo anterior se hace necesario proponer una metodología con técnicas ágiles para el área de pruebas de la compañía Techrea, con el propósito de facilitar el control y gestión de los proyectos de pruebas, al igual que permitir el seguimiento sobre las diferentes fases del proceso de pruebas (Planeación, diseño, ejecución, finalización y cierre).

Capítulo 1

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1 Identificación del problema

Actualmente TeChrea S.A.S tiene un procedimiento estructurado para el manejo de las pruebas de software. Es sabido que las pruebas de software se encuentran contenidas dentro del ciclo de vida del desarrollo, ya que en esta actividad es donde se realiza gran parte de las validaciones y verificaciones necesarias para la certificación de productos software con un aceptable nivel de calidad, el cual es exigido en el mercado tecnológico.

Un reporte de gestión realizado al interior de la compañía evidencio que un 45% de los proyectos ejecutados y entregados a los clientes, consumió tiempos similares de ejecución, independientemente de su naturaleza "proyectos pequeños y grandes", de igual manera se evidenciaron problemas criticos tardiamente, restringiendo la toma de deciones de forma oportuna.

Por lo anterior se hace necesario proponer una metodología con técnicas ágiles para el área de pruebas de la compañía Techrea, con el propósito de facilitar el control y gestión de los proyectos de pruebas.

Formulación del problema

¿Cómo estructurar una metodología utilizando técnicas ágiles que apoyen la ejecución y gestión de pruebas de software en TeChrea S.A.S que contribuya a la disminución de reprocesos, tiempos de ejecución de los proyectos, al igual que facilite el acceso a la información relevante para la toma de decisiones en la organización?

1.2 Objetivos de la Investigación

1.2.1 Objetivo General

Crear una metodología de pruebas utilizando técnicas ágiles, basada en la definición y construcción de estrategias, procedimientos y estándares a ser utilizados en las etapas de la fase de pruebas de software para TeChrea S.A.S

1.2.2 Objetivos Específicos

Identificar fortalezas y falencias contenidas en el proceso actual que maneja el área de pruebas, mediante la observación de sus procesos y entrevistas al personal involucrado.

Crear un nuevo proceso testing, basado en técnicas ágiles a partir de uno ya existente, que aporte dinamismo a la ejecución de pruebas dentro del área.

Aplicar la metodología creada dentro del área de pruebas de la compañía y evaluar su comportamiento, utilizando un proyecto de pruebas piloto.

1.3 Justificación práctica

El presente trabajo de grado se justifica a un nivel práctico, ya que con la investigación a realizar se aportaran las bases necesarias para mejorar las actividades que realizan actualmente las área de pruebas de la empresa TeChrea S.A.S, estructurando su procedimiento en beneficio de la mejora en la calidad de los artefactos software que se desarrollan y a su vez brindando una visión clara del estado de los proyectos para facilitar la toma de decisiones respecto a la priorización de actividades y así evitar retrasos y tiempos muertos, al igual brinda una visión clara de la importancia que tiene el software en cualquier ámbito en el que las personas se desenvuelven.

Por lo tanto da un acercamiento de alto nivel a lo indispensable que es la calidad dentro de un artefacto software. Derivando así en la oportunidad de acercarse y conocer más productos y tendencias existentes en el mercado tecnológico, como lo son las metodologías agiles, y porque estas metodologías permitirán la flexibilidad y el cambio que sufre continuamente el área de pruebas de software por ultimo aporta un camino para la adquisición, profundización y aplicación de conocimientos en el área de ingeniería de software, específicamente la fase de pruebas. Finalmente en el ámbito tecnológico aportará bases para incentivar aún más el desarrollo de productos innovadores y metodologías aplicables a la gestión y pruebas de software.

1.4 Hipótesis

Cuanto mayor estructurado sea el proceso de gestión y control sobre las actividades del área de pruebas de software, menores serán los tiempos de ejecución de los proyectos.

1.5 Marco de Referencia

1.5.1 Marco teórico

Los proyectos han estado presentes en la dinámica de las empresas, ya sea de una manera implícita o explícita; para el desarrollo de cualquiera de sus actividades es necesaria la conformación de un equipo de trabajo y por ende de una o varias metodologías, por medio de las cuales se permita implementar las soluciones aplicables a todo tipo de compañías.

Gestión de proyectos

La gestión de proyectos predictiva o clásica es una disciplina formal de gestión, basada en la planificación, ejecución y seguimiento a través de procesos sistemáticos y repetibles según (Juan Palacio, 2008).

- Establece criterios de éxito para obtener un producto definido, en el tiempo previsto y con el coste estimado.
- Asume que el proyecto se desarrolla en un entorno estable y predecible.

- El objetivo de su esfuerzo es mantener el cronograma, el presupuesto y los recursos.
- Divide el desarrollo en fases a las que considera "ciclo de vida", con una secuencia
- Definición clásica de proyecto: construcción de un resultado único, en unas fechas previstas y con unos recursos.

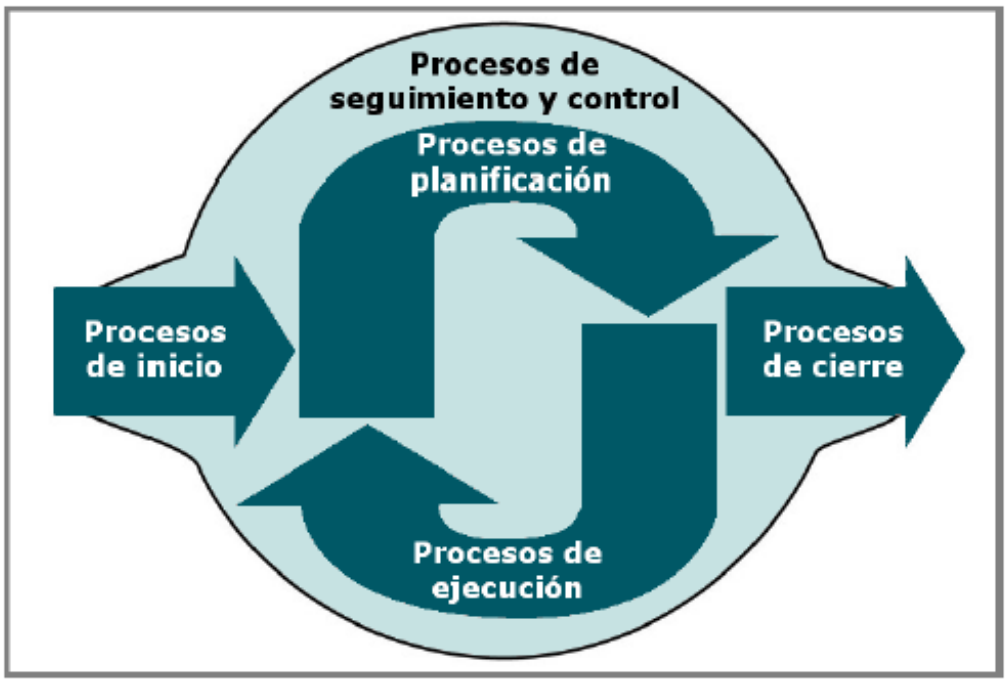


Figura 1.1: Grupos de procesos de la gestión de proyectos (Swebok 2012)

Gestión de proyectos clásica

En la actualidad los proyectos de software están evolucionando de una manera acelerada y exponencial alejándose en gran medida de sus inicios, lo cual supone un cambio en la forma de construir software y a su vez estrategias diferentes de trabajo, la rigidez que aportan los proyectos tradicionales no se adapta a las necesidades propias de nuestro tiempo, los requisitos cambiantes y las nuevas necesidades del mercado requieren de nuevos enfoques, los cuales satisfagan dicha necesidad, la gestión de proyectos brinda una reducción de tiempo agilidad y lo más importante una flexibilidad cuando sea requerido. A continuación algunas características del manifiesto ágil:

- La capacidad de respuesta al cambio, sobre el seguimiento de un plan.
- La colaboración con el cliente frente a la negociación contractual.
- Los productos que funcionan frente a especificaciones y documentaciones innecesarias.
- A las personas y su interacción por encima de los procesos y las herramientas.

Para lo anterior se han propuesto diversas metodologías como lo son Scrum, Agile Unified Process (AUP), Lean Software Development, eXtreme Programming (XP) entre muchas otras, las cuales aportan valor a su manera al desarrollo de software, en resumen según (Juan Palacio, 2008). "La gestión ágil de proyectos no es una gestión de anticipación (requisitos, diseño, planificación y seguimiento),sino de adaptación (visión, exploración y adaptación)".

La gestión de proyectos clásica define un nivel de planeación rígida y secuencial, por otro lado la gestión ágil propone otro enfoque, el cual se basa en la adaptabilidad, permitiendo así cambios necesarios dentro de los proyectos.

A continuación se muestra en la figura 1.2 los dos tipos de gestiones según (Juan Palacio, 2008). La gestión predictiva pide al equipo el cumplimiento del plan, mientras la gestión adaptable pide al equipo el mayor valor posible para una visión de producto. Para responder a la pregunta

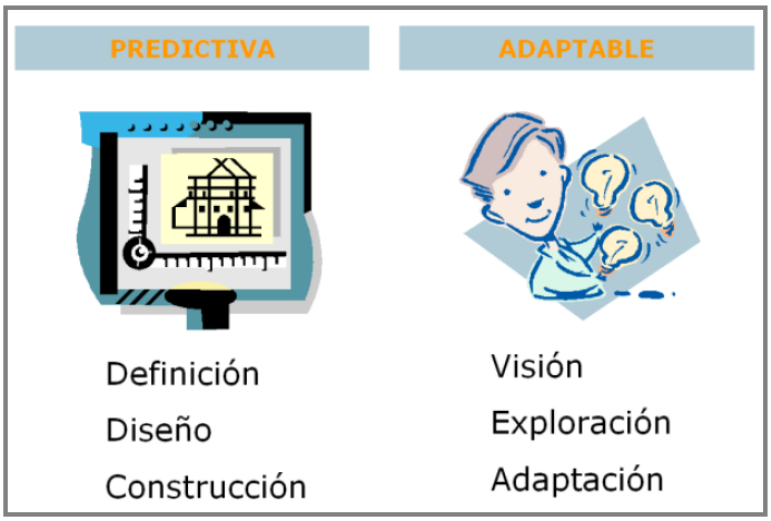


Figura 1.2: Predictiva Vs Adaptable (Juan Palacio, 2008)

de cuál de los dos enfoques utilizar se tienen que tener en cuenta diversos factores tanto del proyecto como de la organización o empresa para quien desee implementarlos. Se estipula que las características relevantes a tener en cuenta para saber cuál de los dos enfoques a utilizar es la siguiente:

- Principal prioridad de negocio.
- Estabilidad de los requisitos.
- Rigidez del producto.
- Coste de prototipo.7
- Criticidad del sistema.
- Tamaño del sistema.

A continuación se muestra un cuadro comparativo que ayudará a tomar la decisión de que enfoque es mejor utilizar dependiendo el proyecto.



Figura 1.3: Criterios de idoneidad para un proyecto (Juan Palacio, 2008))

Metodología Ágil para Pruebas

Existen varios enfoques de como se puede abordar la ejecución de las pruebas de software; como lo es la metodología Scrum, esta metodología ágil involucra las siguientes actividades dentro de su estructura:

- Backlog (repositorio) de los procesos o módulos a ser testeados.
- Planificación y desarrollo de Sprints (conjunto de procesos o módulos a ser probados) durante un periodo de 15 o 30 días máximo.
- Seguimiento contínuo del progreso de cada uno de los Sprints planificados (Burn Down Chart).
- Reuniones Diarias (15 a 30 minutos) con el equipo de pruebas (Daily Scrum Meeting) con el fin de discutir: Que se ha realizado desde la última reunión, Que se necesita realizar hasta la próxima reunión y que impedimentos existen, si los hay.
- Reunión de Planificación de los Sprints (Sprint Planning Meeting) con el equipo de pruebas, con el fin de distribuir los módulos o procesos a ser probados.
- Reunión de Revisión de los Sprints (Sprint Review Meeting), con el propósito de discutir aspectos que permitan optimizar los Sprints a realizar o pendientes. Teniendo claridad sobre conceptos, técnicas y metodología de pruebas de software, es necesario tener en cuenta que al igual que en otros procesos de la Ingeniería de Software, se requiere planificar aspectos como actividades, tareas, recursos y el tiempo en el cual se va a desarrollar.

1.6 Metodología de la Investigación

La base del presente trabajo se elaborará bajo el método inductivo – deductivo ya que la finalidad de este es dar solución a una problemática conocida. Ya que comprendiendo una necesidad se puede desarrollar una solución idónea. La cual apoyará y mejorará la visión de las pruebas de software dentro de los proyectos, mediante la utilización de metodologías ágiles de desarrollo como lo es Scrum.

Recolección de información (RI).

La recolección de información necesaria para la realización del presente trabajo de grado, se apoyará en la utilización de fuentes como artículos y libros especializados en ingeniería de software, los cuales serán consultados en bibliotecas públicas y en la existente dentro de la universidad. Se programarán una serie de entrevistas a personas especializadas en el tema de metodologías agiles y pruebas de software y aquellas involucradas en el ámbito ingenieril, en la cual se indagará y se recolectará información de forma verbal, por medio de la observación del entorno las cuales serán consignadas y analizadas dentro del marco de las diferentes actividades del ciclo de vida del desarrollo del software, y por ultimo mediante la realización de encuestas específicas sobre el tema las cuales serán publicadas en internet.

Análisis de información (AI)

Una vez recolectada la información necesaria mediante las diferentes fuentes de la etapa anterior (RI) se procederá a realizar una revisión documental para filtrar, analizar y simplificar dicha información, mediante el uso de diversas técnicas como lo es la separación de datos por criterios contextuales, temporales, cronológicos, buscando determinar así cuál de esta información es realmente útil y necesaria para dar solución a la problemática del área de pruebas de software.

Diseño de estrategias (DE)

Una vez filtrada la información, se creará un plan organizado de actividades que guiaran el proceso de investigación "la estrategia" es un mapa estructurado y muy detallado de actividades y procesos a seguir para no perder el rumbo de la investigación y la finalidad de la misma.

1.6.1 Marco conceptual

En el desarrollo de software se tiene una gran gama de metodologías y herramientas aplicables a los proyectos, ya sea para su gestión, ejecución, validación, control y seguimiento, todo esto busca finalizar los proyectos con éxito. Infortunadamente pese a todos estos medios de apoyo las estadísticas demuestran que la gran mayoría de los proyectos no culminan como inicialmente fueron planteados. Según el informe Chaos realizado por Standish Group (2015) se refleja lo siguiente.

La figura 1.4 muestra que entre los años 2011 y 2015 el 31% de los proyectos evaluados fueron exitosos, ya sea porque fueron entregados a tiempo, dentro del presupuesto establecido inicialmente, o porque cumplían con las características y funciones estipuladas en su concepción, al igual que a corte del año 2015 subió un 1% respecto al corte del año 2014.

También se evidencia que los proyectos fracasados oscilan entre el 18% y 22% en un transcurso de 5 años (2011 -2015) estos se filtraron por proyectos cancelados, proyectos que nunca se entregaron ni implementaron y por los nunca utilizados.

Pero lo más significativo es la tasa del 56% reflejada en los proyectos reto entre los años 2011

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Figura 1.4: Relación de proyectos exitosos de la Standish Group International (2015))

y 2015 por causas como, retrasos en las entregas, por encima del presupuesto inicial, o fueron entregados con menos características requeridas al inicio del proyecto.

Estas estadísticas dan un acercamiento más real a la verdadera problemática existente en el desarrollo de software, pero no indica que son las únicas causas por las cuales los proyectos de software fracasan, ya que solo se evalúan variables como tiempo de entrega, y características completas entregadas, dejando a un lado variables imprescindibles como la calidad del producto, los riesgos de la fabricación del software, satisfacción del cliente.

Gestión de proyectos

Junto con la calidad de software los proyectos se rigen bajo diversas metodologías para su desarrollo, muchos de los problemas a la hora de garantizar el éxito de los proyectos se basa en la baja gestión aplicada a los mismos, iniciando desde la licitación del proyecto, hasta su entrega final, según Boehm, B. (1981). "La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir los requisitos del mismo", al igual que lo ilustra la guía SWEBOK y PMBOK (2012), la gestión de proyectos, es secuenciada por los siguientes ítems (Iniciación y alcance, planificación, seguimiento, revisión y evaluación).

Medición del software

Luego de indagar un poco sobre la calidad de software y la gestión de proyectos, y su importancia dentro del ciclo de vida del desarrollo. Se tiene que abarcar el tema de medición en el software, ya que esta práctica permite trazar y medir lo ocurrido dentro del proyecto, evaluar si las acciones realizadas en el transcurso del proyecto fueron exitosas o por el contrario erradas, todo esto con el fin de realizar una mejora continua a la hora de realizar nuevos proyectos.

Las métricas de software reúnen, analizan, y reportan datos de algún componente o proceso, el cual permite una previa comparación con valores de referencia, permitiendo la toma de decisiones según el resultado de dicho cotejo. Al igual son una herramienta fundamental a la hora de gestionar proyectos, ya que aportan a los líderes, medios para controlar, predecir, entender y monitorear los escenarios dados en los proyectos.

Dentro de las métricas existen atributos imprescindibles a tener en cuenta, como lo son el esfuerzo, el coso y los defectos encontrados, los cuales permiten evaluar un proyecto en curso,

hacer seguimiento de riesgos potenciales, prever posibles atrasos, realizar ajustes en la marcha sobre las actividades y demás acciones necesarias en los proyectos software.

