



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**“ANÁLISIS Y EVALUACIÓN DEL PROCESO DE DESARROLLO DE
SOFTWARE PARA EL ÁREA DE CONSULTORÍA DE LA EMPRESA
POWERDATA CHILE”**

**TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN
TECNOLOGÍAS DE LA INFORMACIÓN**

LUIS EDUARDO RIOSECO NORAMBUENA

PROFESOR GUIA:
MARIA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISION:
ALEXANDRE BERGEL
JOHAN FABRY
GONZALO ROJAS DURAN

SANTIAGO DE CHILE
2016

Resumen

Sin duda que cuando una empresa de desarrollo de software no obtiene los resultados proyectados en un periodo determinado es un hecho no menor. Cuando los objetivos de crecimiento económico vuelven a estar bajo lo esperado, en periodos consecutivos, pasa a ser un tema preocupante dentro la organización, y si además, esto va acompañado con fuga de clientes y de capital humano, se vuelve crítico.

Sin embargo, no saber los motivos que son la causa raíz de estos hechos debe ser por lejos la mayor preocupación de la empresa y debiese ser el principal foco a remediar si se quiere idear un plan para revertir la situación.

Es precisamente lo anterior lo que aqueja actualmente a la empresa de desarrollo de software PowerData, quienes han venido presentando malos resultados en los últimos 3 años consecutivos. La empresa cree que esta ineficiencia se debe a seguir un proceso de desarrollo de software inapropiado.

El objetivo de este proyecto de tesis es trabajar sobre el core de negocio de la empresa PowerData, que es el desarrollo de software a medida. Para esto primero se modeló el proceso actual de desarrollo de soluciones a través del lenguaje SPEM 2.0 utilizando la herramienta EPF Composer.

Una vez modelado el proceso, se analizó y midió estáticamente los roles involucrados, procesos y tareas que abarca el modelo completo. Este análisis dio como resultado la identificación de una serie de potenciales errores o puntos de ineficiencia.

La siguiente actividad fue implementar y medir este modelo a partir de dos proyectos piloto de la empresa, y en base a esto se identificaron los factores que pudieron haber estado incidiendo en los malos resultados de la organización. En el análisis de la ejecución de los proyectos piloto se intentó identificar si los errores o ineficiencias potenciales identificadas en el análisis estático se manifestaban efectivamente en ineficiencias en la ejecución de los proyectos para comprobar que tan efectivo es el análisis estático como predictor de problemas en el desarrollo práctico de proyectos desarrollados bajo la metodología de la empresa.

Tabla de contenido

1	Introducción.....	1
1.1	Contexto.....	1
1.2	Problema a resolver	2
1.3	Objetivos	2
1.3.1	Objetivo General.....	2
1.3.2	Objetivos Específicos	3
1.4	Estructura del trabajo	3
2	Marco Teórico	4
2.1	Proceso de desarrollo de software.....	5
2.1.1	SPEM	5
2.1.2	Marco de uso conceptual.....	6
2.2	Estructura del meta modelo SPEM 2.0	7
2.3	EPF Composer.....	9
3	Definición del modelo de desarrollo de software de PowerData	10
3.1	Comercial	11
3.2	Planificación de proyecto	13
3.3	Requisitos	14
3.4	Kick off	15
3.5	Diseño	16
3.6	Implementación.....	17
3.7	Testing	18
3.8	Paso a producción.....	19
3.9	Cierre de proyecto.....	20
3.10	Control de cambios	21
4	Análisis estático del proceso	22
4.1	Roles	22
4.2	Tareas.....	24
4.3	Productos de trabajo	27
4.4	Diagnóstico	30

5	Aplicación del modelo en proyectos piloto	33
5.1	Proyecto SIETE, Banco Central de Chile	33
5.1.1	Planificación proyecto SIETE	34
5.1.2	Logs proyecto SIETE	34
5.1.3	Desviaciones proyecto SIETE	36
5.2	Proyecto Datamart Multicanal, Banco Estado de Chile	36
5.2.1	Planificación del proyecto Datamart Multicanal	37
5.2.2	Logs proyecto Datamart Multicanal	38
5.2.3	Desviación proyecto Datamart Multicanal.....	39
6	Análisis y Discusión de Resultados.....	39
6.1	Roles sobrecargados	39
6.2	Productos de trabajo que son input de muchas tareas	40
6.3	Tareas con muchos inputs	40
6.4	Otros	40
7	Conclusiones.....	41
8	Bibliografía	43

1 Introducción

1.1 Contexto

PowerData es una consultora de origen español dedicada a la gestión de datos, principalmente enfocada a la realización de proyectos ETL/extracción, transformación y carga de datos. La compañía está presente en España, Argentina, México, Perú y Chile. En este último país trabajan aproximadamente 40 personas entre administrativos, comerciales y profesionales TI.

Su misión la definen como crear valor para sus clientes aportándoles, mediante su apoyo experto en la Gestión de sus Datos, soluciones empresariales de gran calidad y fiabilidad, que les ayuden a conseguir mejoras apreciables y duraderas en sus resultados de negocio.

Dentro de su visión como empresa, ellos la definen como ofrecer a sus clientes un acceso sencillo a su información, minimizando riesgos y retrasos en la entrega de ésta, sea dispar o heterogénea, en tiempo real, obteniendo un coste total de propiedad mucho menor en comparación a soluciones manuales.

Cada país tiene autonomía propia, aunque la misión y visión de la empresa son únicos para toda la organización. Además, PowerData cuenta con roles jerárquicos definidos por niveles, los cuales van desde N1 a N5 y definidos de la siguiente forma:

N1. Consultor Junior; Rol técnico.

N2. Consultor Senior; Rol técnico.

N3. Jefes de proyecto y Product managers; Rol de gestión y técnico.

N4. Gerentes de cuenta; Rol comercial.

N5. Gerente de operaciones y Gerente de administración y finanzas; Rol gerencial.

El núcleo de negocio de la empresa es la venta de licencias de los productos de la compañía "Informática", de los cuales PowerData es el único partner y representante oficial en el país. Por otro lado, también vende servicios de consultoría como proyectos cerrados o como bolsas de horas. Para ello, y dependiendo de la magnitud del proyecto, se forman equipos de trabajo compuestos de consultores, quienes levantan requerimientos, desarrollan las soluciones, realizan pruebas y control de calidad, apoyados con un jefe de proyecto encargado de la gestión del mismo.

PowerData está actualmente involucrado en proyectos con clientes del rubro de la banca, retail, seguros, empresas de servicio y entidades estatales.

1.2 Problema a resolver

Durante los últimos años, 2013, 2014 y 2015, la empresa ha registrado pérdidas de hasta un 22,5% en algunos periodos en el área de consultoría, y lo que es más preocupante, ha perdido clientes que se creían fidelizados debido al no cumplimiento de objetivos y/o de plazos en los proyectos vendidos.

Los malos resultados han traído una serie de consecuencias negativas. Las más evidentes son las mermas económicas, pero además, inestabilidad laboral y un clima interno que no favorece el desarrollo profesional.

No existe una metodología de desarrollo de software formal y transversalmente aplicada dentro de la empresa, lo que dificulta el seguimiento de los proyectos, y pocas veces se definen hitos formales sobre los cuales se pueda hacer revisión de avances. Como consecuencia, no se detectan a tiempo las desviaciones con respecto a la planificación establecida.

Dentro de los puntos a mejorar, uno de los que más destaca es la poca capacidad de estimar correctamente los plazos de tiempo y recursos para los proyectos, ya que no existe una metodología formal para estas estimaciones. Por ejemplo, estimando en base a comparación de proyectos similares ya que escasamente se guarda historia.

Como consecuencia de una escasa planificación, se hace muy difícil un buen análisis de riesgo, que muchas veces no son detectados y que como consecuencia derivan en problemas reales que provocan retrasos o controles de cambios que provocan desviaciones en el proyecto.

1.3 Objetivos

Como consecuencia de lo anterior definimos los siguientes objetivos para este trabajo de tesis.

1.3.1 Objetivo General

Modelar y analizar el proceso de desarrollo de software de la empresa PowerData Chile, aplicándolo en proyectos reales para medir su eficacia al ser utilizado por la organización.

1.3.2 Objetivos Específicos

- Definir el proceso de desarrollo de software
- Definir métricas estáticas para el proceso, y métricas dinámicas para la ejecución de proyectos dentro del área de consultoría de PowerData.
- Analizar las métricas estáticas obtenidas del proceso.
- Aplicar el proceso definido en dos proyectos piloto y registrar las métricas dinámicas.
- Analizar los datos derivados de las métricas dinámicas obtenidas.
- Comprobar si los errores y oportunidades de mejora identificadas con las métricas estáticas coinciden con las ineficiencias y errores identificados con las métricas dinámicas.

1.4 Estructura del trabajo

En el capítulo 2 describimos los conceptos básicos asociados al trabajo de grado, definiciones y marco conceptual. En el capítulo 3 se aborda el modelo de desarrollo de software de la empresa, y la metodología con la que fue modelado.

En el capítulo 4, una vez modelado el proceso se analizó estáticamente cada uno de los roles, tareas y productos de trabajo generados durante todo el flujo, además de entregar un diagnóstico basado en este análisis. Luego, el capítulo 5 describe la aplicación del modelo definido anteriormente a dos proyectos piloto de similares características entre sí, mostrando los resultados de esta actividad a través de su planificación y mediciones.

En el capítulo número 6, se analizan los resultados de las mediciones tanto estáticas como empíricas y se abordan los resultados obtenidos de ambos análisis. Finalmente, en el capítulo 7, el lector encontrará las conclusiones obtenidas de este trabajo de grado.

2 Marco Teórico

La medición del proceso de desarrollo de software implica mediciones de las actividades relacionadas con el software siendo algunos de sus atributos típicos el esfuerzo, el coste y los defectos encontrados¹.

Para medir del proceso es necesario modelarlo, y para ello se utilizó la herramienta EPF Composer², entorno gráfico que facilita la tarea de formalización de procesos de software, a través de la notación SPEM 2.0³.

Una vez modelado el proceso de desarrollo de software, se tomó un conjunto de métricas sobre el modelo definido, las cuales fueron analizadas con la ayuda de la herramienta AVISPA⁴. A la hora de establecer dichas métricas se consideraron tres niveles de alcance:

- Nivel de modelo: métricas que se aplican para medir la complejidad estructural del modelo de procesos en su conjunto (se representan en la Tabla 1).
- Nivel de los elementos fundamentales del modelo (actividad, rol del proceso y producto de trabajo).
- Medición del tiempo real de las actividades planificadas, aplicándose a los dos proyectos pilotos.

La tabla 1 muestra el conjunto de métricas estáticas que se analizaron luego de la definición del proceso.

Métrica	Definición
NA	Número de actividades del modelo de procesos
NPT	Número de productos de trabajo del modelo de procesos:
NRP	Número de roles que intervienen en el proceso

*Tabla 1, métricas obtenidas de la medición del proceso de desarrollo de software.*⁵

¹Fenton y Pfleeger, 1997

² <https://eclipse.org/epf/>

³<http://www.omg.org/spec/SPEM/2.0/>

⁴ J.A. Hurtado, A. Bergel, and M.C. Bastarrica. Analyzing software process models with AVISPA. ICSSP2011:23-32.

⁵ Piattini Velthuis, M. G., García Rubio, F. Ó., Garzás Parra, J., Genero Bocco, M. F. 2008. Medición y estimación del software p 70.

2.1 Proceso de desarrollo de software

Un proceso de desarrollo de software es un conjunto de acciones que permiten transformar de forma eficiente la necesidad de un usuario en una solución de software efectiva.

La definición de un proceso de desarrollo de software es una descripción de este proceso en la que se identifican roles, tareas y artefactos, y que permiten guiar a los ingenieros durante el desarrollo de software.

Un proceso definido permite:

- Comunicación efectiva acerca del proceso entre los usuarios, desarrolladores, gerentes, clientes e investigadores.
- Mejora de la comprensión de la gerencia, entregando una base precisa para la automatización del proceso y facilitando la movilidad del personal.
- Facilita la reutilización del proceso, disminuyendo los costos asociados a la definición de procesos.
- Soporta la evolución del proceso proveyendo de medios efectivos para el aprendizaje del proceso y un sólido fundamento para la mejora de procesos.
- Ayuda en la administración del proceso. La administración efectiva requiere planes claros y una manera precisa y cuantificada de medir el estado contra ellos. Los procesos definidos hacen posibles tales herramientas.

A menudo los procesos de desarrollo de software son definidos usando notaciones informales o simples documentos de texto en lenguaje natural, debido principalmente a la facilidad con que esto puede ser implementado. Sin embargo, este tipo de notación hace más difícil la mejora ulterior del proceso. En cambio, las notaciones formales pueden requerir entrenamiento previo para desarrollarlas, pero permiten algún tipo de procesamiento automatizado a partir de ellas.

2.1.1 SPEM

Dentro de las notaciones formales que hoy existen para la especificación de procesos de software, destaca la segunda versión del Meta-Modelo de Ingeniería de Procesos de Software y Sistemas (Software and Systems Process Engineering Meta-Model) o SPEM 2.03. Esta especificación creada por el Object Management Group4 (OMG) provee los conceptos necesarios para el modelamiento, documentación, presentación, administración e intercambio de procesos y métodos de desarrollo de software.

2.1.2 Marco de uso conceptual

Conceptualmente, un proceso que ha sido especificado usando SPEM, permitiría:

- **Proporcionar librerías de elementos estandarizados y reutilizables.**

Las personas involucradas en el desarrollo de software, necesitan conocer los métodos y prácticas que se requieren en su trabajo: cómo especificar y gestionar requerimientos, cómo diseñar casos de prueba, etc., así como conocer los productos de trabajo que estas tareas generan o usan. SPEM permite establecer una base de conocimientos estandarizada, generada a partir de elementos tales como: descripción de tareas, descripción de productos de trabajo, guías, plantillas, mejores prácticas, etc. Estos elementos conforman lo que SPEM define como *biblioteca del método*. Desde esta biblioteca se extraen los componentes que permitirán estructurar los procesos de desarrollo.
- **Permitir la elaboración, administración y crecimiento sistemático de procesos.**

Los equipos de desarrollo necesitan definir el modo de aplicar sus métodos y mejores prácticas al ciclo de vida de un proyecto. Es decir, necesitan definir o seleccionar un proceso de desarrollo. SPEM permite la creación sistemática de estos procesos, basada en los elementos existentes en la biblioteca del método de la organización. SPEM permite, por ejemplo, crear ciclos de vida modulares y luego combinarlos para crear un proceso de desarrollo determinado. Una vez creados, estos procesos pueden ser representados a través de diagramas de flujo o estructuras de descomposición de trabajo. También es posible crear patrones de proceso o módulos que representen procesos de referencia de determinadas disciplinas, tecnologías o estilos de desarrollo (por ejemplo, OpenUp, SCRUM, etc.) que luego pueden ser compartidos y usados por terceros.
- **Permitir el uso de contenido y procesos adecuados a cada proyecto.**

SPEM permite usar sólo aquellos elementos de contenido y procesos que son necesarios para cada proyecto de la organización, considerando que ningún proyecto es exactamente igual al anterior y que un proceso nunca es ejecutado de la misma forma. Para ello permite la definición de configuraciones y contenido de método, así como diversos mecanismos de adecuación (*tailoring*).
- **Apoyar la aplicación de los procesos en los proyectos de desarrollo.**

Una definición de proceso sólo entrega valor si permite guiar el comportamiento de los equipos de desarrollo. Los procesos y sus elementos necesitan estar disponibles en el contexto del trabajo diario de los jefes de proyecto, líderes técnicos y desarrolladores, en formatos que

puedan ser usados en los sistemas de la organización. SPEM 2.0 provee estructuras de definición de procesos que permiten indicar cómo un proceso debería ser usado dentro de estos sistemas. Por ejemplo, una definición de proceso hecha mediante SPEM puede incluir información que indique que ciertas tareas deben ser repetidas varias veces en un proyecto o que múltiples ocurrencias de ellas pueden ser efectuadas en paralelo.

2.2 Estructura del meta modelo SPEM 2.0

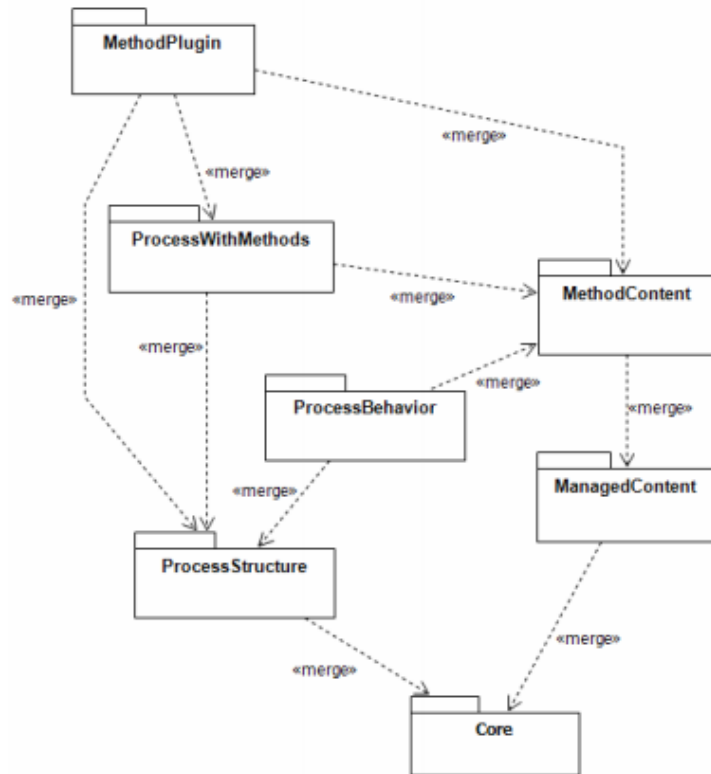


Figura 1 – Estructura del metamodelo

El meta-modelo SPEM está estructurado en 7 paquetes o unidades lógicas principales (Figura 1). Cada una de ellas extiende a la unidad de la que depende, agregándole nuevas estructuras y capacidades. El mecanismo “merge” de UML 2 que se realiza entre los paquetes tiene como efecto la extensión gradual de las capacidades modeladas, unidad por unidad. Como resultado, las unidades definidas en una capa más baja pueden ser usadas en una implementación de SPEM sin las unidades de las capas superiores.

Estos 7 paquetes presentan las siguientes características:

1. **Núcleo (Core):** Contiene todas las clases y abstracciones del meta-modelo que el resto de las clases (pertenecientes a otros paquetes) usan como base.