La norma ISO/IEC 15939 según MACDONALD, M. (2010). "Define los procesos necesarios para mejorar el proceso de medida del software e identificar como debe ser dicho proceso para poder atender correctamente las necesidades de información técnica y de gestión de software". Los propósitos de esta norma son; brindar un proceso unificado, pero adaptable utilizando una terminología común. Al igual permite realizar una combinación entre datos extraídos de un proceso y el de un producto y cruzarlos. En él desarrollo de la aplicación web SGGP se trabajó cíclicamente en todas y cada una de las iteraciones y versiones, es decir se realizaban los cambios necesarios acordes a los requerimientos del cliente, en las etapas que así lo requirieran, alineados a la metodología de desarrollo extreme programming (XP).

Las pruebas de software pueden ser consideradas como el proceso de verificar y validar un artefacto software en ciertas condiciones específicas, monitoreando sus resultados, evaluando los componentes del componente, proceso o sistema, pero estas van mucho más a fondo dentro de los proyectos de software ya que no solo son las encargadas de certificar la calidad de un producto software sino también de filtrar en gran medida posibles defectos embebidos en etapas anteriores.

Conceptos de Testing.

Las pruebas de software hasta hace poco han adquirido relevancia en los proyectos de software, ya que estas aportan el mayor porcentaje en la calidad de los productos software. Cuando se ejecutan pruebas de software es de vital importancia conocer algunas de las características más importantes de esta fase, como lo pueden ser los siguientes conceptos:



Figura 1.5: Conceptos relacionados con las pruebas de software. (Majchrzak, 2012))

La figura 1.5 muestra los diferentes tipos de incidentes que se pueden presentar cuando se están realizando pruebas de software, se evidencia que al desarrollar cualquier artefacto software al estar este creado por humanos, este software esta implícitamente compuesto por errores, ya que es un humano quien inyecta dichos errores, luego al realizar los diversos tipos de pruebas sobre dichos artefactos lo que se evidencian en las incidencias son defectos, el error inicial se transformó en un defecto del software, el cual es corregido y re testeado de nuevo, y cuando ese artefacto software que se construyó pasó a una etapa productiva, donde ya los usuarios finales realizan diversas actividades sobre él, en este punto es donde se encuentran fallos.

El proceso de depuración no es una prueba como tal, pero se genera en consecuencia de una, en este caso la figura 1.6 muestra sus diversos flujos.

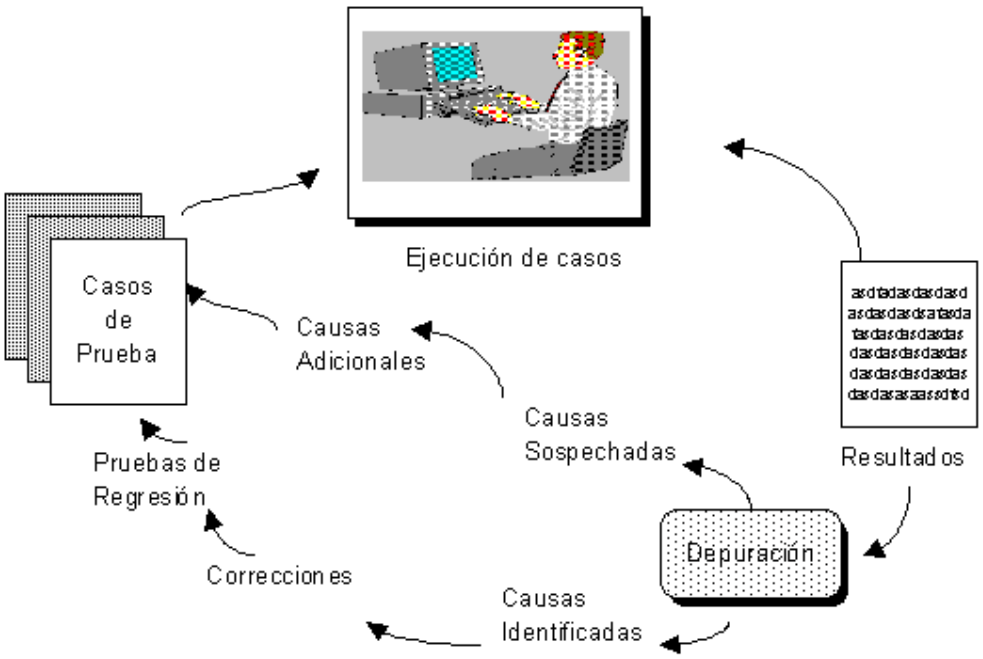


Figura 1.6: Procedo de depuración (Roger S. Pressman, 2005)

Groso modo los escenarios anteriores se repiten n veces en el transcurso de las pruebas de software, esta fase del desarrollo de software contiene los siguientes ítems:

- Casos de Prueba (TestCases): corresponde a un flujo normal y alternativo de pasos a ser probados, de acuerdo a condiciones establecidas y resultados esperados de un determinado requisito o funcionalidad de la aplicación.
- Defectos (Bugs): desperfectos o anomalías que pueden causar que el sistema falle. Normalmente son la manifestación del Error.

Técnicas de Pruebas de Software

Existen dos grandes grupos de técnicas de pruebas de software:

- Caja Negra (Black Box): se centra en comparar las entradas con las salidas esperadas, sin detenerse en detalles de su desarrollo (código fuente)(Jenkins, 2008,p. 9)
- Caja Blanca (White Box): se centra en analizar el código fuente y la lógica interna del software (Jenkins, 2008, p.9) Los resultados de la experiencia del autor se enfocan en las pruebas de Caja Negra (Black Box) donde se valida requisitos vs funcionalidades esperadas en la aplicación.

Agile Unified Process (AUP)

Esta metodología se deriva de una más completa la Rational Unified Process (RUP), AUP propone un esquema dividido en cuatro fases las cuales son inspección, elaboración, construcción, transición.

- Inspección: identificación del alcance y dimensión del proyecto, propuesta de la arquitectura y del presupuesto del cliente.
- Elaboración: Confirmación de la idoneidad de la arquitectura.
- Construcción: Desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados.
- Transición: Validación e implantación del sistema.

Scrum

Es una metodología ágil de desarrollo relativamente nueva que toma su nombre y principios de las observaciones sobre nuevas prácticas de producción, realizadas por Hirotaka Takeuchi e Ikujijo Nonaka a mediados de los 80.

Scrum es una metodología de desarrollo extremadamente simple pero requiere de un compromiso alto del equipo de trabajo, ya que a diferencia de las metodologías tradicionales esta no se basa en el seguimiento de un plan, por el contrario se enfoca en la adaptación continua a las circunstancias de la evolución del proyecto

Este método se basa en el trabajo bajo iteraciones, las cuales son la base del desarrollo ágil, Scrum gestiona su evolución en reuniones breves preferiblemente días o con periodos más extensos según sean los casos específicos de los proyectos, sin llegar a sobrepasar una semana, en estas reuniones todo el equipo revisa el trabajo realizado con anterioridad y el previsto para realizar, se exponen las dificultades que se presentaron, se plantean acciones correctivas, se evalúan las necesidades de próximos ciclos de la metodología, a continuación se muestra una imagen la cual ilustra el proceso de Scrum de una manera más clara.

En la figura 1.7 se muestra con claridad la pila del producto, la cual contiene todos los requerimientos y actividades a realizar durante el proyecto, la pila de sprint contiene aquellas actividades seleccionadas como prioritarias, las cuales se están manejando en un ciclo especifico "Sprint" el cual debería tardar en su ejecución de una a cuatro semanas según la complejidad del proyecto. Se realiza una revisión de iteraciones ejecutadas al finalizar el sprint, este método propone una evaluación constante de sus actividades, lo cual proporciona una gran ventaja respecto a otros métodos.

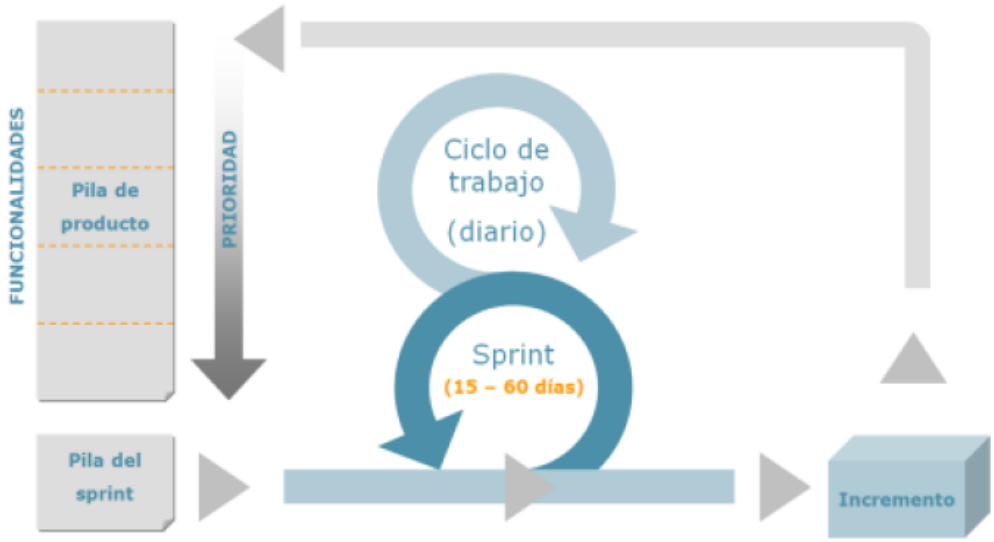


Figura 1.7: Ciclo de trabajo de Scrum (Juan Palacio, 2008))

En resumen cada ciclo o sprint es evaluado, esto con el fin de encontrar mejores formas de realizar los subsiguientes sprints, quedándose con las buenas prácticas y desechando las que no, los equipos de trabajo se organizan de una forma autónoma y colaborativa, lo cual genera una eficiencia y valor agregado al proceso.

Con lo mencionado anteriormente se deslumbra cual es el modo de operar de Scrum, pero esta forma de trabajo seria obsoleta si no intervinieran las personas, el verdadero potencial de esta metodología lo dan las personas.

Capítulo 2

ARQUITECTURA EMPRESARIAL ALTO NIVEL

Este Anexo describe la arquitectura empresarial del proyecto a un alto nivel, se enfoca en el desarrollo de los siguientes diagramas:

- Modelo de negocio
- Modelo de Codificación
- Modelo de infraestructura
- Modelo de motivación
- Modelo de migración / implementación

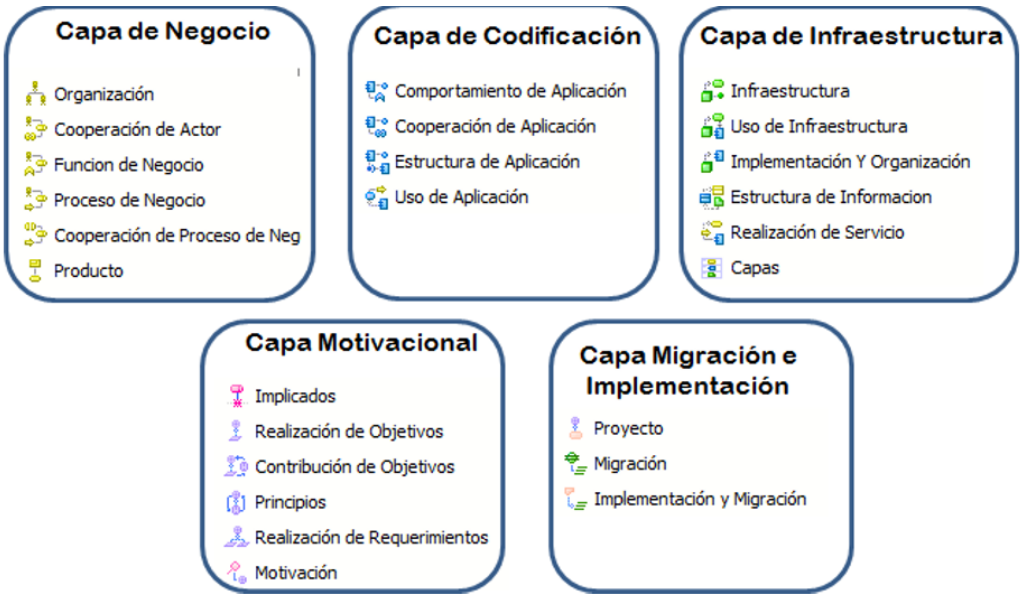


Figura 2.1: Capas de la arquitectura empresarial - SJBC, Coloso Archimate

2.1 Capa de negocio

2.1.1 Metamodelo del punto de vista de organización

Este metamodelo hace referencia a la organizacion interna de la compañia techrea, es representado comunmente con gráficas organizacionales las cuales muestran las competencias, autoridad y las responsabilidades de los involucrados.

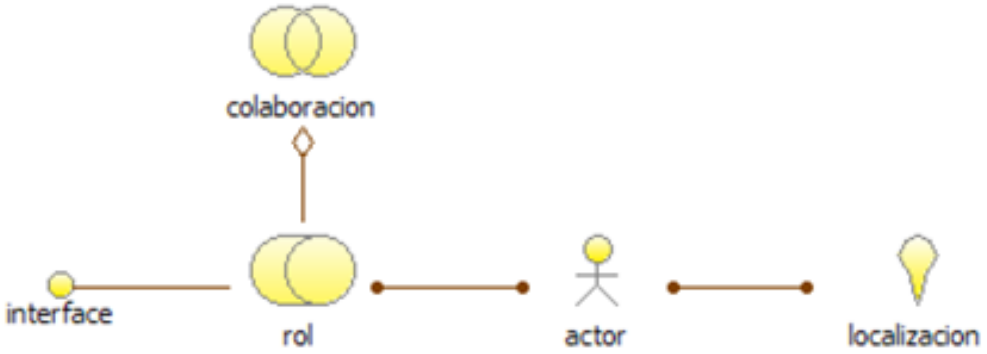


Figura 2.2: Metamodelo del punto de vista de organización

2.1.2 Modelo del punto de vista de organización

Este modelo representa la distribución jerarquica de la organización, en la cual el área global es UIP (Unidad de Innovación y Proyectos), a la cual pertenecen las áreas de proyecto y dentro de cada proyecto se encuentran las áreas de desarrollo, calificación e integración de los desarrollos manejados por la compañía.

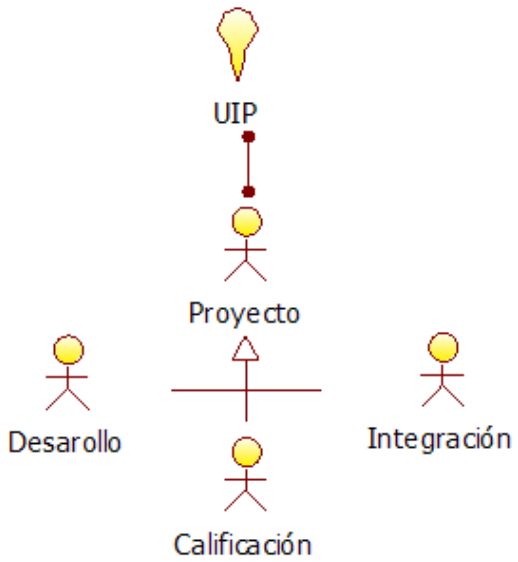


Figura 2.3: Módelo del punto de vista de organización - SJBC, Coloso Archimate

2.1.3 Metamodelo del punto de vista de cooperación de actor

Este punto de vista se enfoca en la relación que existe entre actores y su entorno, es útil para determinar las dependencias y colaboraciones mostrando la cadena de valor o la red en la cual los actores operan.

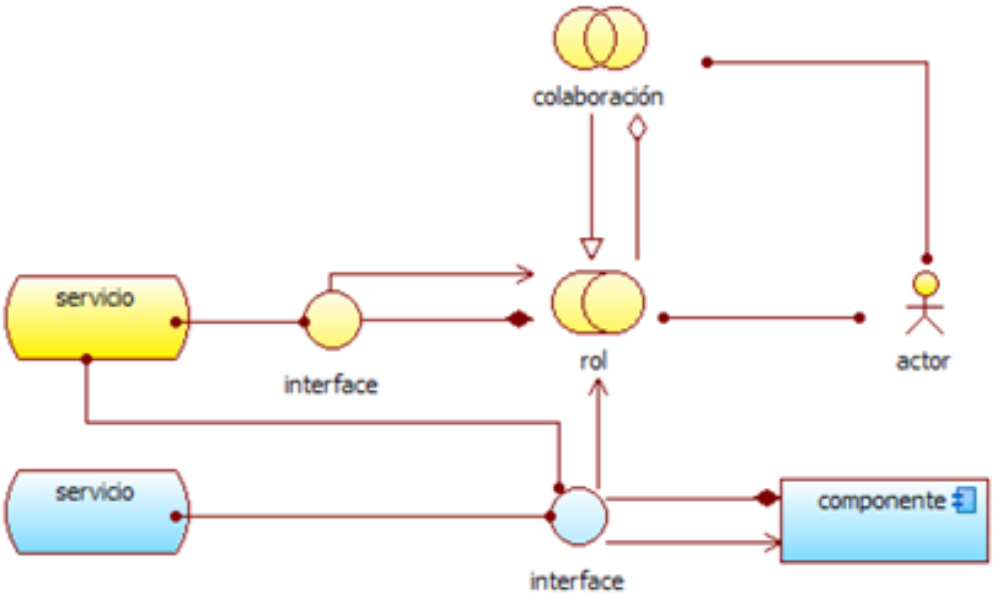


Figura 2.4: Metamodelo punto de vista de cooperación de actor - SJBC, Coloso Archimate

2.1.4 Modelo del punto de vista de cooperación de actor

Este modelo representa la cooperación existente dentro de la organización, en la cual el área de proyectos, desarrollo, calificación, integración y gerencia intecactuan entre si, al igual que muestra las diversas actividades de cada una de las áreas involucradas en el proceso.