2. **Estructura de Proceso (Process Structure):** Este paquete define la base para todos los modelos de proceso. Sus estructuras de datos permiten representar procesos como un desglose de Actividades anidadas con referencias a clases de Roles, así como a clases de Productos de Trabajo. Estas estructuras son usadas para representar procesos básicos y de alto nivel, sin documentación textual.
3. **Comportamiento de Proceso (Process Behavior):** Este paquete permite extender las estructuras estáticas del paquete anterior a través del uso de modelos de comportamiento. Sin embargo, no provee modelos de comportamiento propios, sino que provee de “enlaces” a modelos definidos externamente por terceros. Por ejemplo, un proceso definido con los conceptos del paquete de Estructura de Proceso puede ser enlazado a diagramas de actividad de UML 2 para representar el comportamiento de dicho proceso.
4. **Contenido Administrado (Managed Content):** En muchos casos, los procesos no sólo son representados por modelos, también son documentados y administrados usando descripciones en lenguaje natural. Para muchos enfoques de desarrollo de software, una documentación en lenguaje natural tiene mayor valor que un conjunto de modelos preciso; el paquete de Contenido Administrado introduce conceptos para el manejo de este tipo de contenidos, los que pueden ser usados de forma separada o en conjunto con conceptos de Estructura de Proceso.
5. **Contenido del Método (Method Content):** Este paquete provee los conceptos que permiten construir una base de conocimientos de desarrollo, independiente de procesos específicos o proyectos en particular. Permite definir elementos reusables que proveerán una base de conocimiento documental de métodos y técnicas de desarrollo de software; los procesos pueden re usar estos elementos del Contenido del Método y relacionarlos en secuencias parcialmente ordenadas que pueden ser adaptadas a proyectos específicos.
6. **Proceso con Métodos (Process With Methods):** Este paquete define nuevas estructuras y redefine las ya existentes en el paquete de Estructura de Proceso, integrándolas con los conceptos del paquete de Contenido del Método. Mientras que el Contenido del Método define métodos y técnicas fundamentales para el desarrollo de software, los procesos se encargan de poner estos métodos y técnicas en el contexto de un modelo de ciclo de vida compuesto, por ejemplo, de fases e hitos.
7. **Plug-in del Método (Method Plugin):** Este paquete entrega conceptos que permiten el diseño y mantención de bibliotecas o repositorios de procesos y contenido de método.

2.3 EPF Composer

Eclipse Process Framework (EPF) es un proyecto de código abierto que es administrado por la Fundación Eclipse. Sus principales objetivos son:

- Proporcionar un marco extensible y herramientas de primer nivel para la creación, gestión, configuración y publicación de procesos formales. EPF Composer tiene catálogos de procesos predefinidos para situaciones típicas de procesos que pueden ser adaptados a necesidades individuales.
- Proveer a los desarrolladores con una base de conocimiento de capital intelectual que les permita buscar, administrar y desplegar contenido. Esta base de conocimiento puede ser usada como referencia y material educativo, y forma la base para el proceso de desarrollo. EPF Composer está diseñado para ser un administrador de contenido que provee una estructura de administración y un aspecto común para todo el contenido.

3 Definición del modelo de desarrollo de software de PowerData

Dentro de los objetivos de este trabajo de tesis se estableció definir el proceso de desarrollo de software de PowerData. Esta tarea se logró luego de 8 sesiones de entre dos a tres horas cada una en conjunto con la empresa y un equipo del departamento de ciencias de la computación de la Universidad de Chile.

La metodología que se utilizó fue entrevistas personales con distintos profesionales que ejercían un rol específico en cada una de las partes del proceso. A través de estas entrevistas se definió el proceso completo de desarrollo de software de PowerData modelándolo con la herramienta EPF Composer.

A continuación, se detalla el modelo general del proceso de desarrollo de software de PowerData modelado en EPF Composer.

El modelo se compone de 10 actividades (ver figura 2). Cada una de ellas se detalla a partir del punto 3.1. Estas actividades son:

- Comercial
- Planificación de proyecto
- Requisitos
- Kickoff
- Diseño
- Implementación
- Testing
- Paso a producción
- Cierre de proyecto
- Control de cambios

Esta última actividad, es transversal a casi todo el modelo de desarrollo de software, ya que puede presentarse en cualquier etapa del mismo, sólo antes de paso a producción y cierre de proyecto.

A continuación, la figura 2, muestra una imagen general de las actividades que componen el modelo de desarrollo de software de la empresa.

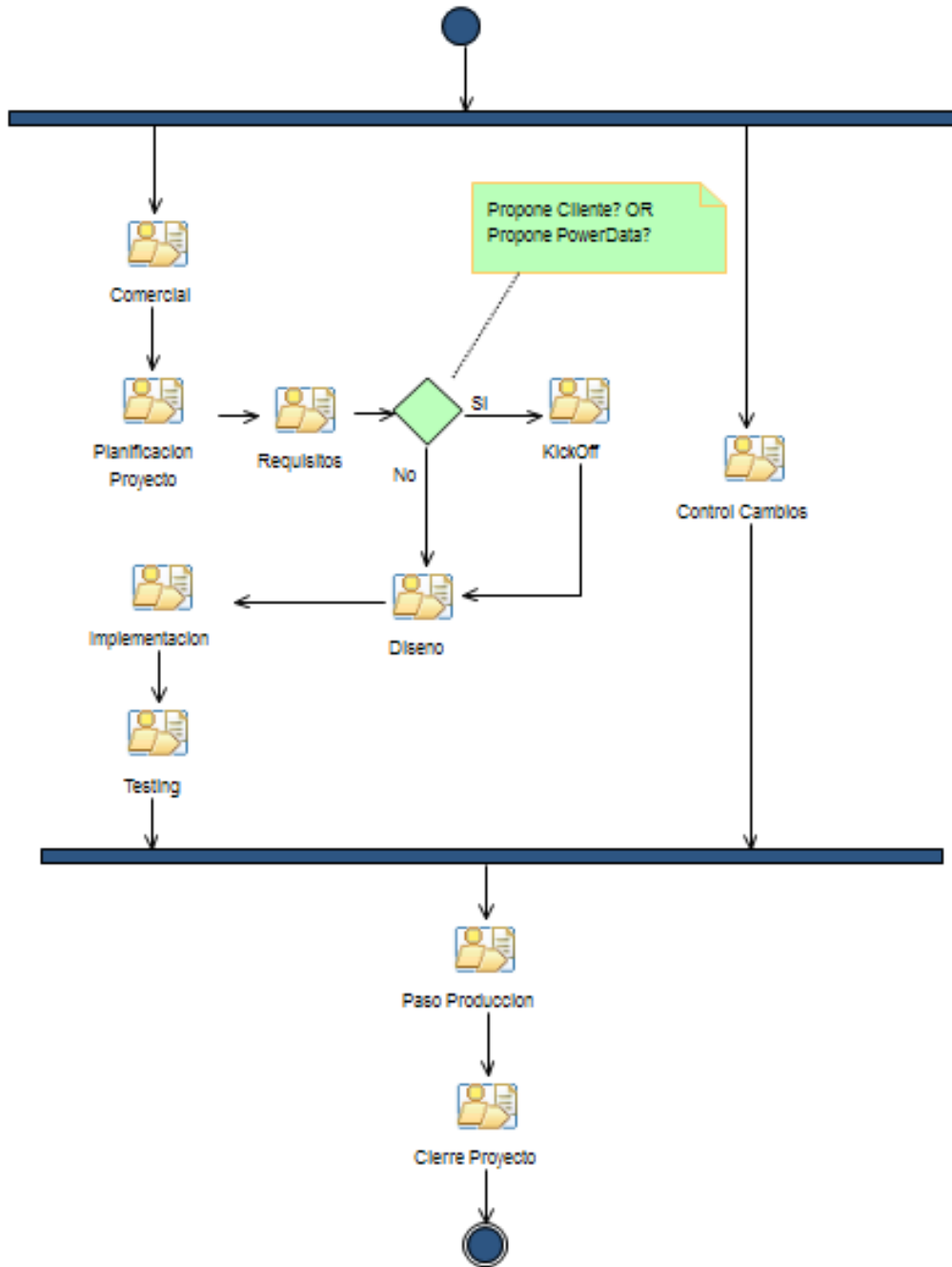


Figura 2 – Modelo del proceso de desarrollo realizado en EPF Composer.

3.1 Comercial

La actividad comercial tiene relación con el origen de nuevas oportunidades de negocio a partir de clientes nuevos o clientes ya fidelizados. En esta actividad la empresa planifica sus propuestas de proyecto, negocia y finalmente si se acepta, se da curso a la propuesta lo que gatilla las siguientes actividades del proceso. Se

generan artefactos como la orden de compra y el contrato, que son requisitos para iniciar el proyecto.