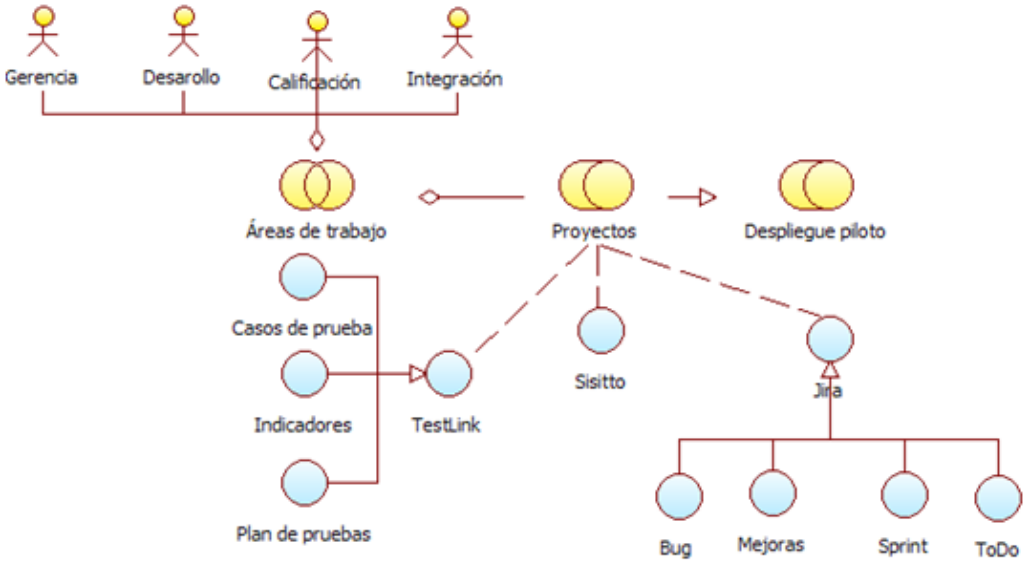


Figura 2.5: Modelo punto de vista de cooperación de actor- SJBC, Coloso Archimate

2.1.5 Metamodelo del punto de vista función de negocio

Este punto de vista muestra las funciones principales del negocio y cómo intercambian información entre ellas. Estas funciones de negocio muestran las actividades más importantes sin tener en cuenta los cambios organizacionales o los desarrollos tecnológicos.

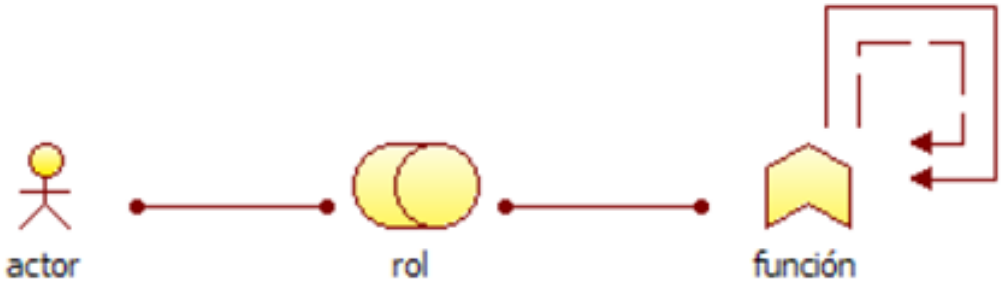


Figura 2.6: Metamodelo del punto de vista función de negocio- SJBC, Coloso Archimate

2.1.6 Modelo del punto de vista función de negocio

En el proceso global se encuentran las áreas de proyecto el cual inicia la apertura de la actividad y realiza sus diferentes artefactos, posteriormente es enviado al área de desarrollo la cual se encarga de realizar la implementación de la solución, una vez finalizada es enviada al área de pruebas para su calificación, una vez la calificación del software este aprobada se envía a integración el cual desplegara la solución a producción.

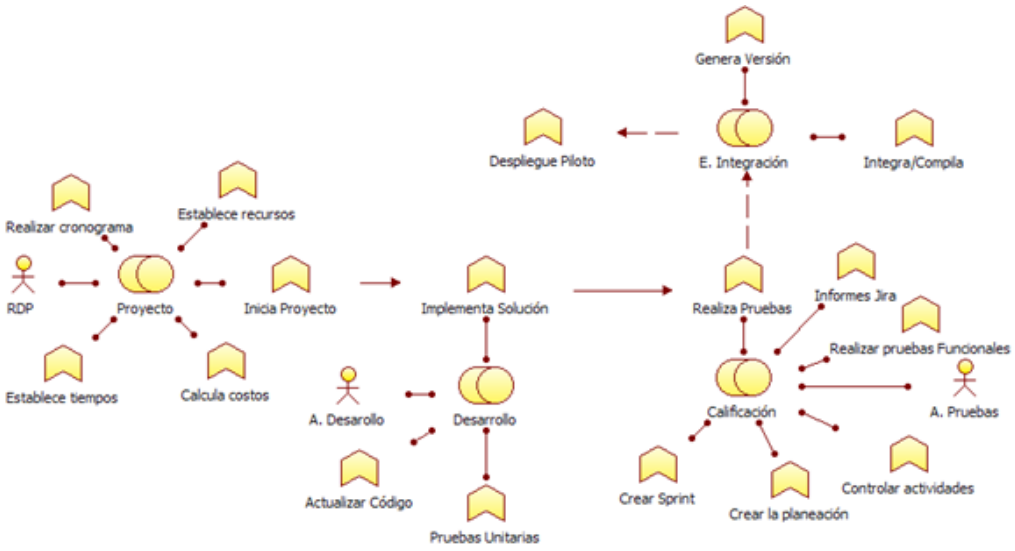


Figura 2.7: Modelo del punto de vista función de negocio - SJBC, Coloso Archimate

2.1.7 Metamodelo del Punto de vista de Proceso de Negocio

Este módelo muestra la estructura de uno o más procesos de negocio en un alto nivel, como también los servicios ofrecidos externamente por un proceso de negocio y su contribución a la realización de los productos de la compañía, se asignan roles a los procesos de negocio y así se puede identificar las capacidades y responsabilidades de cada actor.

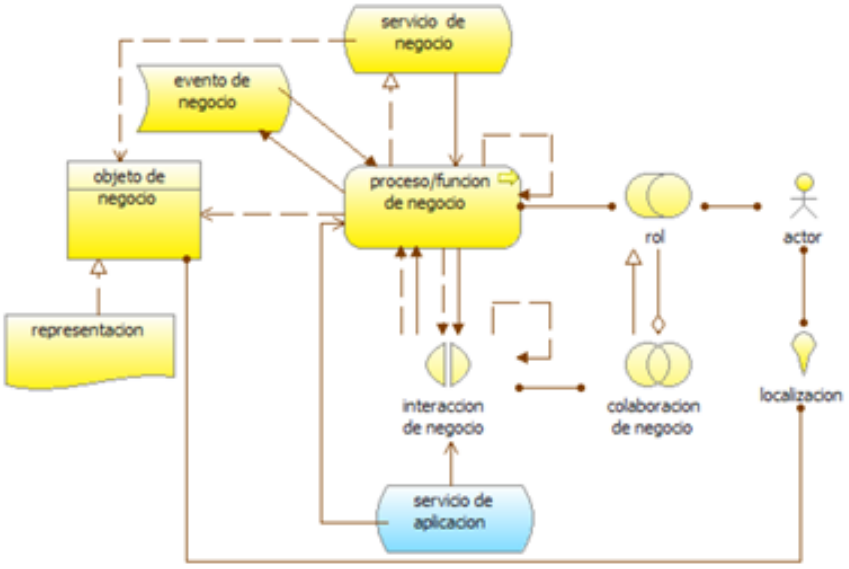


Figura 2.8: Metamodelo del Punto de vista de Proceso de Negocio - SJBC, Coloso Archimate

2.1.8 Módelo del Punto de vista de Proceso de Negocio

Este modelo muestra el proceso integral a un alto nivel de los proyectos de pruebas ejecutados en la compañía, tiene tres principales procesos los cuales, el primero realiza el inicio proyecto, una vez iniciado realiza la planeación, análisis y diseño del proyectos, luego se desarrolla y se prueba la solución, una vez culminada se despliega en producción.

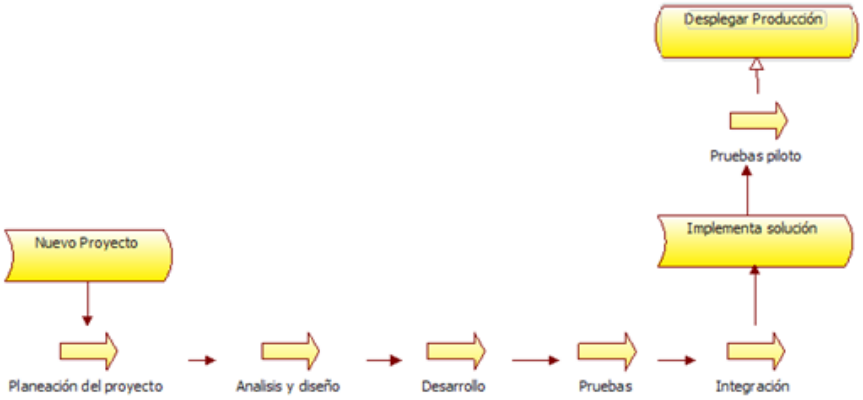


Figura 2.9: Módelo del Punto de vista de Proceso de Negocio - SJBC, Coloso Archimate

2.1.9 Metamodelo del Punto de vista de Cooperación de Proceso de Negocio

El mismo metamodelo del punto de vista de negocio es aplicable al punto de vista de cooperación de proceso de negocio. Este es punto de vista tiene aspectos importantes como: - Relaciones causales entre los principales procesos de negocio.

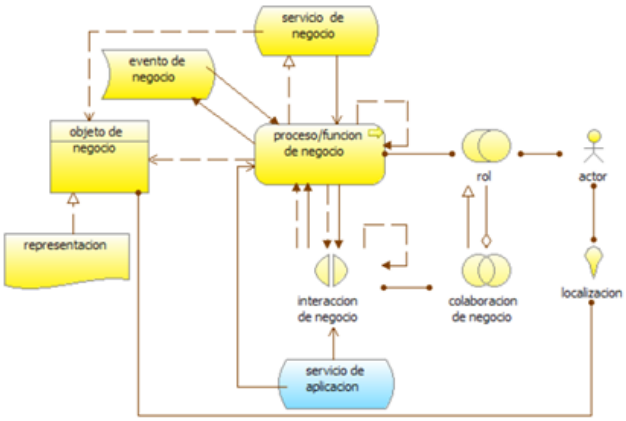


Figura 2.10: Metamodelo del Punto de vista de Cooperación de Proceso de Negocio - SJBC, Coloso Archimate

2.1.10 Módelo del Punto de vista de Cooperación de Proceso de Negocio

Este modelo punto de vista de cooperación de proceso de negocio muestra la relación entre los procesos del área de pruebas y sus diversas actividades.

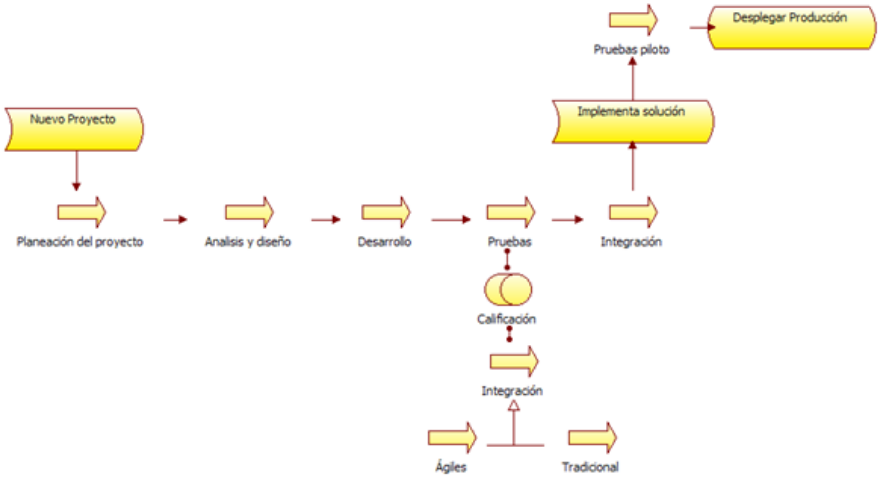


Figura 2.11: Módelo del Punto de vista de Cooperación de Proceso de Negocio - SJBC, Coloso Archimate

2.1.11 Metamodelo del Punto de vista de producto

Describe los productos que se ofrecen a los clientes u otros involucrados define la composición de estos produectos en terminos de servicios de negocio o aplicación. El producto tiene un valor (importancia del producto) y está compuesto por contratos y servicios(negocio y/o aplicación).

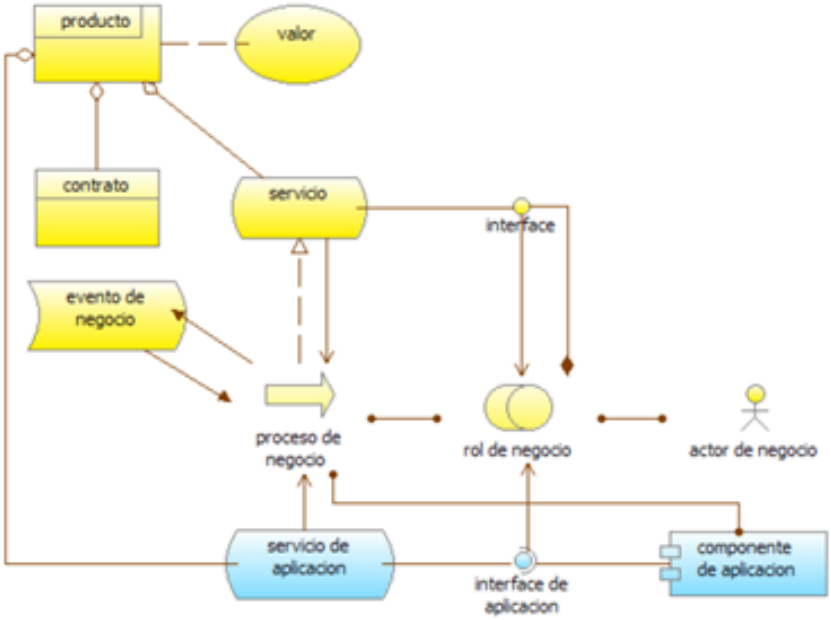


Figura 2.12: Metamodelo del Punto de vista de producto - SJBC, Coloso Archimate

2.1.12 Módelo del Punto de vista de producto

El producto visible de Techcrea es la entrega de software de calidad. Se debe cumplir el entregar el software como el cliente lo solicito y en los tiempos establecidos al inicio del proyecto.

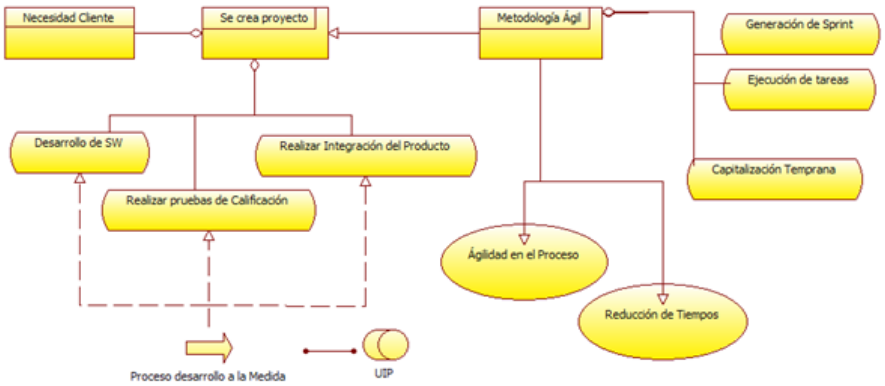


Figura 2.13: Módelo del Punto de vista de producto - SJBC, Coloso Archimate

2.2 Capa Codificación

2.2.1 Metamodelo comportamiento de aplicación

El punto de vista de comportamiento de aplicación describe el comportamiento interno de una aplicación, es decir, cómo se realizan los servicios de apliacación. Con este punto de vista se puede identificar el comportamiento de las principales aplicaciones.

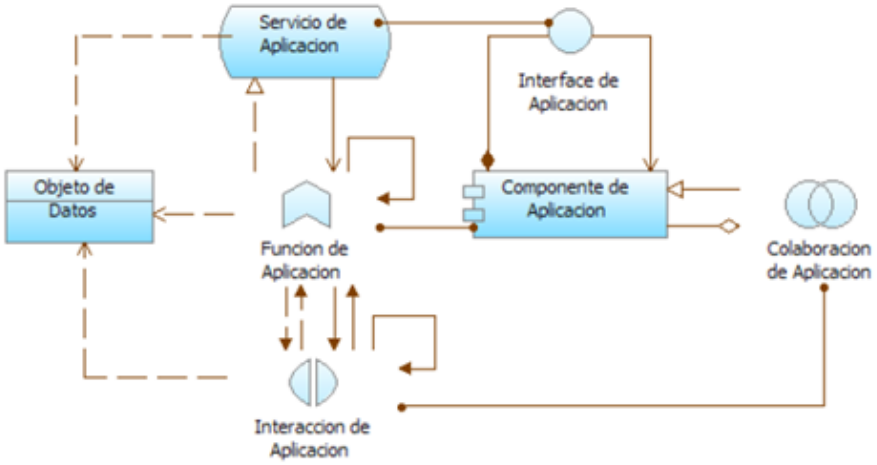


Figura 2.14: Metamodelo comportamiento de aplicación - SJBC, Coloso Archimate

2.2.2 Módelo comportamiento de aplicación

Este modelo muestra el comportamiento actual de las aplicaciones que intervienen en el proceso de pruebas de la compañía, y sus diferentes estados en las fases del ciclo de vida del proyecto.