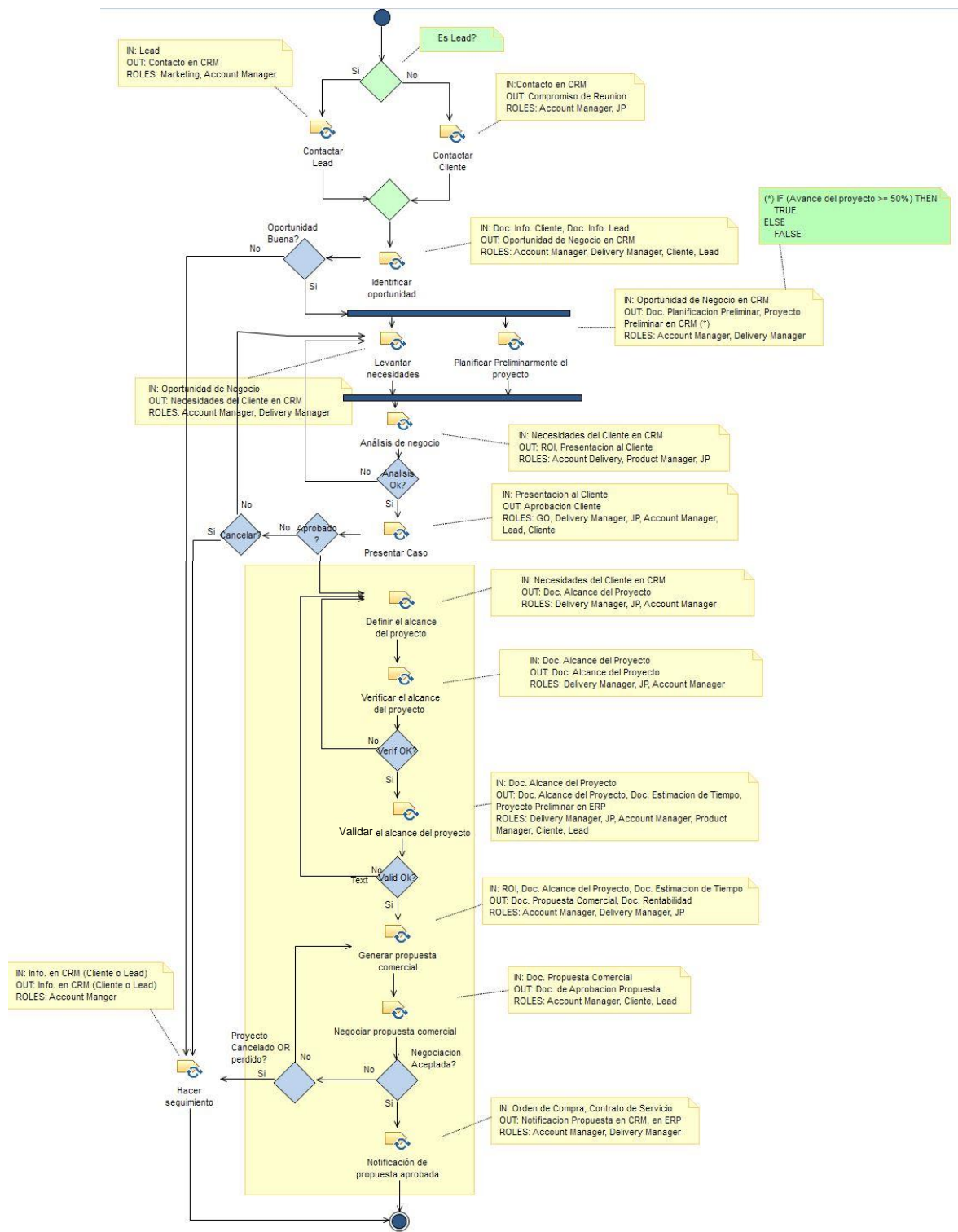


Figura 3 – Modelo de la actividad comercial.

Como se menciona anteriormente, en la figura 3 se modeló la actividad “Comercial”, que es la primera de las actividades al iniciar un nuevo proyecto de la empresa. A modo de explicación, la tarea del modelo “Contactar Cliente”, que es la primera tarea de esta actividad, tiene como entrada (“IN” en la etiqueta de la tarea) el contacto del cliente en el CRM de la compañía, como salida (“OUT” en la etiqueta) se genera un compromiso de reunión con el cliente, y finalmente los roles que participan (“ROLES”) son el jefe de proyectos o JP y el account manager.

De igual forma como lo explicado anteriormente se deberá leer el modelamiento de los procesos realizado con EPF Composer para las demás actividades.

3.2 Planificación de proyecto

En esta actividad se planifican las siguientes tareas que tendrá el proyecto, se considera la planificación de los recursos involucrados, se informa a administración y finanzas de la planificación y se notifican a los involucrados que tendrá el proyecto.

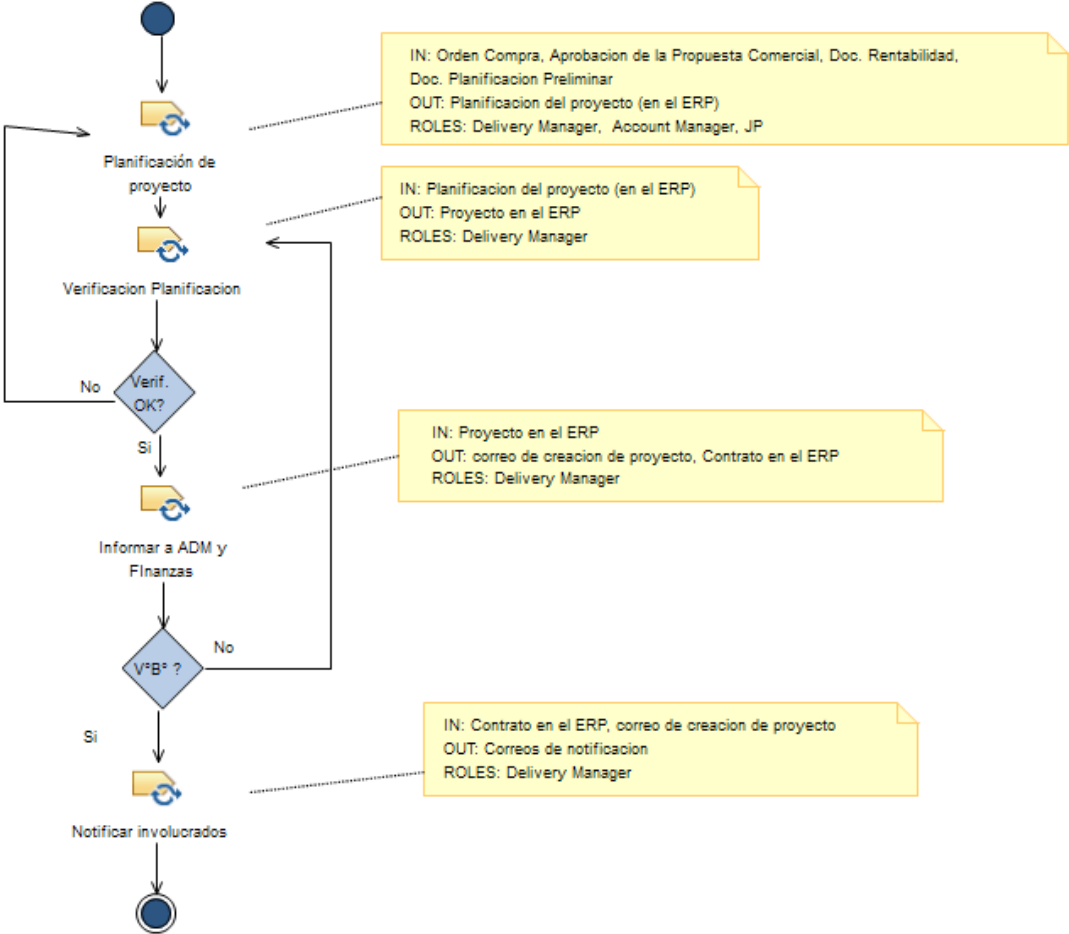


Figura 4 – Modelo de la planificación de proyecto.

3.3 Requisitos

En la actividad de requisitos se analizan las necesidades del cliente respecto al proyecto. Estas necesidades son verificadas y luego validadas por el cliente. De esta actividad se genera un documento de requisitos que será la base sobre la cual se desarrollará el proyecto.

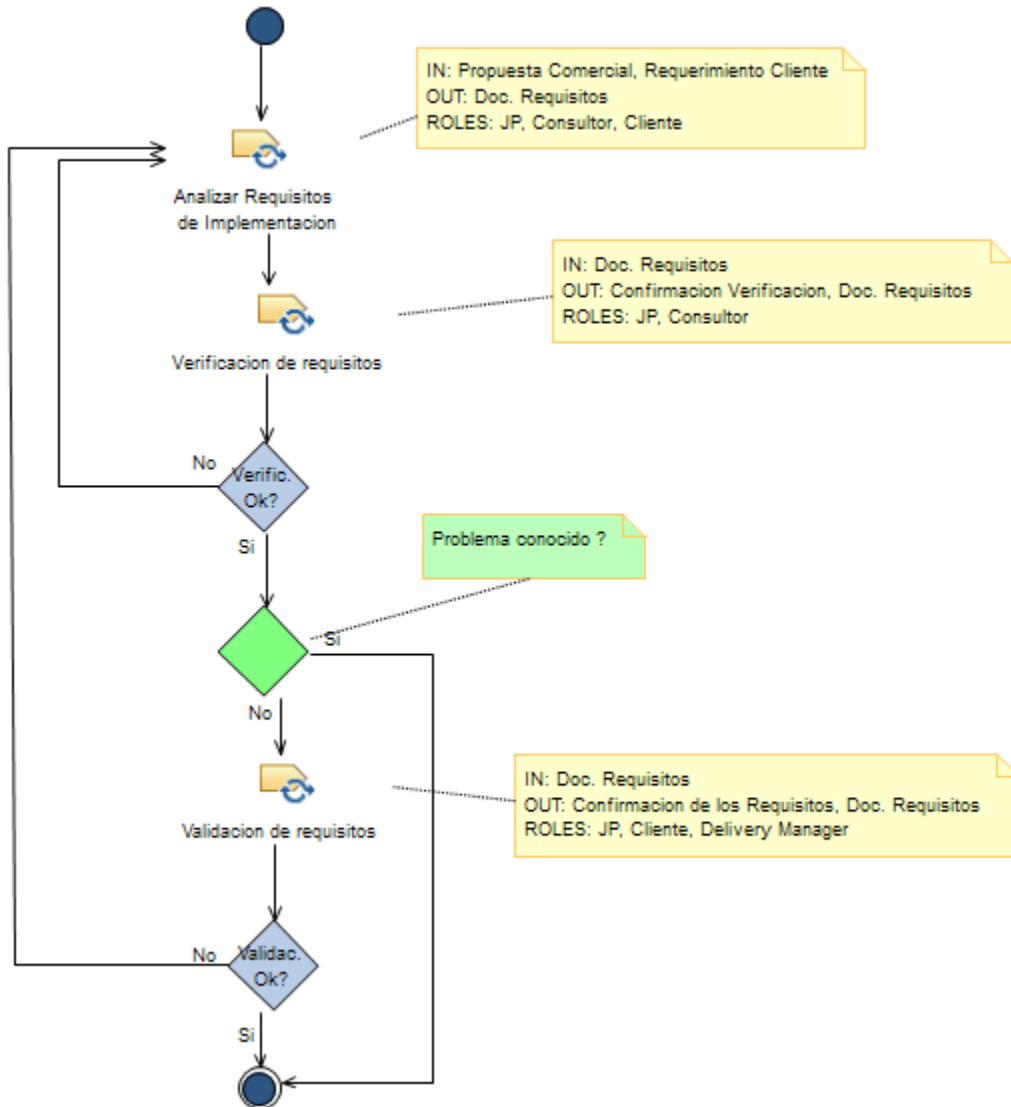


Figura 5 – Modelo de desarrollo de requisitos.