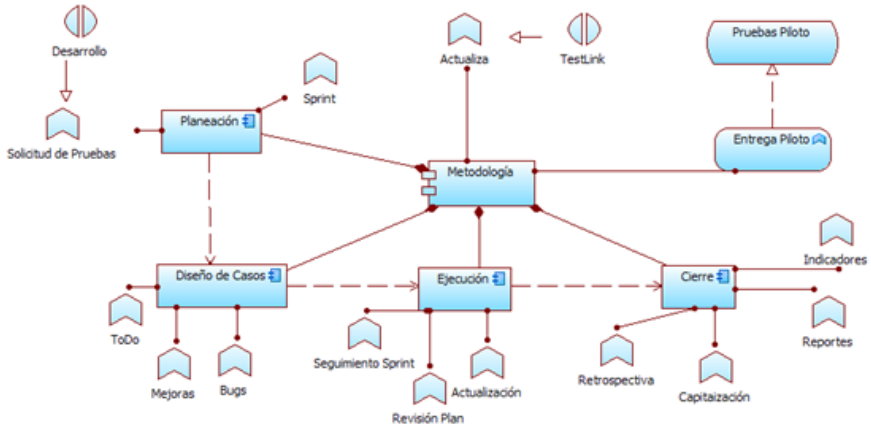


Figura 2.15: Módelo comportamiento de aplicación

2.2.3 Metamodelo Punto de vista de Cooperación de Aplicación

Este modelo muestra la relación que existe entre los componentes de aplicacion en terminos del flujo de informacion entre ellos o los servicios que ofrecen y usan.

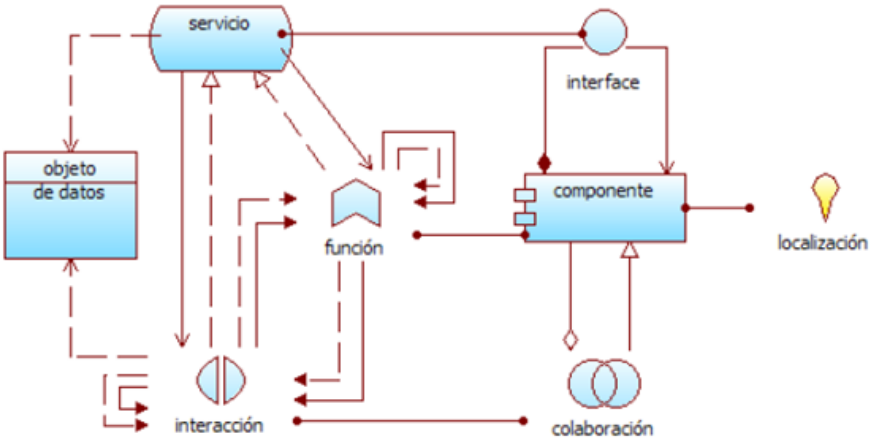


Figura 2.16: Metamodelo Punto de vista de Cooperación de Aplicación - SJBC, Coloso Archimate

2.2.4 Módelo Punto de vista de Cooperación de Aplicación

El proceso propuesto en este trabajo de grado estará disponible en el componente metodología y calificación respectivamente.

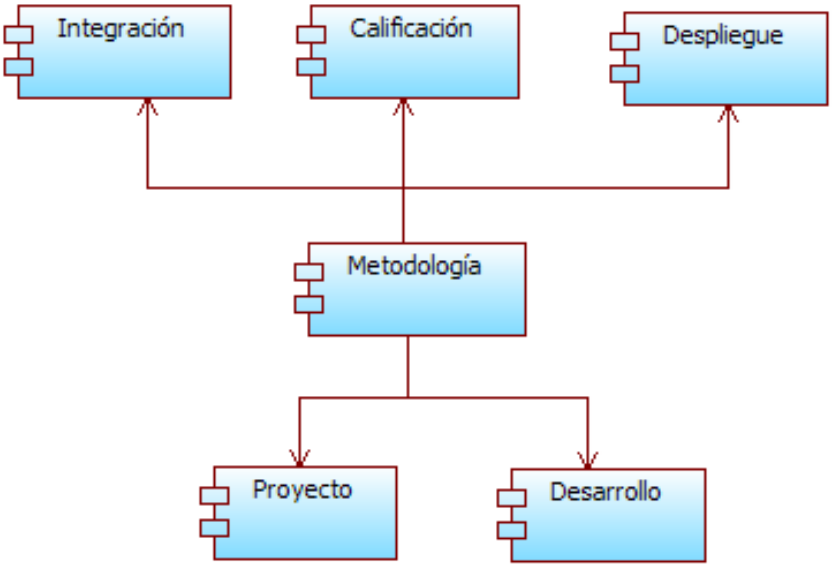


Figura 2.17: Módelo Punto de vista de Cooperación de Aplicación

2.2.5 Metamodelo Punto de vista de Estructura de Aplicación

Este punto de vista muestra la estructura de aplicaciones o componentes, esto es útil para el diseño o entendimiento de la estructura de las principales aplicaciones y componentes con sus respectivos datos asociados.

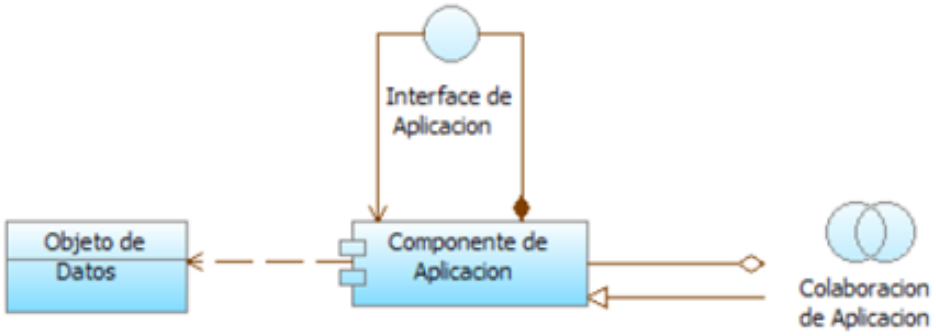


Figura 2.18: Metamodelo Punto de vista de Estructura de Aplicación - SJBC, Coloso Archimate

2.2.6 Módelo Punto de vista de Estructura de Aplicación

Los componentes más importantes en el proceso de pruebas de la compañía son los servicios desarrollo, pruebas e implementación de software.

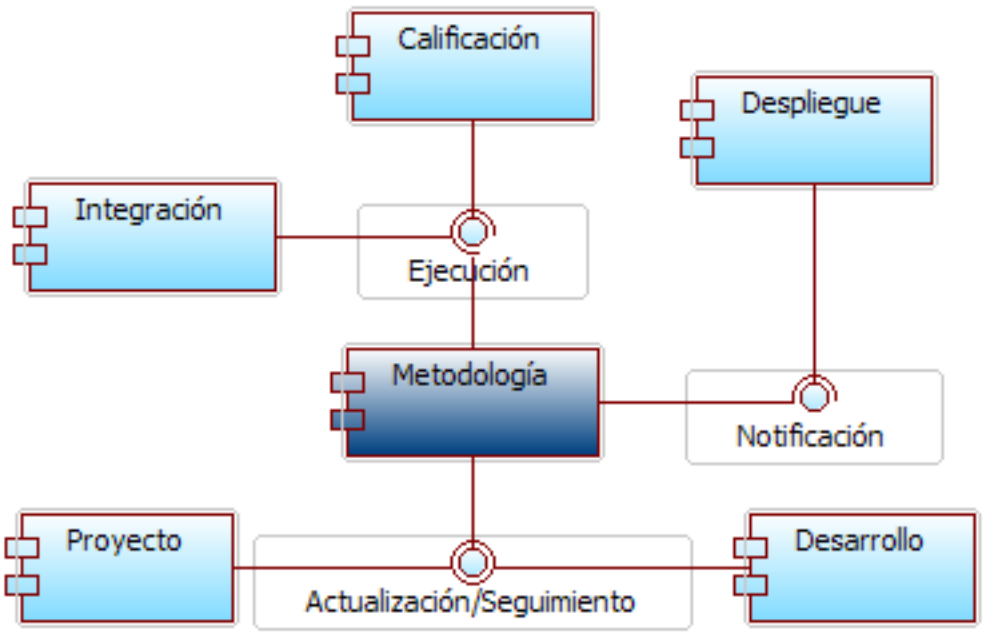


Figura 2.19: Módelo Punto de vista de Estructura de Aplicación - SJBC, Coloso Archimate

2.2.7 Metamodelo Punto de vista de Uso de Aplicación

Este punto de vista describe cómo las aplicaciones son usadas para soportar uno o más procesos de negocio y cómo estás son usadas por otras aplicaciones. Sirve para identificar los servicios que son necesarios para los procesos de negocio y otras apliacciones.

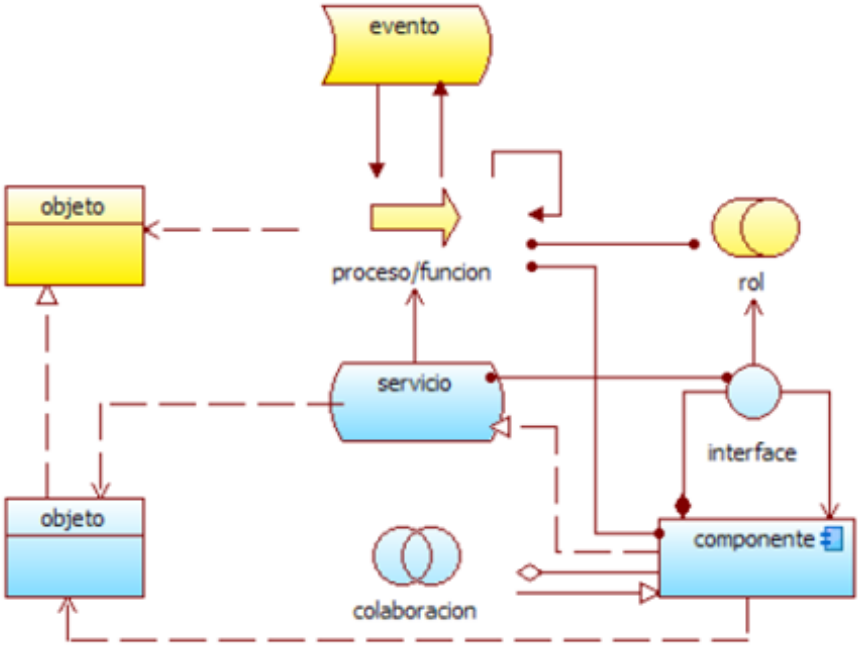


Figura 2.20: Metamodelo Punto de vista de Uso de Aplicación - SJBC, Coloso Archimate

2.2.8 Módelo punto de vista de Uso de Aplicación

En el caso puntual del area de pruebas de software, se tienen los componentes de calificacion e integracion, los cuales son utilizados por los analista de prebas e integradores de producto.

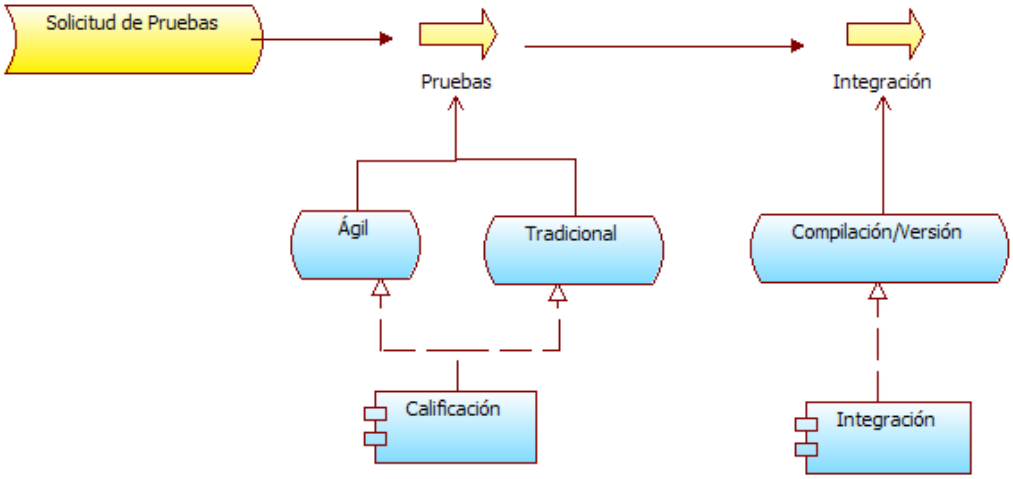


Figura 2.21: Módelo punto de vista de Uso de Aplicación - SJBC, Coloso Archimate

2.3 Capa Infraestructura

2.3.1 Metamodelo Punto de vista de Infraestructura

Este metamodelo muestra los elementos de hardware y software, como lo son dispositivos fisicos, redes, bases de datos y demas.

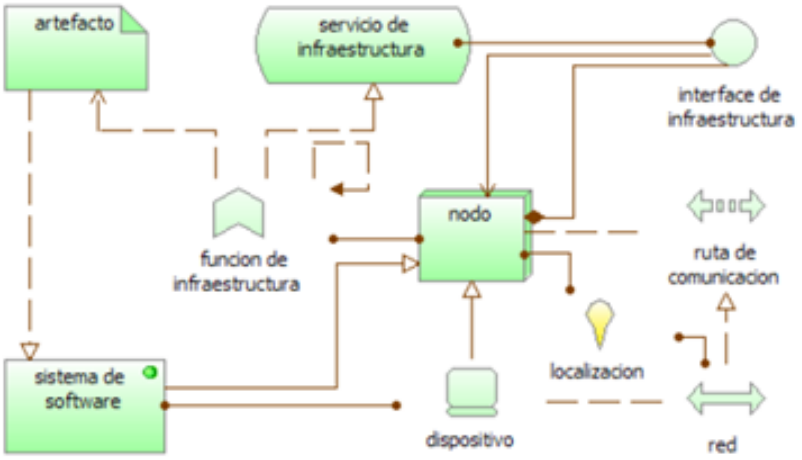


Figura 2.22: Metamodelo Punto de vista de Infraestructura - SJBC, Coloso Archimate

2.3.2 Módelo Punto de vista de Infraestructura

Este modelo muestra la estructura hardware y software de la compañía techrea, en la cual contine sus diversos componentes y aplicaciones y la interconexion ente ellos.

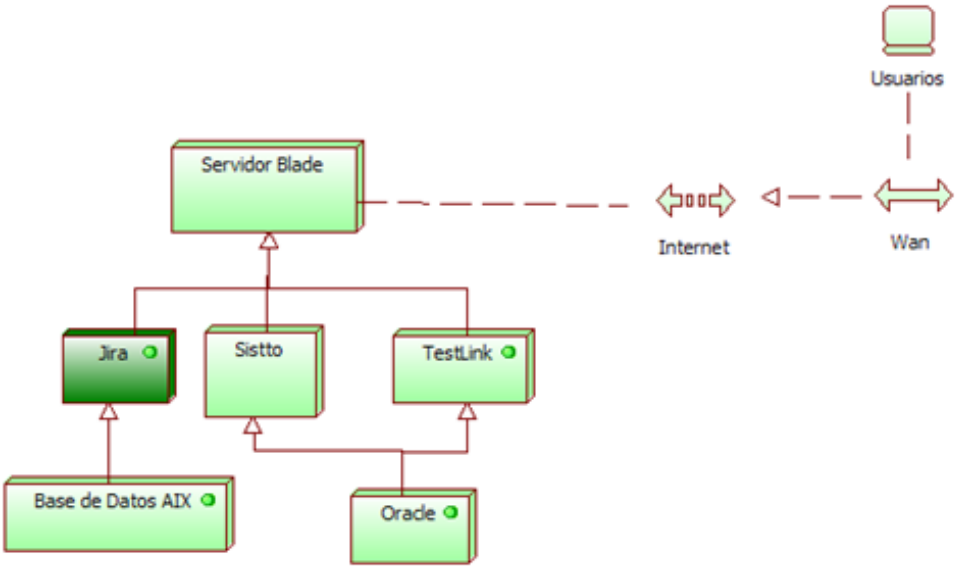


Figura 2.23: Módelo Punto de vista de Infraestructura - SJBC, Coloso Archimate

2.3.3 Metamodelo Punto de vista de Uso de Infraestructura

Este modelo muestra el sistema de software y redes los cuales son entregados a las aplcaciones. Esto muestra el rendimiento y la escalabilidad de la infraetructura, teniendo en cuenta las aplicaciones que harán uso de esta infraestructura.

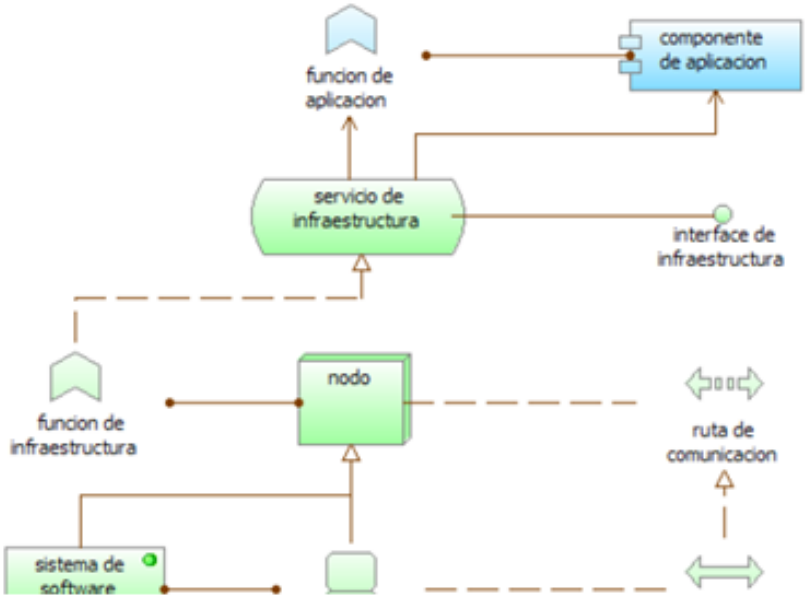


Figura 2.24: Metamodelo Punto de vista de Uso de Infraestructura - SJBC, Coloso Archimate

2.3.4 Módelo punto de vista de Uso de Infraestructura

Este modelo describe tres componentes de aplicación que se usan para prestar el servicio de pruebas de software dentro de la compañía.