3.4 Kick off

Esta actividad tiene como propósito dar el inicio formal al proyecto de desarrollo. En ella se incluyen todos los stakeholders, se planifica y se realiza una reunión donde se presentan las propuestas comerciales y la planificación del proyecto.



Figura 6– Kick off.

3.5 Diseño

En esta actividad se genera, verifica y valida el diseño que tendrá la solución. De esta actividad se obtiene el documento de diseño, el cual será la guía con la que trabajarán los consultores para desarrollo del proyecto.

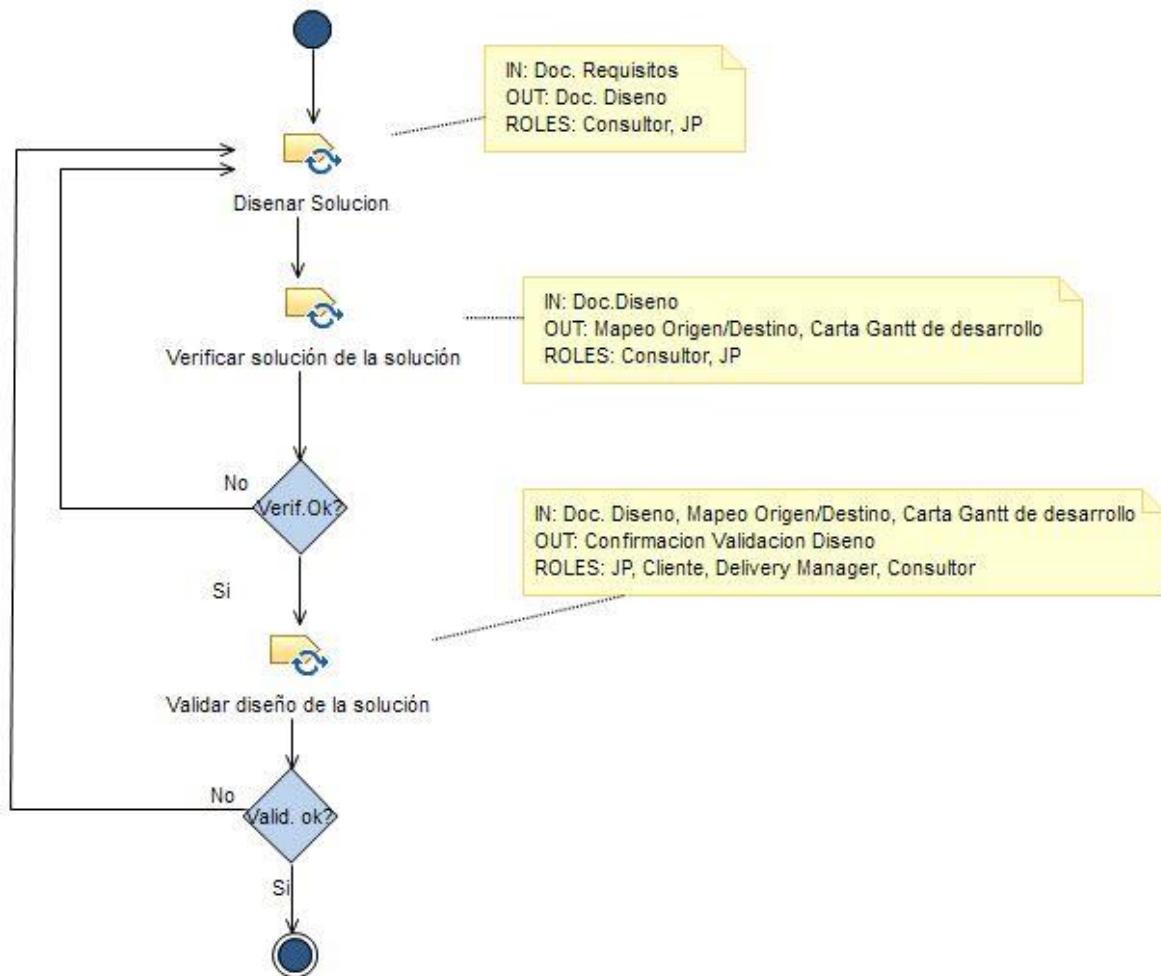


Figura 7 – Modelo de diseño.

3.6 Implementación

En esta actividad se lleva a cabo el desarrollo como tal de la solución. Existen tareas para validar el ambiente de trabajo, coordinar las tareas de los stakeholders, y realizar el desarrollo. En esta actividad también se ejecutan los casos de pruebas unitarias y se analizan los posibles errores. En caso de que los errores encontrados no se encuentren dentro del alcance del proyecto definido en la actividad de requisitos, será necesario revisar nuevamente el alcance del proyecto y de ser necesario aplicar un control de cambio.

Si los errores encontrados son parte del alcance entonces requerirá iteraciones hasta que sean corregidos y la solución sea entregada para ser validada por QA en la siguiente actividad.

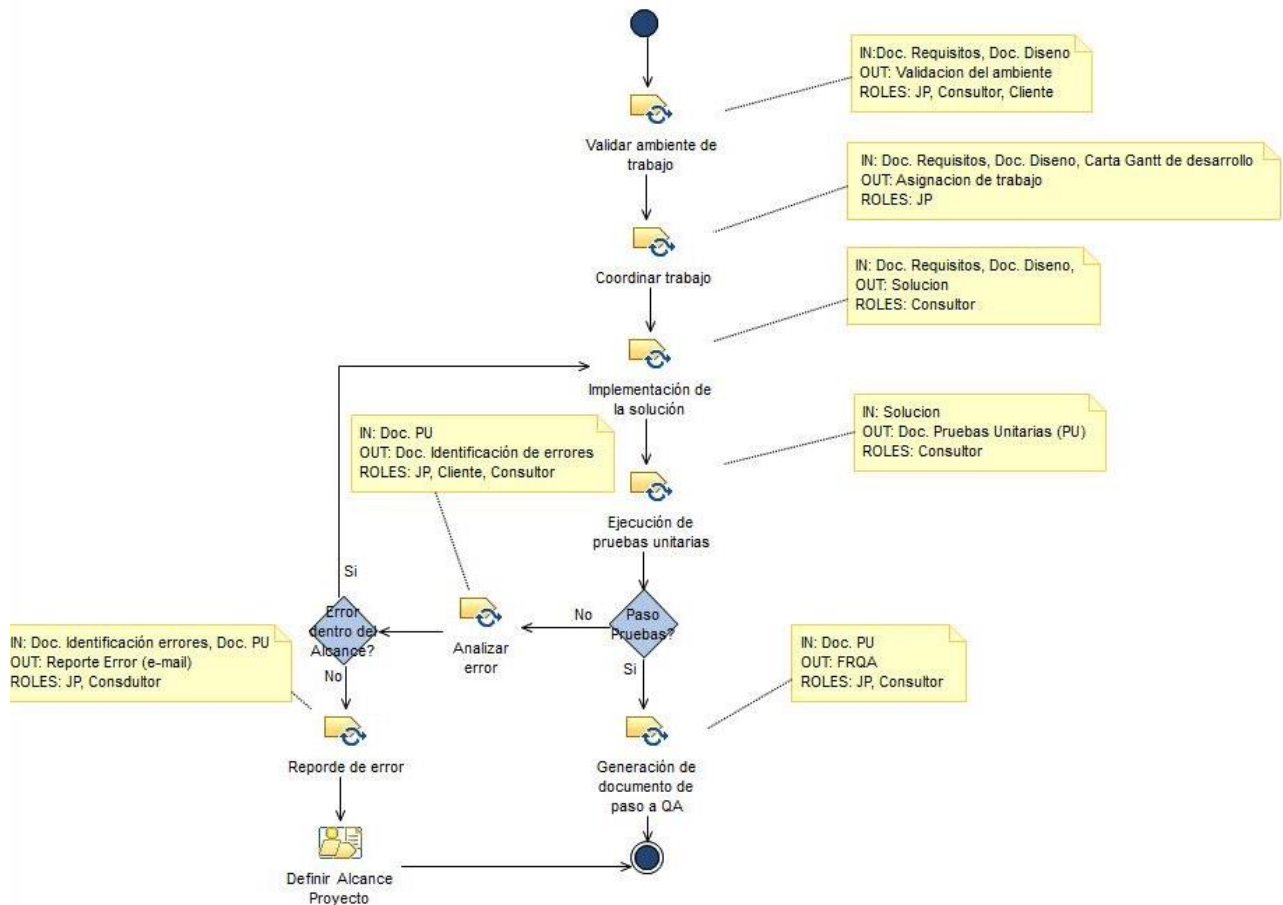


Figura 8 – Modelo de implementación de la solución.

3.7 Testing

Durante la actividad de testing, el consultor de PowerData, junto con su contraparte del cliente, coordina las pruebas de la solución desarrollada. Para eso se planifican y ejecutan los casos de prueba.

Luego de ejecutado el plan de pruebas, se analizan los posible errores y defectos encontrados. Si estos errores o defectos se encuentran dentro del alcance del proyecto, entonces el desarrollo vuelve a la etapa de implementación para que sean corregidos. De lo contrario, si los errores encontrados no están dentro del alcance, se evalúa el alcance y de ser necesario se genera un control de cambios para incluir la solución de los errores o defectos.

Finalmente, cuando el plan de pruebas es completado sin errores ni defectos, se genera un documento que contiene los resultados para ser aprobado por el cliente

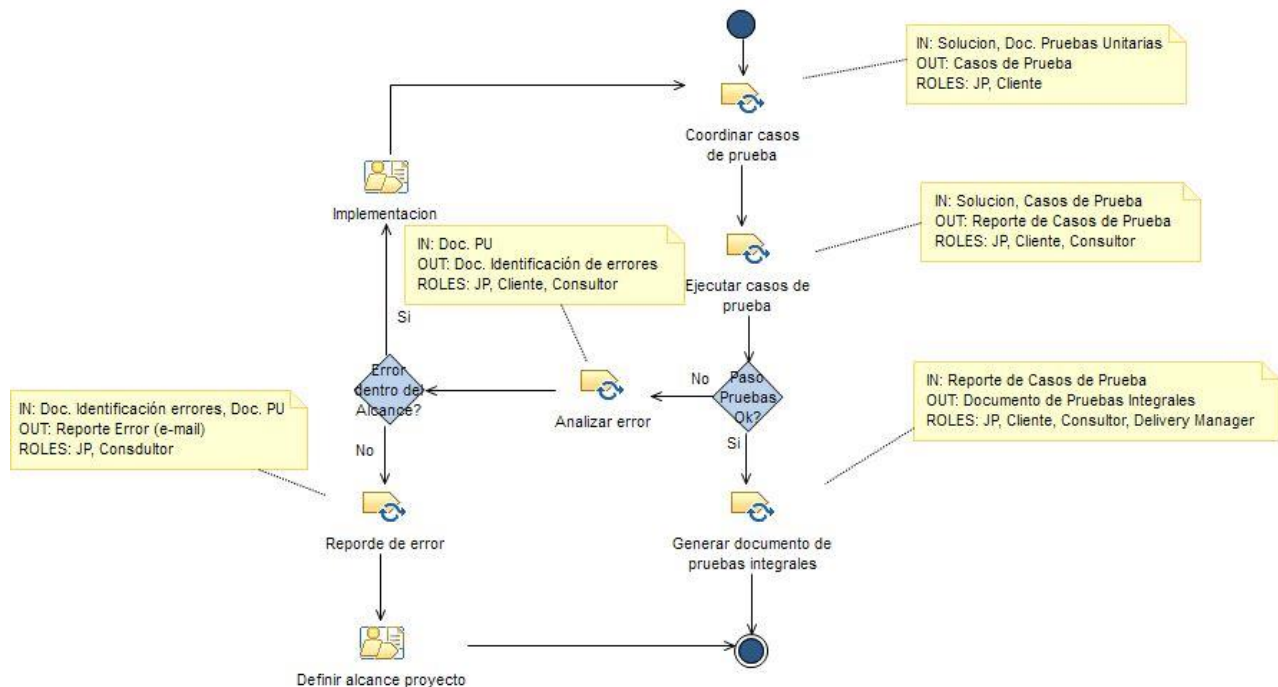


Figura 9 – Modelo de testing de la solución.

3.8 Paso a producción

Una vez que el desarrollo está finalizado y se completa con éxito el plan de pruebas, el siguiente paso de la planificación es el paso a producción de la solución. Para ello, se generan todos los artefactos que sean requeridos por el cliente y se coordina con los stakeholders la fecha de esta actividad para finalmente concretarla de acuerdo a lo planificado.

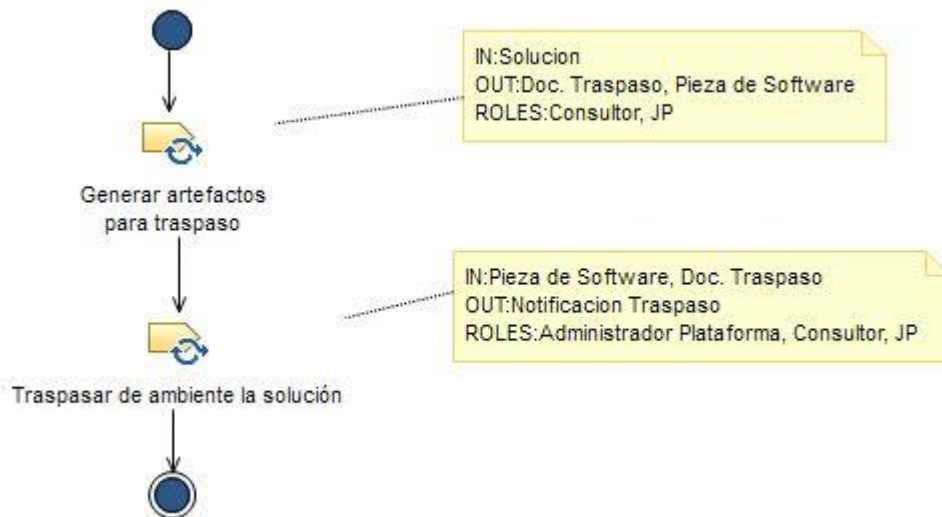


Figura 10 – Modelo de paso a producción.

3.9 Cierre de proyecto

Esta actividad tiene por objetivo el cierre formal del proyecto. Para ello se genera un documento de cierre que contiene el resumen del proyecto y su planificación. Este documento es validado por el cliente y se entrega y acepta como hito de cierre en una reunión generada para este propósito.

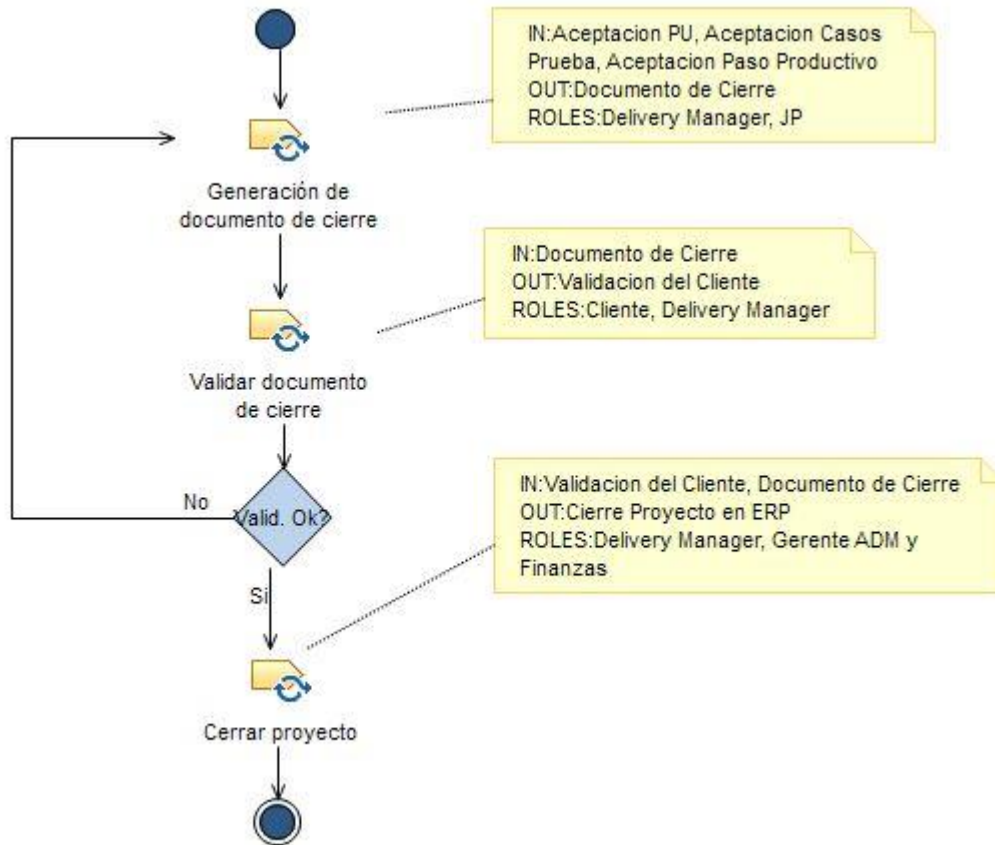


Figura 11 – Cierre del proyecto.

3.10 Control de cambios

El control de cambios, es una actividad que puede gatillarse transversalmente a lo largo de casi todo el modelo de desarrollo de software de PowerData, sólo a condición de ser antes del paso a producción y del cierre del proyecto.

La actividad de control de cambios involucra la solicitud formal del cambio por parte del cliente. Esta solicitud es verificada tanto técnica como comercialmente por PowerData. Si una de estas verificaciones no es factible, se informa al cliente y se evalúan nuevas alternativas. Por el contrario, si ambas evaluaciones son factibles, entonces se validan con el cliente y se notifica la aprobación del control de cambio a los stakeholders.