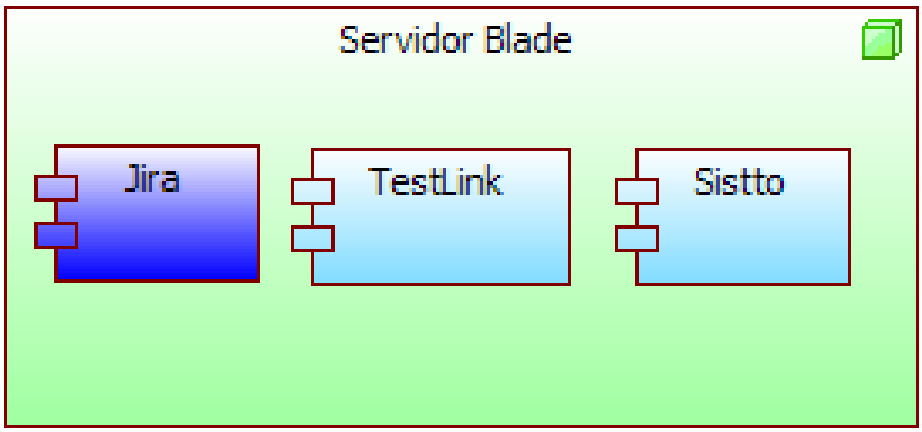


Figura 2.25: Módelo punto de vista de Uso de Infraestructura - SJBC, Coloso Archimate

2.3.5 Metamodelo Punto de vista de Implementación y despliegue

El metamodelo muestra cómo se realiza una aplicación en la infraestructura. Esto comprende el mapeo de aplicaciones y la información usada por estas aplicaciones en modo de artefactos de software y componentes de almacenamiento como bases de datos o diferentes archivos.

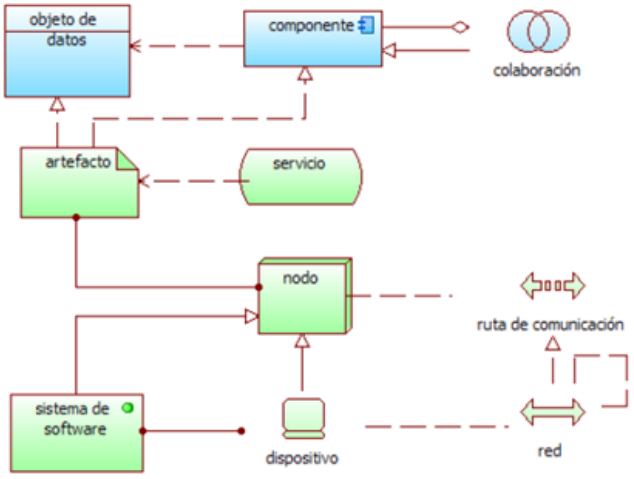


Figura 2.26: Metamodelo Punto de vista de Implementación y despliegue - SJBC, Coloso Archimate

2.3.6 Módelo Punto de vista de Implementación y despliegue

Este modelo describe la interacción de los diferentes componentes de aplicación y la infraestructura que los soporta, en este caso la interacción entre el portal web Jira y la base de datos Oracle, la cual esta instalda en un AIX.

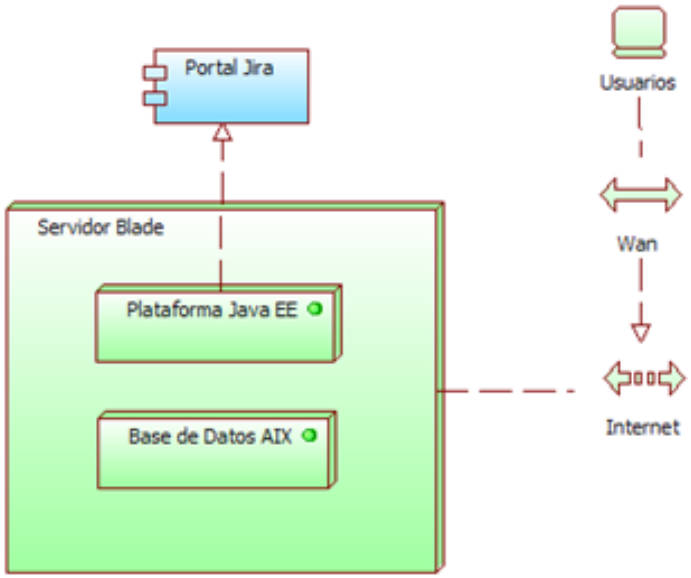


Figura 2.27: Módelo punto de vista de Implementación y despliegue - SJBC, Coloso Archimate

2.3.7 Metamodelo punto de vista de Estructura de Información

El punto de vista de estructura de información el semejante a los modelos de innformación usados en la programación orientada a objetos en donde se dibujan las clases, objetos y demás, muestra cómo la representación de la estructura de los datos de negocio es convertida a una estructura de datos de aplicación.

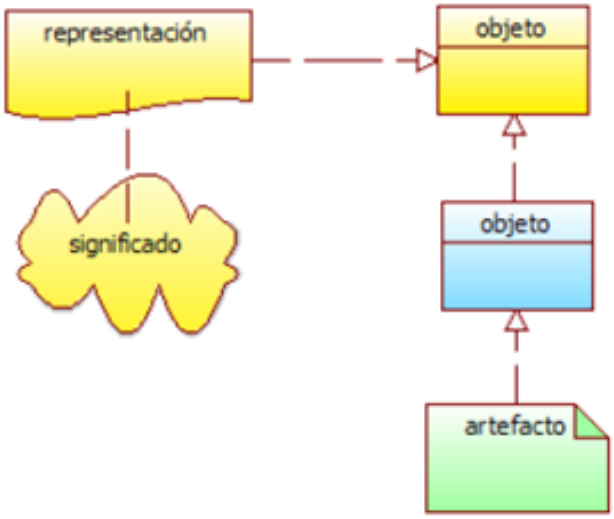


Figura 2.28: Metamodelo punto de vista de Estructura de Información - SJBC, Coloso Archimate

2.3.8 Módelo punto de vista de Estructura de Información

El modelo describe los objetos de información que hace parte de la sección de pruebas de software y su interacción con las diferentes herramientas, al igual se muestra el nuevo PugIn ágil de la herramienta Jira, en el cual se manearan las estructuras y funcionalidades ágiles del proceso de pruebas.

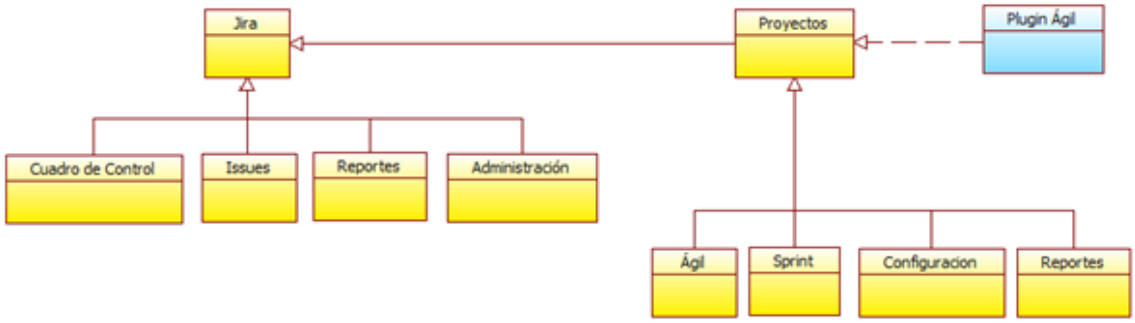


Figura 2.29: Módelo punto de vista de Estructura de Información - SJBC, Coloso Archimate

2.3.9 Metamodelo Punto de vista de Realización del Servicio

Este punto de vista resalta cómo los servicios de negocio son realizados por los procesos implícitos.

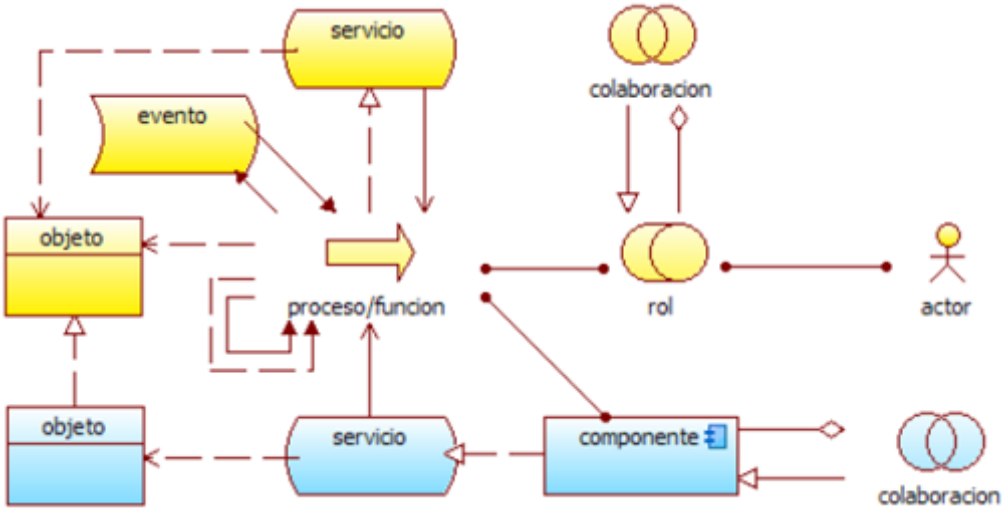


Figura 2.30: Metamodelo Punto de vista de Realización del Servicio - SJBC, Coloso Archimate

2.3.10 Módelo Punto de vista de Realización del Servicio

El modelo muestra como la metodología tiene acceso a las diferentes áreas del proceso de pruebas y las herramientas que este contiene.

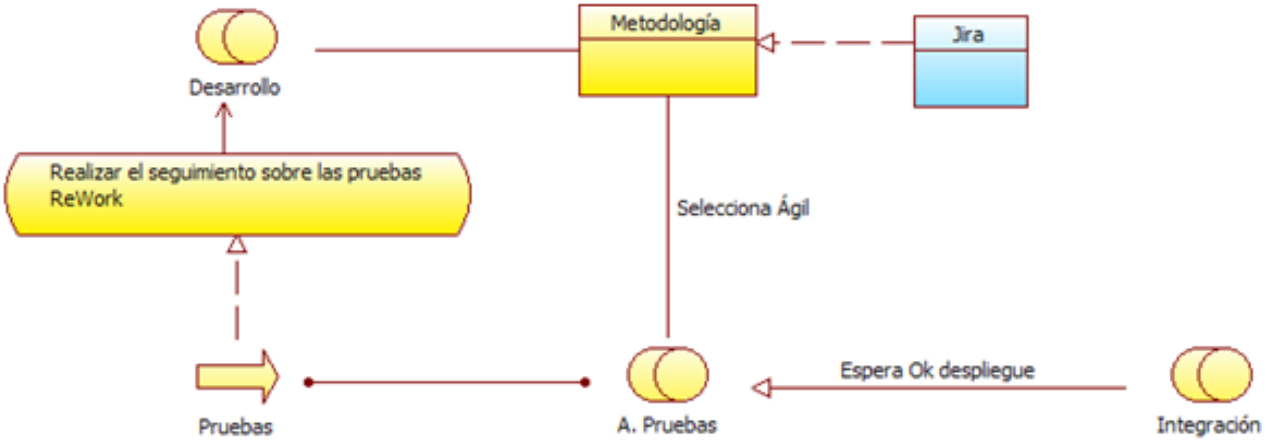


Figura 2.31: Módelo Punto de vista de Realización del Servicio - SJBC, Coloso Archimate

2.3.11 Módelo Punto de vista de Capas

Este modelo muestra diferentes capas de negocio en donde se separan los actores, servicios de negocio, servicios de aplicación y la infraestructura que los soporta. En este caso se muestra la capa de negocio en la cual interactúan las pruebas de software realizadas, la capa de aplicación en la cual interviene la herramienta Jira la cual contiene el plugin ágil, y por último la capa de infraestructura la cual contiene una base de datos Oracle montada en un AIX.

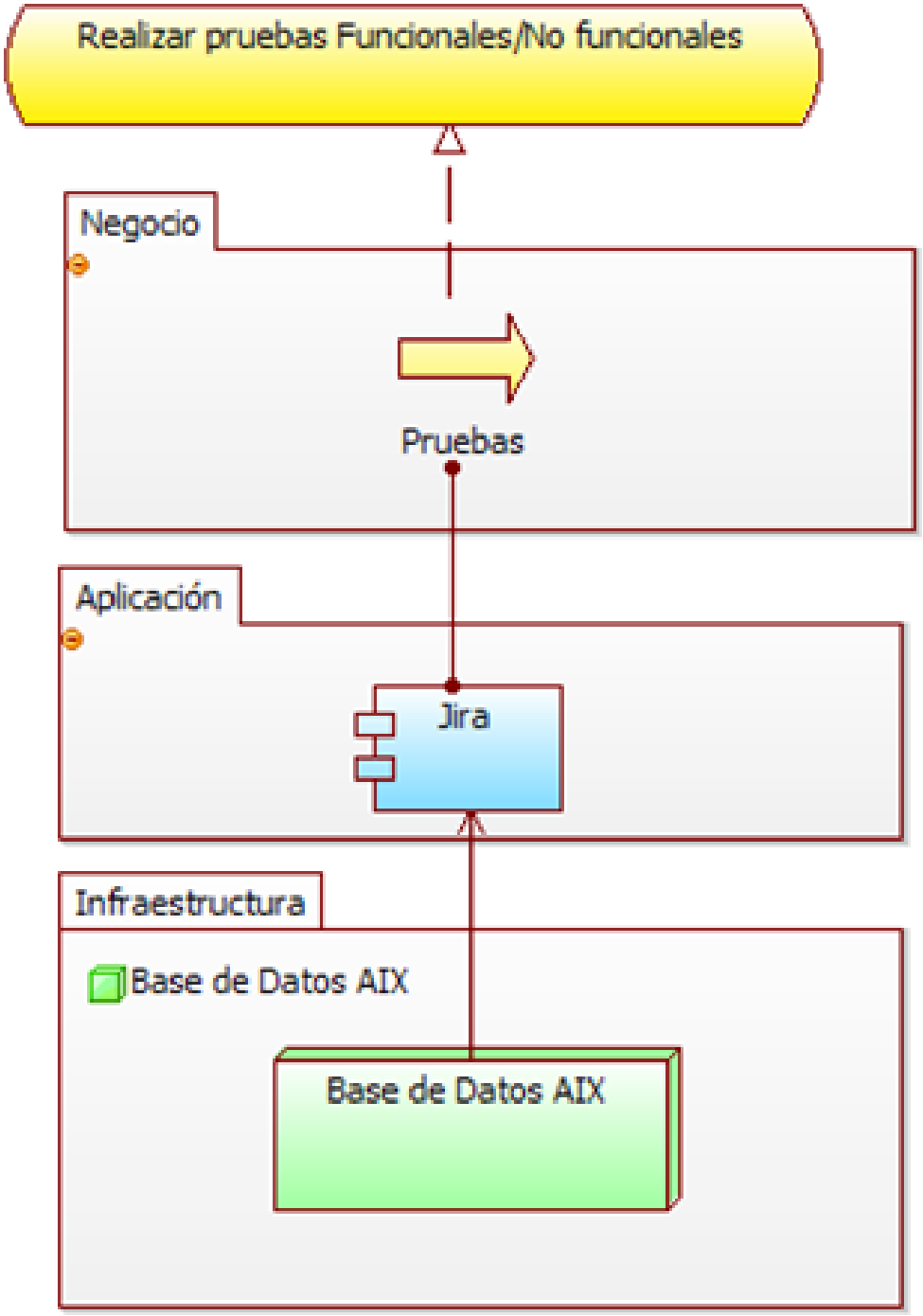


Figura 2.32: Módelo Punto de vista de Capas

2.4 Capa Motivacional

2.4.1 Metamodelo Punto de vista de Implicados

Este metamodelo representa a los implicados, los manejadores para cambio y las valoraciones, siempre en términos de oportunidades y amenazas. Se describe también los enlaces a los objetivos que direccionan las preocupaciones y valoraciones.

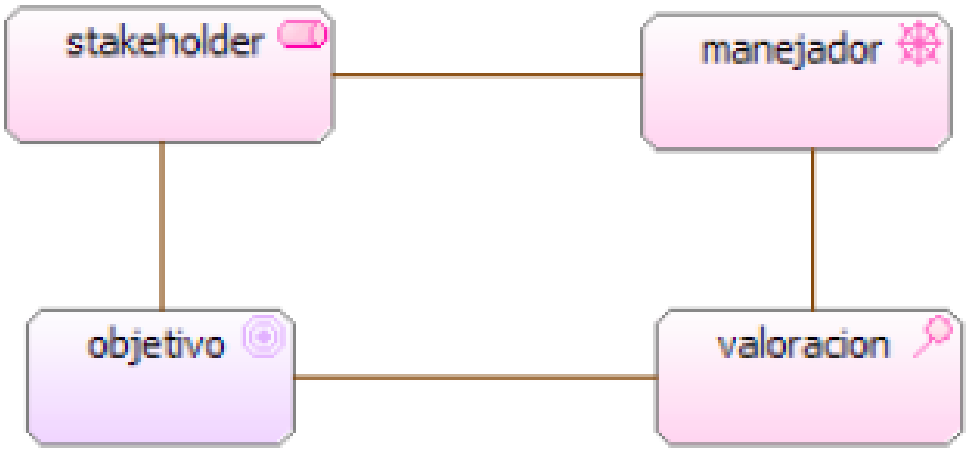


Figure 2.33: Metamodelo Punto de vista de Implicados - SJBC, Coloso Archimate

2.4.2 Módelo Punto de vista de Implicados

Este modelo describe tres implicados dentro del proceso de pruebas de la compañía, el coordinador de pruebas encargado de administrar los recursos y cronogramas de los proyectos, los líderes de pruebas encargados de apoyar y guiar a los testers en la ejecución de las pruebas, y por último los testers encargados de la ejecución de las pruebas y su posterior calificación.