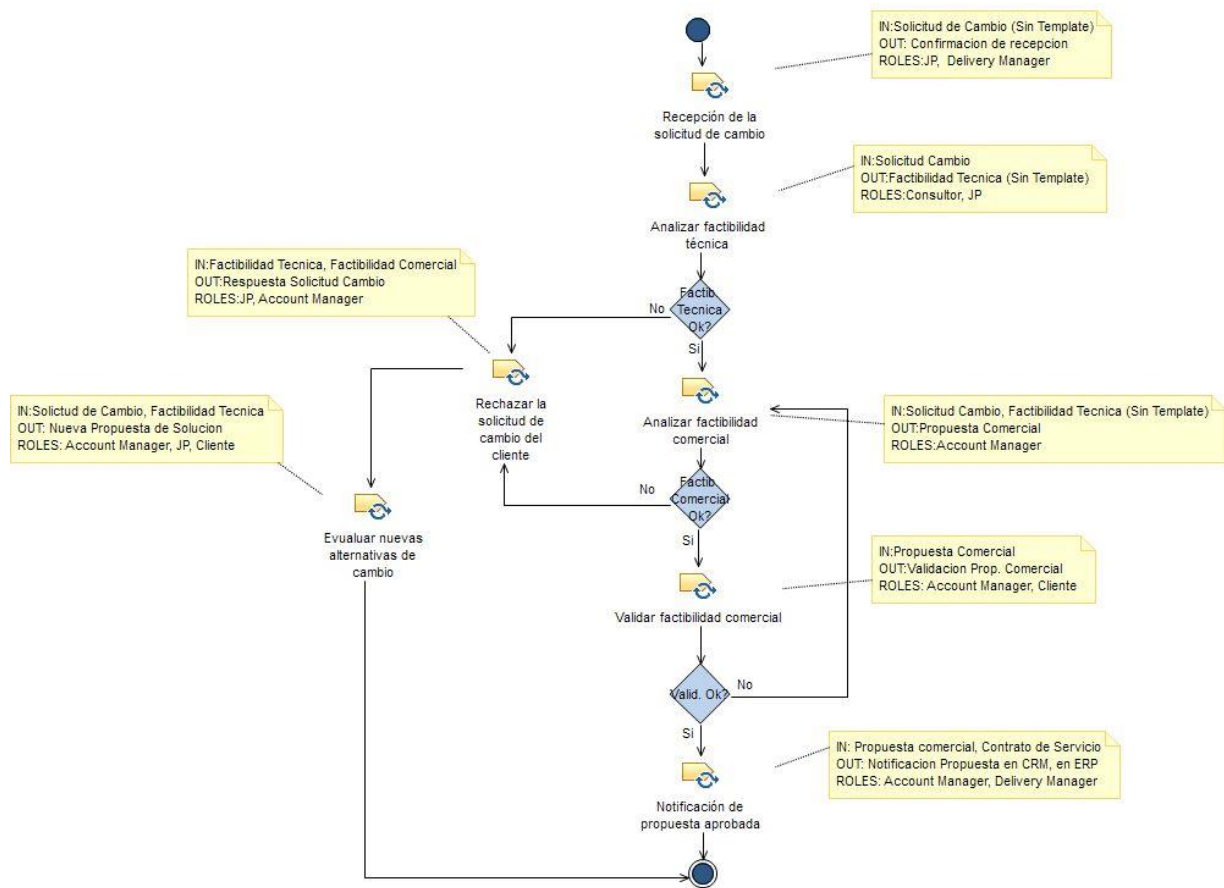


Figura 12 – Modelo de control de cambios.

4 Análisis estático del proceso

El análisis del modelo del proceso se realizó con la ayuda de la herramienta AVISPA⁶. Esta herramienta toma un modelo de proceso especificado en EPF Composer y gráficamente presenta diferentes patrones resaltando los errores potenciales a través de indicadores comprensibles e intuitivos. Los enfoques de AVISPA han sido validados aplicándolos en la definición y análisis de algunos modelos de proceso de la industria de software chilena y algunos procesos públicos disponibles desde la comunidad de Eclipse Process Framework. Estos estudios muestran ampliamente la utilidad práctica del enfoque dirigido por modelos para lograr modelos de procesos de alta calidad.

4.1 Roles

En el modelo del proceso de desarrollo de software de PowerData se definieron 10 roles que participan a lo largo de todo el proceso. El detalle de estos roles y su definición es la siguiente:

1. **Account Manager (AC):** Este rol es el encargado de generar nuevos negocios para la compañía, administrar a los clientes y generar con ellos nuevas oportunidades.
2. **Administrador de plataforma (ADM):** Rol encargado de administrar la plataforma PowerCenter. Esta plataforma es el framework sobre el cual la empresa desarrolla sus proyectos de software. Este es un rol propio de cada cliente con el que trabaja PowerData.
3. **Cliente (CLT):** Se refiere a la contraparte de PowerData dentro de un proyecto.
4. **Consultor (CTR):** Este rol es el encargado de la toma de requerimientos de un proyecto, el desarrollo, la documentación y pruebas del mismo.
5. **Delivery Manager (DM):** Este rol es el encargado de la administración del proyecto, principalmente la administración de los recursos. Su gestión comienza al inicio del proyecto y se extiende durante todo el ciclo de vida.
6. **Gerente de administración y finanzas (GAF):** Encargado de las finanzas de la compañía.
7. **Gerente de operaciones (GO):** Rol encargado de la parte operativa de la empresa, su principal función es la de administrar los proyectos de PowerData.

⁶J.A. Hurtado, A. Bergel, and M.C. Bastarrica. Analyzing software process models with AVISPA. ICSSP2011:23-32.

8. **Jefe de proyectos (JP):** El jefe de proyectos tiene como tarea gestionar los proyectos de desarrollo de software de la empresa; su rol involucra la gestión de los consultores.
9. **Lead (LD):** El lead es el nombre que se le da en PowerData a un prospecto de cliente.
10. **Marketing (MKT):** Este rol es el que se le da al encargado de marketing de la compañía. Se encarga de contactar leads y organiza seminarios y presentaciones para generar nuevas oportunidades de negocio.

En AVISPA, la Vista de Roles muestra el proceso enfocado desde los roles que participan en él. La vista presenta cada rol como un nodo, cuyas dimensiones representan el número de productos de trabajo que el rol modifica o crea, y el número de tareas que efectúa. Además, la colaboración de dos roles en la ejecución de una tarea, se representa mediante un enlace entre ellos (ver figura siguiente).

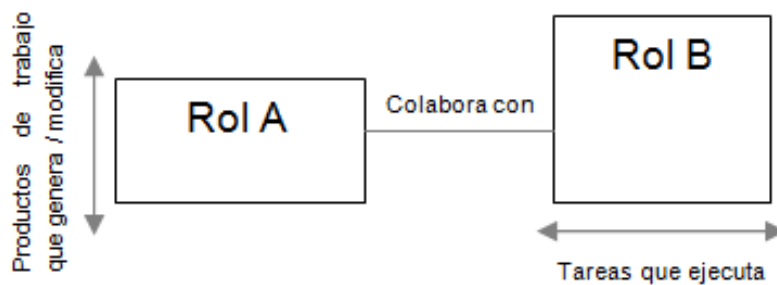


Figura 13 - Representación de nodos y enlaces de la Vista de Roles en AVISPA.

Como resultado del análisis de los roles del proceso de PowerData con AVISPA, se generó la siguiente vista:

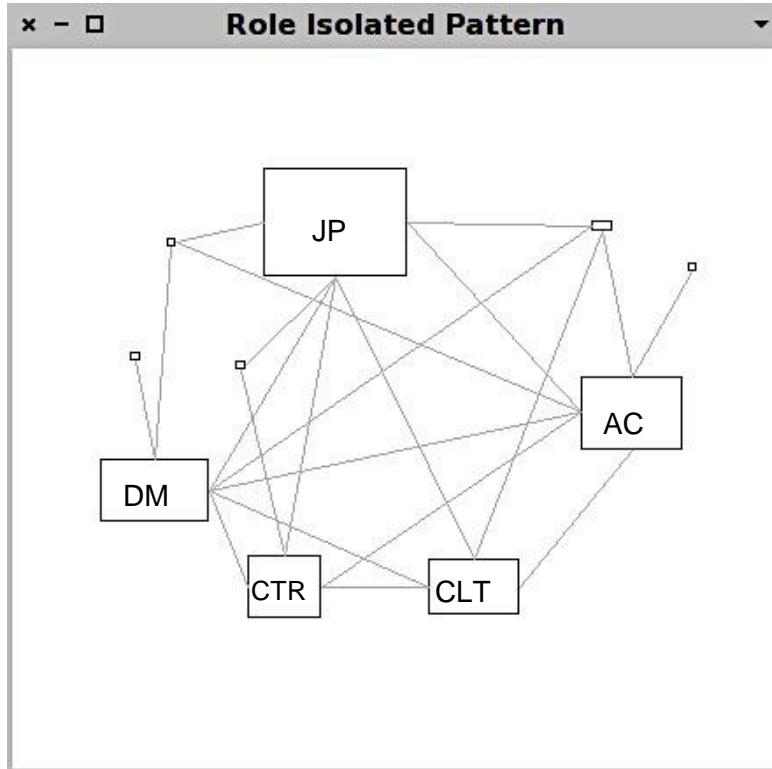


Figura 14 – Visualización de los roles del proceso en AVISPA.

La figura 14 muestra una vista de los roles del modelo de desarrollo de PowerData analizados con AVISPA. Se aprecia que el rol más sobrecargado es el jefe de proyectos: participa directamente en la confección de 3 productos de trabajo, pero está involucrado profundamente con 31 de las 45 tareas que componen el proceso. Esta situación genera riesgo dentro de los proyectos, ya que, si por algún motivo el jefe de proyecto no está disponible, todas las tareas asociadas a él dentro del proceso al igual que los productos de trabajo que genera o modifica se retrasarán.

Algo similar ocurre con otros roles que también están a cargo de ejecutar un gran número de tareas (ancho de los nodos) como el Delivery Manager y el Account Manager.

4.2 Tareas

La Vista de Tareas permite visualizar el proceso de desarrollo desde la perspectiva de las tareas que se realizan durante su ejecución. Cada nodo de esta vista corresponde a una tarea del proceso, cuyas dimensiones representan el número de productos de trabajo que ella usa como entrada (alto) y el que genera o modifica (ancho). En esta vista el color del nodo también tiene un significado: representa la cantidad de roles que participan en su ejecución; mientras más oscuro es el nodo, mayor cantidad de roles involucrados en la tarea y viceversa.

Por último, el enlace entre nodos tiene dirección en esta vista, y representa el orden de precedencia de las tareas (ver figura siguiente).

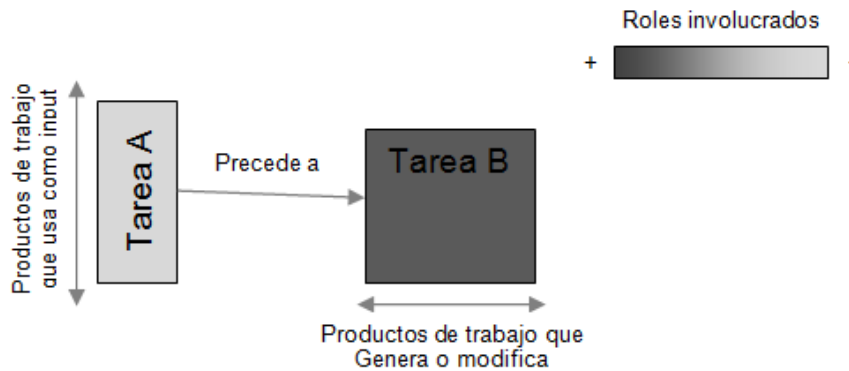


Figura 15 - Representación de nodos y enlaces de la Vista de Tareas en AVISPA.