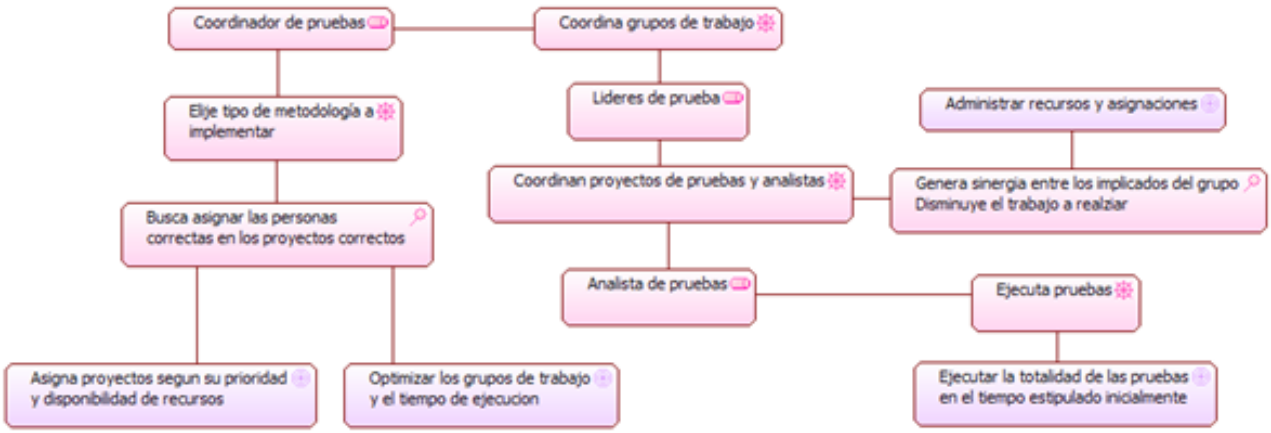


Figura 2.34: Módelo Punto de vista de Implicados - SJBC, Coloso Archimate

2.4.3 Metamodelo Punto de vista de Realización de Objetivos

Permite al diseñador modelar las metas concretas,dentro de requerimientos y restricciones que son necesarias para realizar los objetivos, el refinamiento de objetivos dentro de sub-objetivos es modelado usando la relacion de agregacion.

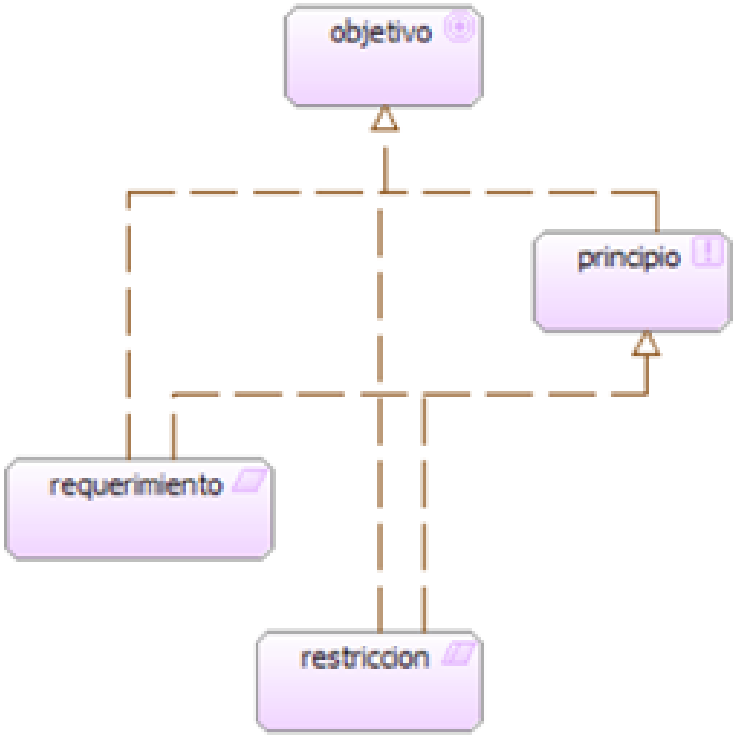


Figura 2.35: Metamodelo Punto de vista de Realización de Objetivos - SJBC, Coloso Archimate

2.4.4 Módelo Punto de vista de Realización de Objetivos

El modelo fija diversos actores involucrados directa o indirectamente dentro del proceso de pruebas de la compañía y sus diversas actividades propias de cada uno.

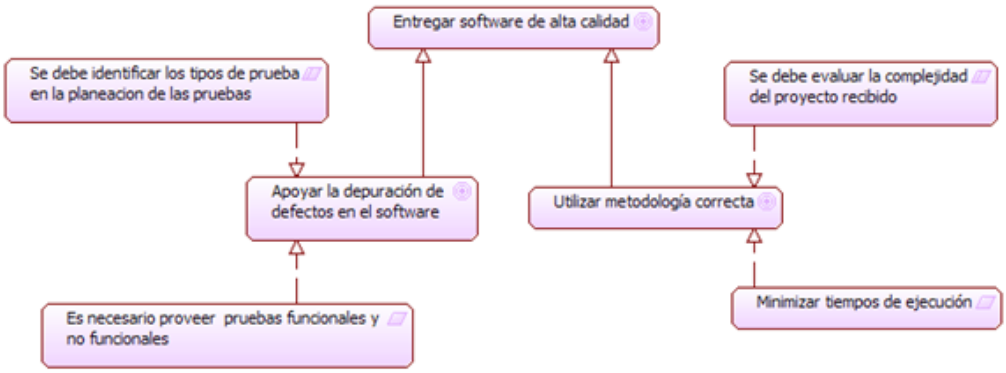


Figura 2.36: Módelo Punto de vista de Realización de Objetivos - SJBC, Coloso Archimate

2.4.5 Metamodelo Punto de vista de Contribución de Objetivos

Este metamodelo permite al diseñador o analista modelar la influencia de las relaciones entre los objetivos y requerimientos. Las vistas resultantes pueden ser usadas para analizar el impacto que los objetivos tienen con otros, o para detectar confligtos entre los objetivos de los implicados.

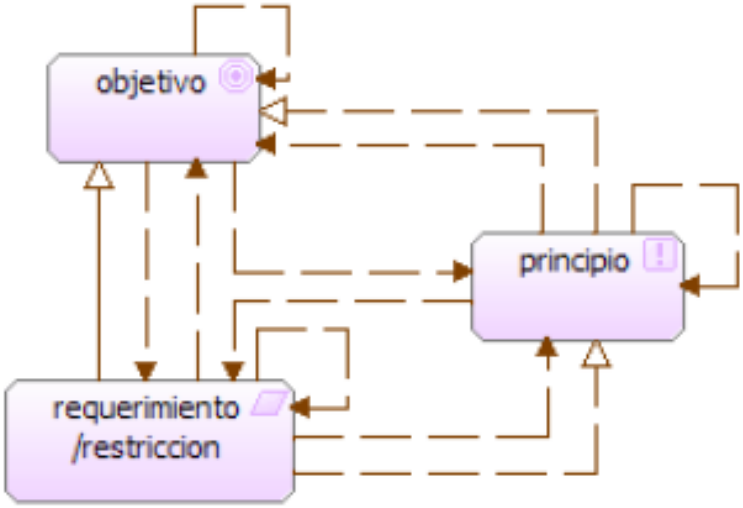


Figura 2.37: Metamodelo Punto de vista de Contribución de Objetivos - SJBC, Coloso Archimate

2.4.6 Módelo Punto de vista de Contribución de Objetivos

El modelo describe el objetivo principal de las pruebas de software, sus principales características y metas como lo son el minimizar los errores encontrados en producción, el entregar un software de alta calidad utilizando la menor cantidad de recursos disponibles para ello, al igual que aportar un dinamismo al área de pruebas con la implementación de un nuevo proceso ágil.

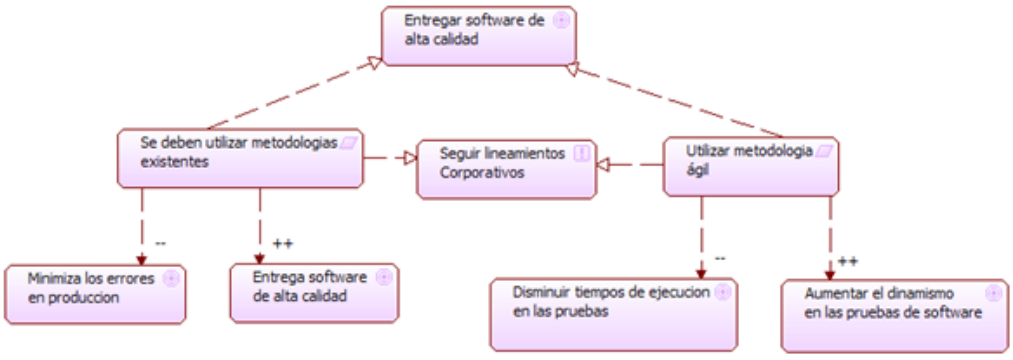


Figura 2.38: Módelo Punto de vista de Contribución de Objetivos - SJBC, Coloso Archimate

2.4.7 Metamodelo Punto de vista de Principios

El punto de vista de principios describe los principios más relevantes en el diseño que se esté planteando incluyendo los objetivos que motivan estos principios, también las relaciones entre los objetivos y los principios.

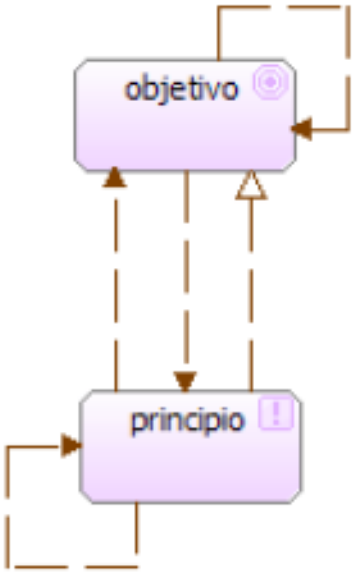


Figura 2.39: Metamodelo Punto de vista de Principios - SJBC, Coloso Archimate

2.4.8 Módelo Punto de vista de Principios

El objetivo principal del área de pruebas de techrea es entregar software de alta calidad, utilizando para ello una metodología estructurada junto con un proceso ágil para el área de pruebas de la compañía.

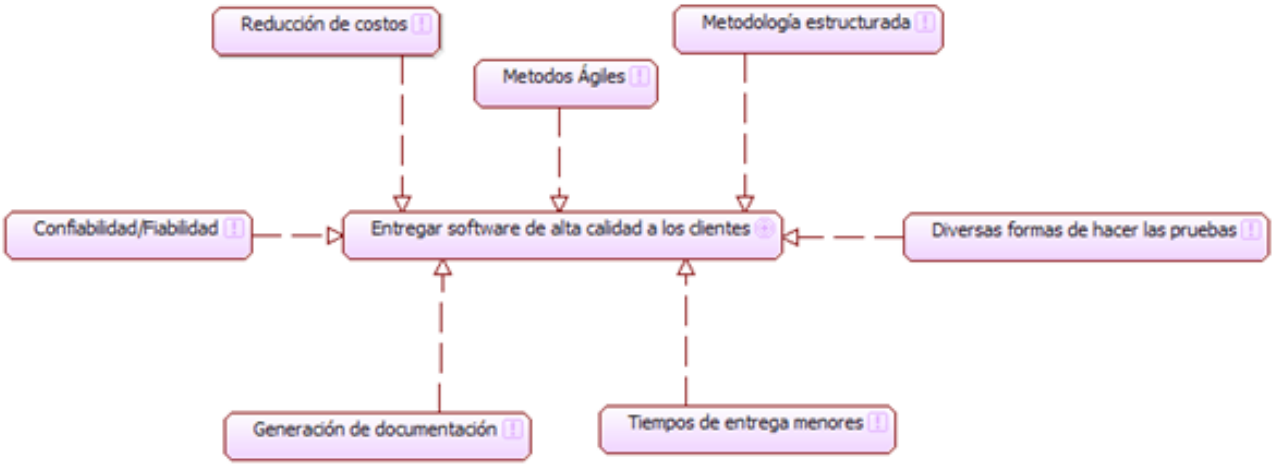


Figura 2.40: Módelo Punto de vista de Principios - SJBC, Coloso Archimate

2.4.9 Metamodelo punto de vista de Realización de Requerimientos

Este metamodelo permite modelar la realización de requerimientos por los elementos del negocio como; actores, servicios de negocio, procesos de negocio, servicios de aplicación, componentes de aplicación, etc.

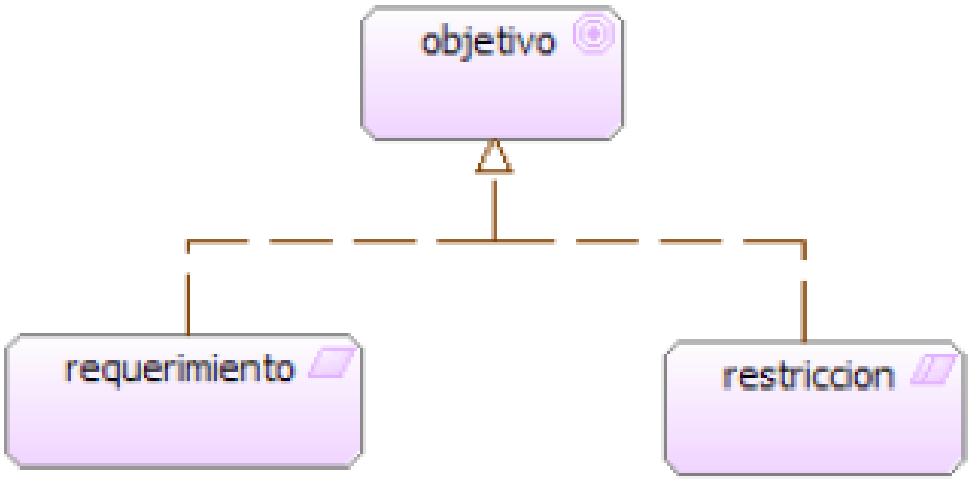


Figura 2.41: Metamodelo punto de vista de Realización de Requerimientos - SJBC, Coloso Archimate

2.4.10 Módelo punto de vista de Realización de Requerimientos

Este modelo muestra la forma en que se relaciona la entrega de software de alta calidad con la utilización de métodos tradicionales y ágiles en pro de satisfacer las necesidades de los clientes.

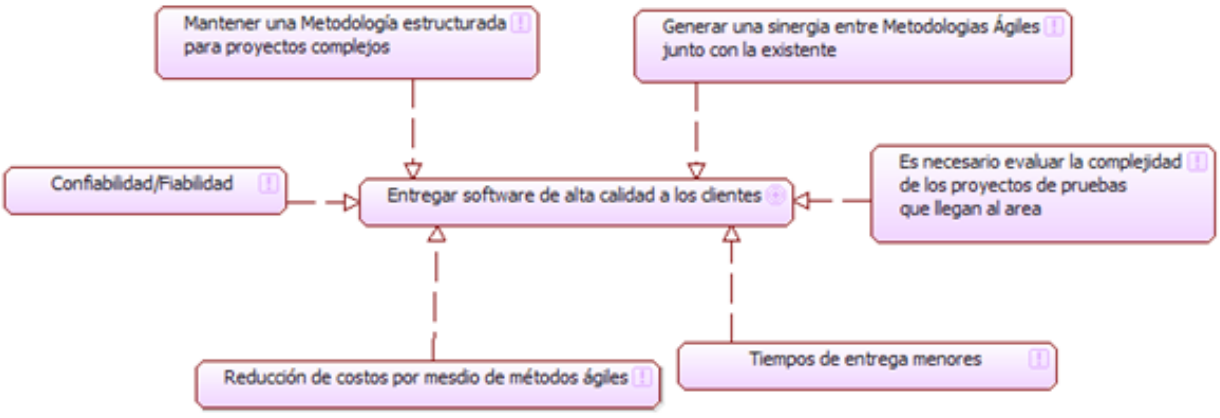


Figura 2.42: Módelo punto de vista de Realización de Requerimientos - SJBC, Coloso Archimate

2.4.11 Metamodelo Punto de vista de Motivación

Este metamodelo muestra el aspecto motivación sin enfocarse en elementos internos, al igual permite ver la relación entre los servicios, actores y aplicaciones.

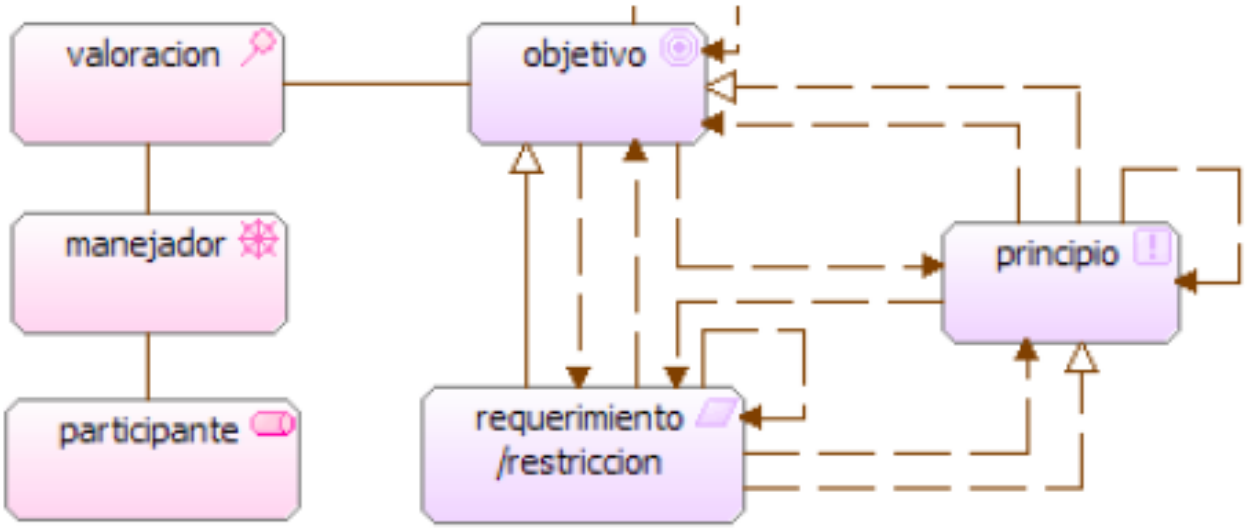


Figura 2.43: Metamodelo Punto de vista de Motivación - SJBC, Coloso Archimate

2.4.12 Módelo Punto de vista de Motivación

En este modelo se presentan requerimientos y el objetivo del área de pruebas, al igual que los actores princiales dentro de cada área del proceso junto con su descripción y objetivos.

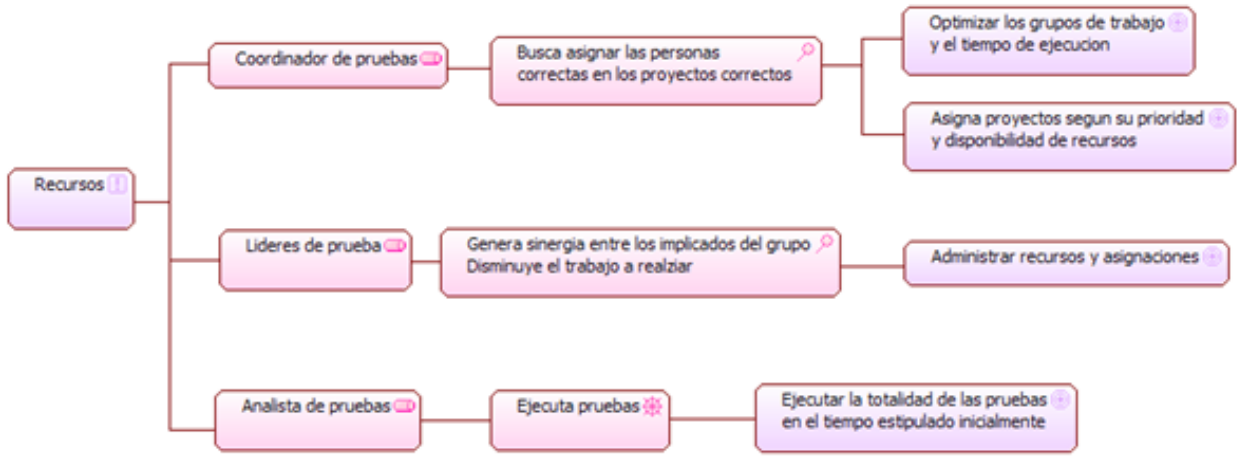


Figura 2.44: Módelo Punto de vista de Motivación - SJBC, Coloso Archimate

Capítulo 3

DESARROLLO DE LA INVESTIGACIÓN

3.1 Situación Actual del proceso del área de pruebas

Para la realización de este trabajo de grado se inició con una revisión de la documentación manejada dentro del área de pruebas de la compañía Techrea S.A.S (Informes de avance, matriz de validación de casos de pruebas, informes finales de los proyectos, manual de procedimientos, políticas del área, estructura organizacional), se realizó una observación a las actividades diarias del grupo de pruebas, en el cual se tomó atenta nota a los procedimientos utilizados, validación de canales de comunicación y demás características implícitas dentro del área. Por último se realizaron entrevistas estructuradas a los miembros de los equipos de pruebas consignadas en el "Anexo 1", y así recabar información relevante para sentar las bases de MEND – TEST.

Dentro de este proceso de análisis del problema, se logró construir un diagrama del procedimiento manejado dentro del área de pruebas, en el cual se visualizan cuatro actores principales dentro del proceso, los cuales son, Coordinador de desarrollo, Coordinador de pruebas, analista de pruebas y testers, a continuación se generalizan dichos roles y su interrelación.

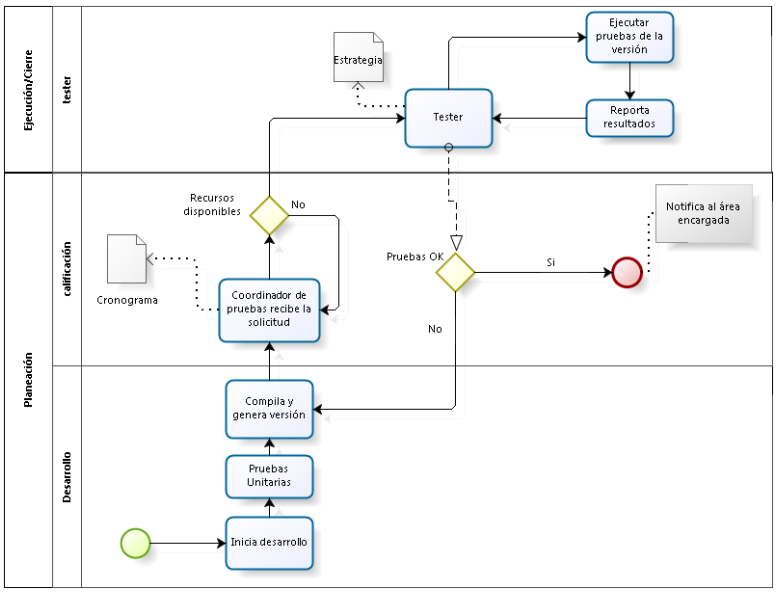


Figura 3.1: Proceso actual Compañía - CEHO - Metodología MEND – TEST

3.1.1 Procedimiento al ingresar un proyecto de pruebas

Luego de la evaluación de la situación actual del modelo manejado por la compañía Techcrea, se realizó un modelo de espina de pescado ilustrado en PRESSMAN, R. S. (2005), en el cual se colocó el problema o efecto en la parte de la cabeza del modelo de espina, y dentro de sus espinas las posibles causas del mismo.

Esto gracias a la colaboración de diversos actores involucrados en el área de pruebas, los cuales por medio de una lluvia de ideas apoyaron en la descripción de las diversas variables que posiblemente están generando el problema.

Con lo anterior se construye el siguiente diagrama de espina de pescado, el cual nos da una visión más acertada en cuanto a la problemática presentada en la compañía.

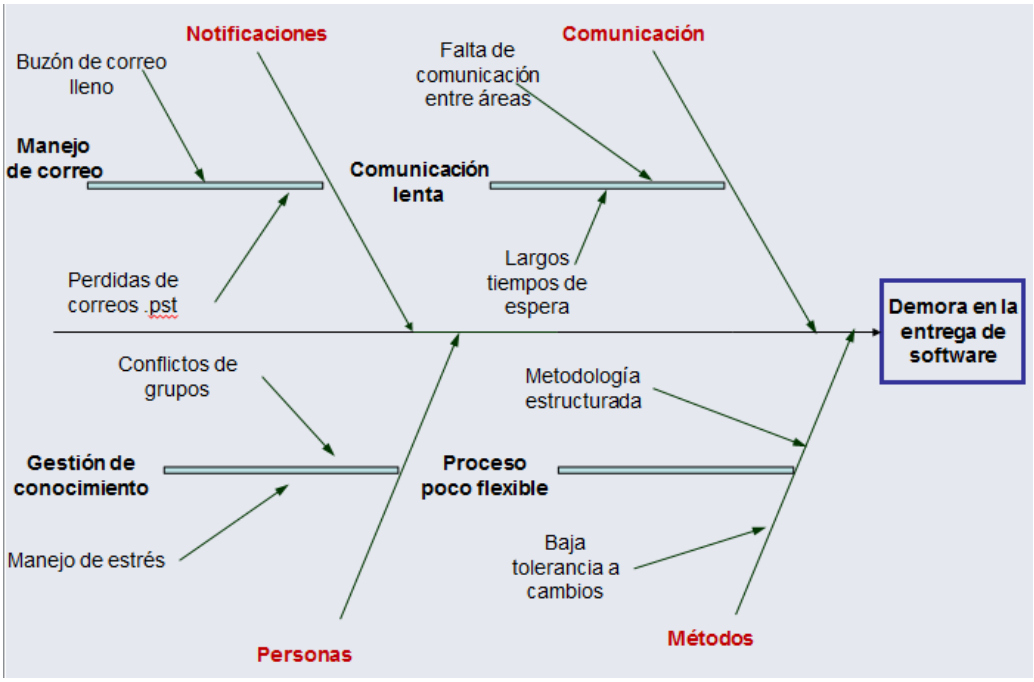


Figura 3.2: Espina de pescado - CEHO - Metodología MEND – TEST

Dentro de las categorías plasmadas en el diagrama de espina de pescado se encuentran las siguientes cuatro:

- Comunicación: Falla en la comunicación entre áreas e inadecuada utilización de canales de comunicación
- Notificaciones: Perdida de correo interno .pst, fallos en el servidor de correos, buzones llenos.
- Gestión de conocimiento: Conflictos de grupos y manejo de estrés.
- Métodos: Baja tolerancia a cambios dentro de los procedimientos a seguir durante las pruebas.
- Efecto: Demora en la entrega del software (La compañía genera retardos en cada etapa de los procesos a la hora de certificar el software, representados en tiempos de ejecución).

3.2 Metodología Mend-Test

En este capítulo se presenta todo lo relacionado y definido para MEND – TEST, metodologia de pruebas funcionales, su definición de actividades, artefactos, herramientas y roles, la cual servirá de guía para realizar pruebas de software de manera rápida y ágil. Dentro de sus principales características se encuentra el permitir realizar pruebas de manera ágil.

Históricamente las metodologías guían los procesos y actividades, tanto en el ámbito académico, como en la industrial en general, es decir están inmersas en cualquier área del conocimiento. Con base en diversas asesorías de expertos en temas relacionados con ingeniería de sistemas y software y reflexiones personales, he llegado a proponer en este trabajo de grado una metodología llamada MEND – TEST aplicable al área de pruebas de la compañía Techrea en las fases de planeación, diseño, ejecución y cierre, la cual aportará en la optimización del proceso de pruebas y el asegurar al máximo la calidad de los entregables.

3.3 Mend-Test

Los procesos y artefactos descritos en esta metodología abarcan desde el inicio del ciclo de pruebas hasta su puesta en producción, especificamente en las fases de planeación, diseño, ejecución y cierre.

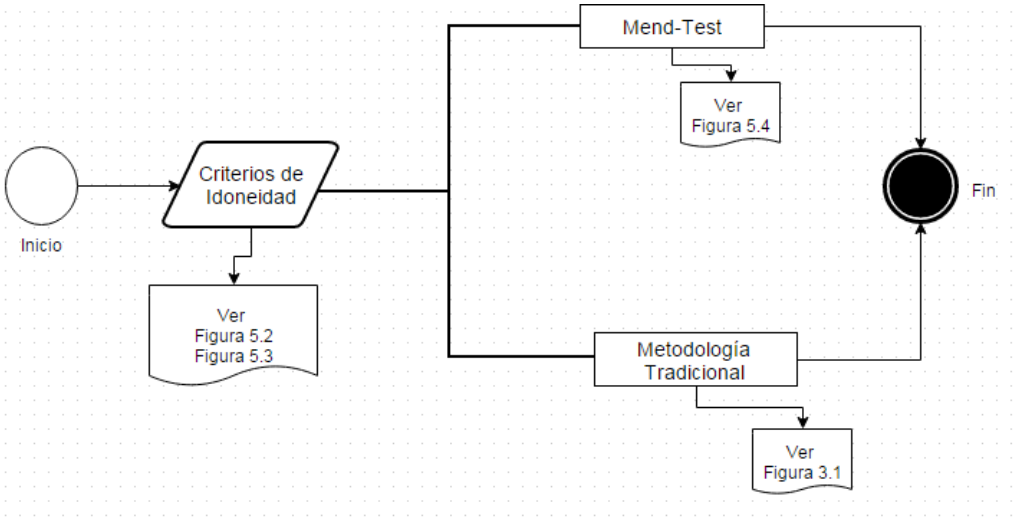


Figura 3.3: Criterios de selección - CEHO - Metodología MEND – TEST

Ingreso de un proyecto: Es el producto de etapas previas de las áreas de comercial, requerimientos y proyectos.

Criterios de idoneidad: Se realiza la evaluación con un cuadro comparativo guiado por lo especificado en los criterios de idoneidad para un proyecto según (Juan Palacio, 2008), adicional se le asignará un valor a cada uno de estos criterios, el cual posteriormente ayudará a tomar la decisión de que enfoque utilizar.

Se debe tener en cuenta las necesidades del proyecto, evaluando su robustez y complejidad, para elegir así la metodología a usar, en busca de obtener éxitos tempranos.

Valor: Comprendido entre 1 y 4 donde 1 es el valor más bajo y 4 el más alto, se sdebe tener en

cuenta que estos criterios no pueden ser el mismo, es decir no se puede indicar que el tamaño del sistema es 4 para el criterio “grande” y 4 para el criterio “pequeño”, siempre uno de ellos debe ser mayor que su competidor.

El valor final dado para cada criterio de idoneidad estará regido por la diferencia entre A y P, donde A es “Adaptable” y P es “Predictivo”. Este resultado seleccionará aquel criterio con mayor puntaje de la iguiente manera.

$$\varepsilon A > \varepsilon P = A \tag{3.1}$$

$$\varepsilon A < \varepsilon P = P \tag{3.2}$$

En donde el resultado es el siguiente:

A = Implementar Metodología MEND – TEST

P = Implementar Metodología Actual

Ejemplo de una implementación con la metodología MEND – TEST

Necesidad	Criterio	Valor	Enfoque	Adaptativo	Predictivo
Prioridad del negocio	Valor	4	A	1	0
	Cumplimiento	3			
Tamaño del sistema	Grande	4	A	1	0
	Pequeño	3			
Coste del prototipado	Bajo	4	A	1	0
	Alto	3			
Criticidad del sistema	Alta	4	A	1	0
	Baja	1			
Rigidez del producto	Dificil de modificar	0	P	0	1
	Modificable	1			
Estabilidad de requisitos	Entorno estable	0	P	0	1
	Entorno inestable	4			
Implementar Metodología MEND – TEST					

Figura 3.4: Implementación con la metodología MEND – TEST - CEHO

Ejemplo de una implemetacion con la metodología actual

Necesidad	Criterio	Valor	Enfoque	Adaptativo	Predictivo
Prioridad del negocio	Valor	4	A	1	0
	Cumplimiento	3			
Tamaño del sistema	Grande	2	P	0	1
	Pequeño	4			
Coste del prototipado	Bajo	4	P	0	1
	Alto	4			
Criticidad del sistema	Alta	4	A	1	0
	Baja	1			
Rigidez del producto	Difícil de modificar	0	P	0	1
	Modificable	1			
Estabilidad de requisitos	Entorno estable	0	P	0	1
	Entorno inestable	4			
Implementar Metodología Actual					

Figura 3.5: Implementación con la metodología MEND – TEST - CEHO

El corazón de esta metodología se centra en tres aspectos principales los cuales de describen a continuación:

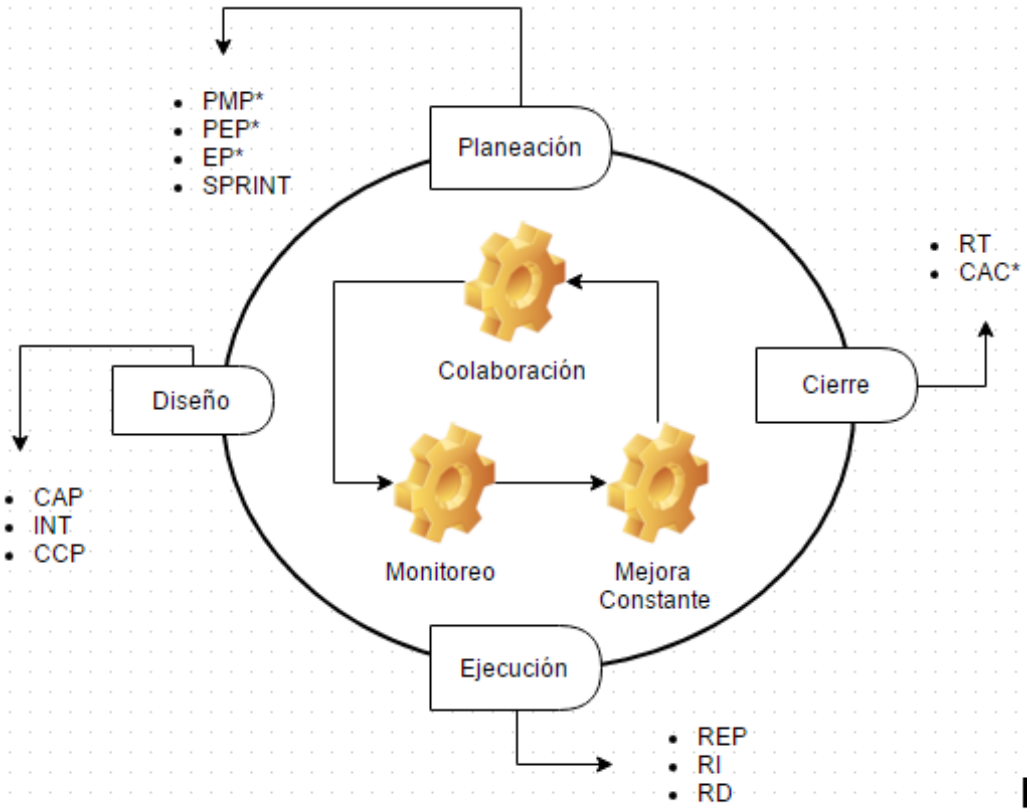


Figura 3.6: Metodología MEND – TEST - CEHO

Colaboración: Se refiere a la colaboración no solo entre los integrantes de un equipo de trabajo, sino también a las áreas de trabajo, integrando proyectos, desarrollo, requerimientos.

Monitoreo: El monitorear un estado actual da una visión de lo que podremos llegar hacer en una línea de tiempo determinada, junto con el control forman una dupla capaz de cambiar el rumbo de los proyectos de pruebas.

Mejora constante: A medida que se avanza en los Sprint y entregas se está implícitamente mejorando el producto por medio de retroalimentaciones realizadas por el grupo de trabajo y por el cliente.

3.3.1 Fases planteadas (Planeación- Diseño - Ejecución - Cierre)

Las siguientes fases son las que guiarán el proceso de pruebas y estarán implícitas en cada sprint creado, se repetirán tantas veces como sea necesario.

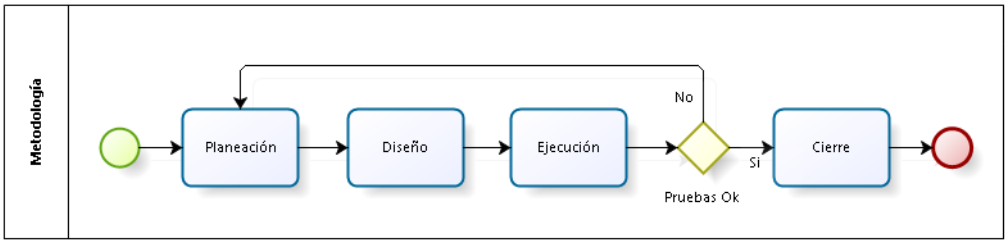


Figura 3.7: Fases de la Metodología MEND – TEST - CEHO

El objetivo de cada fase es:

Planeación: EL objetivo de esta fase es realizar la planeación general del proyecto de pruebas, se realizara el plan de pruebas especifico el cual definen los tipos de pruebas a ejecutar como lo pueden ser, pruebas funcionales, pruebas no funcionales, pruebas de seguridad, posterior, junto con tiempos, recursos, equipos y demás artefactos necesarios para la ejecución de las pruebas.

- PMP =Plan maestro de pruebas *
- PEP= Plan especifico de pruebas *
- EP= Estrategia de pruebas *
- Sprint = Periodo de tiempo entre 1 y 4 semanas

Diseño: En esta fase se realiza la configuración del ambiente de pruebas, se construyen los casos de prueba con sus respectivos datos, en Jira se construyen los ToDo “Los que hacer” y posteriormente se crean los Sprint por cada entrega de desarrollo, desarrollo podrá enviar una o n versiones las cuales al final representaran una versión final, lo anterior dependiendo de desarrollo.

- CAP: Configuración ambiente de pruebas
- INT: Integración de a solución
- CP: Creación casos de pruebas

Ejecución: En esta fase se da inicio al Sprint y se inicia la ejecución de las pruebas previamente planeadas, se toma evidencia de ejecución y se reportan las incidencias de los defectos encontrados tipo Bug.

- REP= Reporte de Estado Proyecto
- RI= Reporte de Incidentes
- RD= Reuniones diarias

Cierre: Al finalizar se realiza un informe de pruebas, donde se evidencia cuantas pruebas fueron ejecutadas, las que no y su respectivo informe de reporte de defectos, al realizar retrospectivas se está documentando y socializando que se hizo bien y mal, que inconvenientes se presentaron en los ciclos de pruebas, que se puede mejorar para el próximo sprint, y las cosas que definitivamente se tienen que dejar de hacer.

- CAC: Cierre Aseguramiento calidad *
- RT: Retrospectiva

Nota: Los marcados con el símbolo " * ", Son aquellos artefactos que solo se producen una única vez. Luego de iniciado el siglo en lo posible no variaran su estructura, a no ser que las circunstancias lo requieran. Para los proyectos que se trabajen con la metodología MEND – TEST iniciaran por la construcción de software, en la cual desarrollo fragmentara sus entregas con versiones RC las cuales iniciaran el ciclo de pruebas.

Para el manejo de pruebas ágiles se trabajara con la herramienta Jira la cual contiene un plugIn para la funcionalidad ágil. En este proceso se crearan actividades marcadas como ToDo las cuales representaran cada una de las tareas necesarias para la ejecución de las pruebas.

Estas actividades ToDo son almacenadas en la pila del producto, una vez creadas las actividades tipo ToDo se priorizan para conformar un Sprint. Estos Sprint serán creados con un tiempo determinado, según lo estipulado en la planeación inicial, una vez es ejecuado el sprint se realizará una retrospectiva de la ejecución, en la cual se consignan las cosas que se hicieron bien y las que se pueden mejorar, con el fin de potenciar los siguientes sprint y no cometer las mismas fallas que en sus sucesores.

Este ciclo se repite cuantas funcionalides entregadas por desarrollo se tengan, y finaliza hasta que se obtenga una entrega completa para ser desplegada en producción. La característica principal de la metodología mend – test es que puede ser adaptada a cualquier tipo de organización, proporcionando un conjunto de procedimientos, plantillas, y herramientas de fácil implementación.

3.3.2 Características

MEND – TEST permite generar un dinamismo en las pruebas de software de la compañía TeChera S.A.S, ya que será utilizado de manera paralela a una metodología ya existente dentro de la compañía, surge de la necesidad de entregar funcionalidades software relativamente pequeñas en tiempos cortos sin alterar su calidad.

3.3.3 Proceso guiado por ciclos "Sprint"

Así como en el desarrollo de software ágil, cada versión entregada representa una funcionalidad que aporta valor al cliente, en las pruebas de software es similar, con la diferencia que en cada sprint creado se agruparán una o varias funcionalidades, las cuales al final aportarán a la entrega de una única versión calificada y aprobada para el cliente.

Actualmente el manejo de las versiones entregadas por desarrollo se basa en la entrega de una

única versión hacia el área de pruebas y posterior a esta entrega realizar la depuración de defectos, corregirlos y generar una nueva versión del producto.

Desarrollo manejará cierto número de Sprint para subdividir las actividades netas de una versión, en la cual empaquetarán el desarrollo de la solución, pruebas unitarias, descarga de artefactos, merge de código fuente, compilación y generación de la versión RC. El manejo de las versiones RC entregadas por desarrollo, se entregarán varias versiones RC según necesidades hacia el área de pruebas y posterior a esta se realizará la depuración de defectos, su corrección y generación de una nueva versión del producto.

A continuación se muestra el diagrama generado para la nueva metodología en paralelo para las pruebas de software.

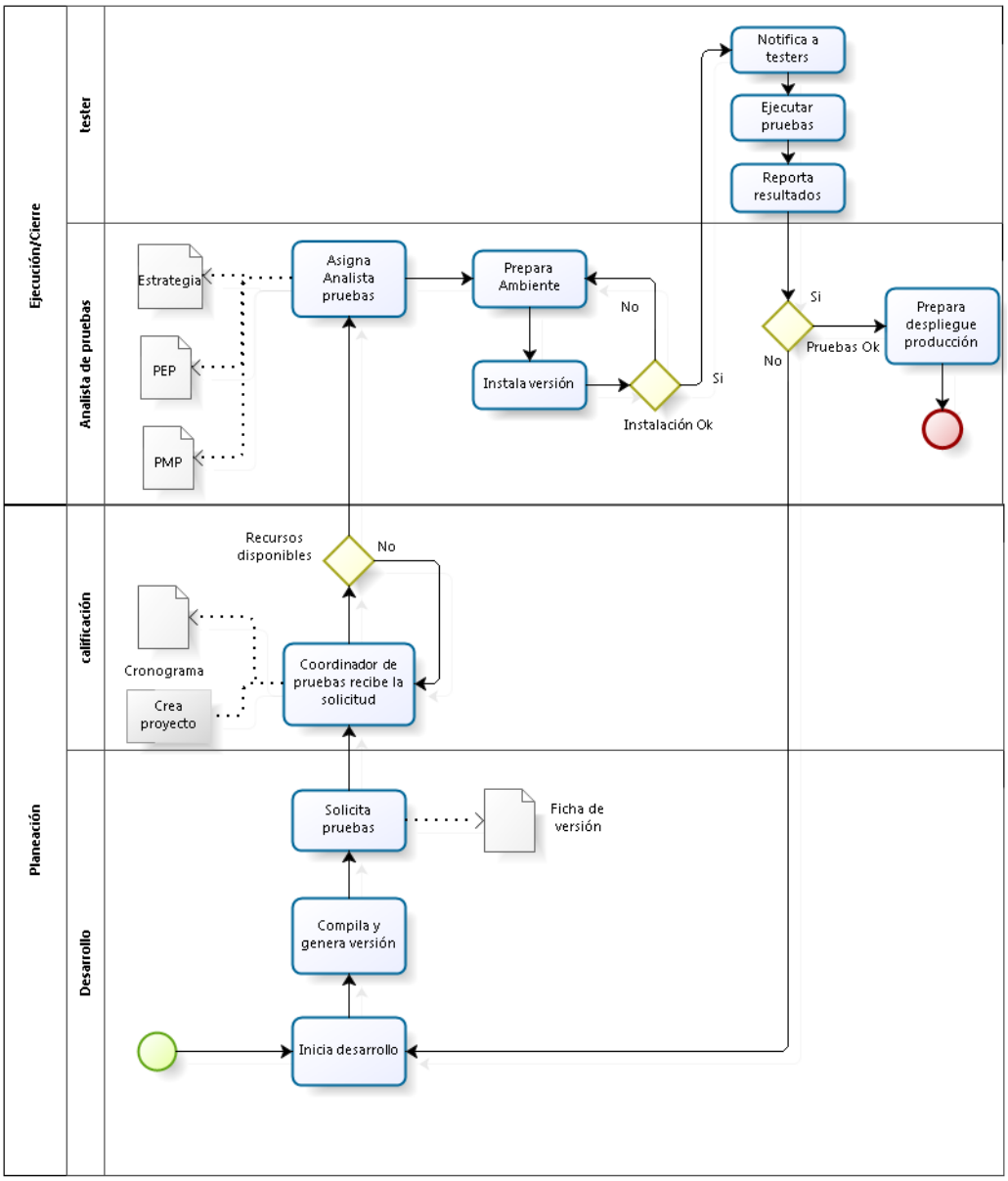


Figura 3.8: Proceso de la nueva Metodología MEND – TEST - CEHO

3.3.4 Roles

Estos roles serán necesarios dentro de los equipos de trabajo Scrum.

Product owner: Responsable de la visión del producto y la gestión económica del proyecto, conecta equipos desarrollo, pruebas, cliente decidiendo que características y funcionalidades ha de tener el sistema y el orden en que se deben implementar en el "Product backlog".

Scrum Master: Es un facilitador, ayuda a aplicar los principios valores y prácticas de scrum, ayuda en la organización para la adopción del proceso.

Equipo auto organizado: Conocimiento global y en diversas tecnologías.

3.3.5 Ventajas de la metodología mend-test

- Permite la toma de decisiones tempranas a medida que se avanza en las pruebas.
- Facilita estar al tanto de lo que pasa en los proyectos
- Muestra resultados a corto plazo
- Aumenta la calidad en las pruebas realizadas
- Permite el monitoreo y control de las actividades propias de las pruebas
- Mayor productividad y eficiencia de las actividades de pruebas
- Se mejoran estimaciones con el tiempo, se van ajustando a medida que se cierran los sprint

3.3.6 Desventajas de la metodología mend-test

- Cambio radical de mentalidad entre los involucrados "Resistencia al cambio"
- Capacitaciones en métodos ágiles "Scrum"
- Implementación de roles específicos dentro del área.
- Exige una madurez de los equipos de trabajo tanto técnica como social.
- Curva de aprendizaje alta, alto nivel de disciplina

Capítulo 4

CIERRE DE LA INVESTIGACIÓN

4.1 Conclusiones

Gracias a la investigación realizada y a la asesoria de expertos, se logro construir una metodología para el área de pruebas, la cual por medio de procesos testing, estructura el trabajo en los proyectos de pruebas de la compañía, proporcionando una visión transversal del estado de las pruebas ejecutadas.

Mediante la implementación de la metodología MEND – TEST, se brindó a la empresa un camino eficaz y funcional para cada iteración del proceso, la cual aportará trazabilidad y seguimiento a los proyectos del área de pruebas de software en las fases de planeación, diseño, ejecución y cierre.

Los grupos de trabajo que conforman el área han recibido con agrado la propuesta de la nueva metodología, ya que les brindara un apoyo a su tarea diaria, permitiéndoles estar al tanto de los acontecimientos y estar en continua comunicación con el área de desarrollo, resolviendo dudas y concertando los diversos tipos de incidentes reportados.

Se realizó el desarrollo de un proceso basado en técnicas ágiles, y su posterior implementación en un proyecto piloto de la compañía, en el cual se pudo observar una optimización de actividades.

El presente trabajo cumplió con las expectativas planeadas inicialmente, aportando a la empresa Techrea S.A.S dinamismo en el área de pruebas y una visión clara y global a la hora de ejecutar pruebas, junto con el establecimiento de una sinergia entre el grupo de pruebas y el área de desarrollo, lo cual aporto a la mejora de la calidad en los productos entregados.

4.2 Recomendaciones

Se recomienda realizar reuniones tempranas entre los equipos de trabajo, en especial área de desarrollo y área de pruebas, para concertar y establecer cuáles de los proyectos pueden ser o no aplicables para la utilización del proceso de MEND – TEST.

El proceso nominal que lleva la compañía no se debe acomodarse al proceso generado por MEND – TEST y la herramienta Jira, por el contrario estos son los que se debe acoplar a ellos, por lo cual es recomendable tomar atenta nota a las anomalías o mejoras en los procesos que mediante el uso puedan surgir.

4.3 Trabajos Futuros

La primera línea de continuación de este trabajo de investigación es el la implementación de un desarrollo guiado por pruebas o Test-driven development (TDD) involucrado las pruebas primero (Test First Development) y el (Refactoring).

La segunda línea de continuación de este trabajo de investigación es es la continuación directa de esta Tesis, Se podría continuar con la aplicabilidad he implementación del nuevo proceso creado en otras compañias, en las cuales se maneje un entorno de pruebas similar al descrito en esta tesis.

4.4 Bibliografía

Boehm, B. (1981). Software Engineering Economics (Prentice-Hall Advances in Computing Science & Technology Series). Mishawaka U.S.A, Prentice Hall PTR.

CIBELLI, C. G. (2012). PHP programación web avanzada para profesionales. Buenos aires: Alfaomega grupo Editor argentino.

Glenford, J. M. (2004). The art of software testing. Word Association John Wiley & Sons, Canada: Inc. GOMEZ, L. GARCÍA, P. DOTES, M. CASTILLO, J. PESA, A. (1993). Diseño y gestión de sistemas de bases de datos. España: Paraninfo, S.A.

MACDONALD, M. (2010). Creación y diseño Web. España: Ediciones Anaya multimedia.

PALACIO, J. (2008). Scrum Manager: Gestión de proyectos. México, SafeCreative.

PRESSMAN, R. S. (2002). Ingeniería del software: Un enfoque práctico. Quinta edición. México D.F, McGraw-Hill Interamericana.

PRESSMAN, R. S. (2005). Ingeniería del software: Un enfoque práctico. Sexta edición. México D.F, McGraw-Hill Interamericana. ROLDÁN, D. VALDERAS, P. LÓPEZ, O. (2010). Aplicaciones Web: Un enfoque práctico. México: Alfaomega grupo Editor,S.A.

SÁNCHEZ, S. A. URBÁN, M. A. GARCÍA, D. (2012). Ingeniería del software: Un enfoque desde la guía SWEBOK. Madrid España: Alfaomega grupo Editor,S.A.de C.V.,México.

HIGHSMITH, James. (2001). Requirements Engineering and agile Software development. EEUU: Addison Wesley.

PALACIO. Juan. (2008). SrumManager Gestión de proyectos. España: Paraninfo, S.A.

4.5 Infografía

The International Software Testing Standard, (2013). ISO/IEC/IEEE 29119 Software Testing. Recuperado el 02 de septiembre de 2015, del sitio Web: http://www.softwaretestingstandard.org/

Federación colombiana de la industria del software y tecnologías informáticas relacionadas (2014). Recuperado el 01 de octubre de 2015, del sitio Web: http://fedesoft.org/

Standish Group International (2015) informe chaos, proyectos exitosos. Recuperado el 15 de Noviembre de 2015, del sitio Web: (http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-spanish2.pdf).

Project managament body of knowledge (2014). El triángulo del proyecto. Recuperado el 18 de julio de 2015, del sitio Web: (http://office.microsoft.com/es-es/project-help/el-triangulo-del-proyecto-HA010351692.aspx).

Agilemanifesto.org (2001). Manifiesto por el Desarrollo Ágil de Software. Recuperado el 20 de junio de 2015, del sitio Web :(http://agilemanifesto.org/iso/es/).