El modelo de desarrollo de software de PowerData se compone de 45 tareas que se pueden ver en el análisis con AVISPA a continuación.

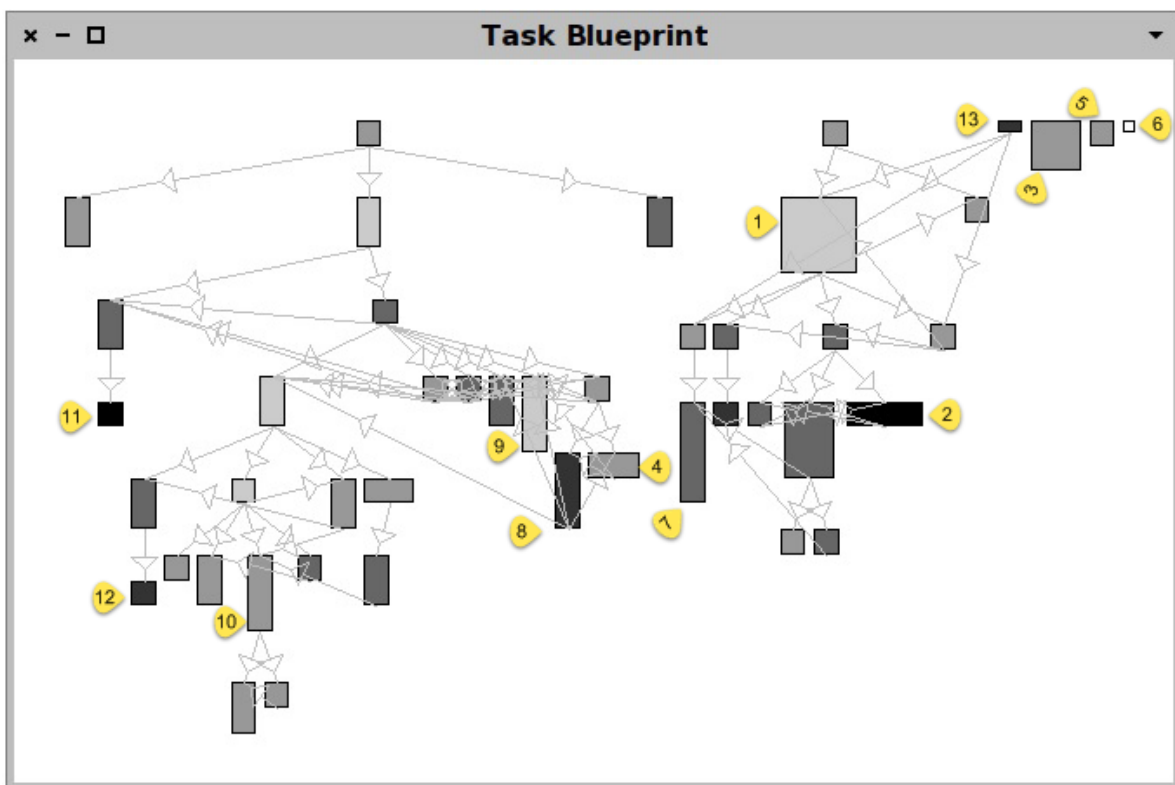


Figura 16 – Visualización de las tareas del proceso en AVISPA.

Se puede observar del análisis (figura 16) que existen pocas tareas que generan gran cantidad de productos de trabajo, lo cual se grafica como el ancho de un nodo, tal como se ve en la figura 15. Estas tareas son la número 1, 2, 3 y 4

correspondientes a: “Hacer seguimiento comercial de una propuesta”, “Validar alcance”, “Notificar propuesta aprobada” y “Verificar diseño de la solución”. Múltiples salidas pueden revelar que las tareas no están especificadas con la granularidad apropiada. Estas tareas podrían ser muy complejas ya que su objetivo no es único.

Otro de los puntos relevantes son aquellas tareas que tienen como input muchos productos de trabajo, es decir, aquellos nodos largos, como se muestra en la figura 16. Existen muchas tareas con esta característica, destacando entre ellas las 7, 8, 9 y 10 que corresponden a “Planificación de proyecto”, “Validar diseño de la solución”, “Coordinar trabajo” y “Generar documento de cierre” respectivamente (figura 15). El riesgo que genera esto es un potencial retraso de las tareas, ya que son necesarios varios productos de trabajo independientes entre sí para comenzar una tarea.

En referencia a la opacidad de los nodos, que tiene relación con la cantidad de roles involucrados en cada tarea, destacan por su oscurecimiento los nodos 2, 8, 11, 12 y 13, correspondiente a las tareas “Validar alcance”, “Validar diseño de la solución”, “Realizar reunión de lanzamiento”, “Generar documento de pruebas integrales” e “Identificar oportunidad” respectivamente. Dentro de este contexto también resulta riesgoso si existen demasiadas tareas en las que participen muchos roles, ya que se corre el riesgo de retraso en la ejecución de las tareas, debido a que se requiere coordinar múltiples personas. Sin embargo, en el modelo no existen demasiadas tareas con estas características, siendo más o menos 5 nodos, y algunos de ellos, como la tarea número 11 (“Realizar reunión de lanzamiento”), no son predecesoras de ninguna otra tarea, lo que podría ser un potencial error ya que no sería necesario ejecutarla para seguir el proceso. Esto podría ser una de las fallas de interconexión entre las áreas comercial y de desarrollo del proceso.

Finalmente, al analizar aquellas tareas que se encuentran aisladas del resto del proceso, es decir, aquellas que no tienen sucesores, se puede observar que en el modelo solo se presentan tres de ellas, la 3, 5 y 6 que corresponden a:

- **Notificar propuesta aprobada:** Esta tarea corresponde a la notificación de aceptación del cliente o prospecto de la propuesta de proyecto al Account Manager de PowerData. Este hito marca el inicio de la planificación del proyecto.
- **Recepción de solicitud de cambio:** Tarea que marca como hito la recepción de una solicitud de cambio que viene del cliente.
- **Verificación de la planificación:** Esta actividad consiste en verificar la planificación realizada por el Delivery Manager del proyecto, tiene por

objetivo revisar tiempos, costos y recursos destinados para comenzar un nuevo proyecto.

Los productos de trabajo que generan estas tareas no son utilizados por otras, y corresponden a actividades de verificación, notificaciones o recepciones que se marcan como realizados con envíos de correo o vistos buenos informales.

Finalmente se puede observar (figura 16) que existen dos grandes grupos de tareas que no están unidos entre sí. El grupo de la derecha corresponde a tareas relacionadas con el desarrollo del proyecto, y el de la izquierda a las tareas relacionadas con la administración, finanzas y marketing del mismo. Llama la atención que no exista alguna tarea que relacione ambas áreas del mismo proyecto entre ellas; esto puede deberse a una omisión al momento de modelar el proceso con PowerData o a que no hay claridad acerca de cómo se relacionan.

En un modelo ideal, lo mejor sería que los nodos de las tareas sean graficados de la manera más cuadrada posible, es decir, que no necesiten tantos input de entrada (largo del nodo) y que no generen una gran cantidad de productos de trabajo (ancho). Adicionalmente lo ideal sería que los nodos fuesen lo más claro posibles, de esta forma se disminuye la excesiva dependencia entre las tareas lo que supone riesgos de retraso en las actividades.

4.3 Productos de trabajo

La Vista de Productos de Trabajo permite observar el proceso enfocado en los productos de trabajo que éste genera. En esta vista cada nodo equivale a un producto de trabajo, cuyos atributos tienen el siguiente significado:

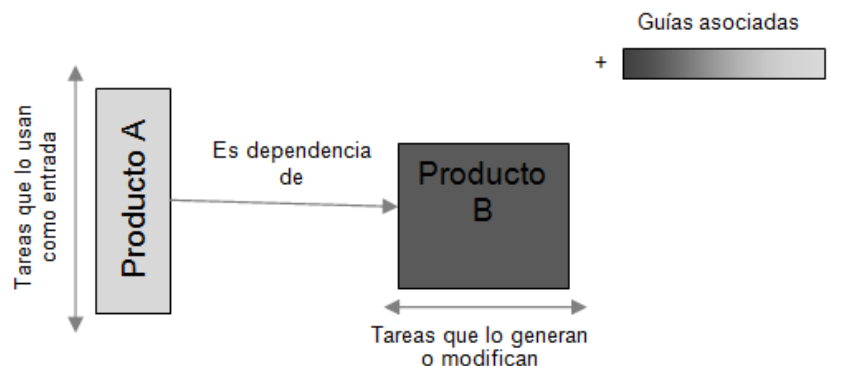


Figura 17 – Representación de los productos de trabajo del modelo en AVISPA.

- **Alto:** Número de tareas para las cuales el producto es una entrada. A mayor altura, mayor cantidad de tareas y viceversa.

- **Ancho:** Número de tareas para las cuales el producto es una salida. A mayor ancho, mayor cantidad de tareas y viceversa.
- **Color:** Número de guías que existen para el producto de trabajo. Mientras más oscuro es el nodo, mayor cantidad de guías existentes y viceversa.
- **Enlace:** Orden de dependencia de los productos de trabajo. Un producto de trabajo depende de otro si el primero es una salida de una tarea donde el segundo es una entrada.

De manera similar a lo que ocurre en la vista de tareas, los nodos están ordenados por dependencia, de arriba abajo, con los productos de trabajo sin dependencias en la parte superior.

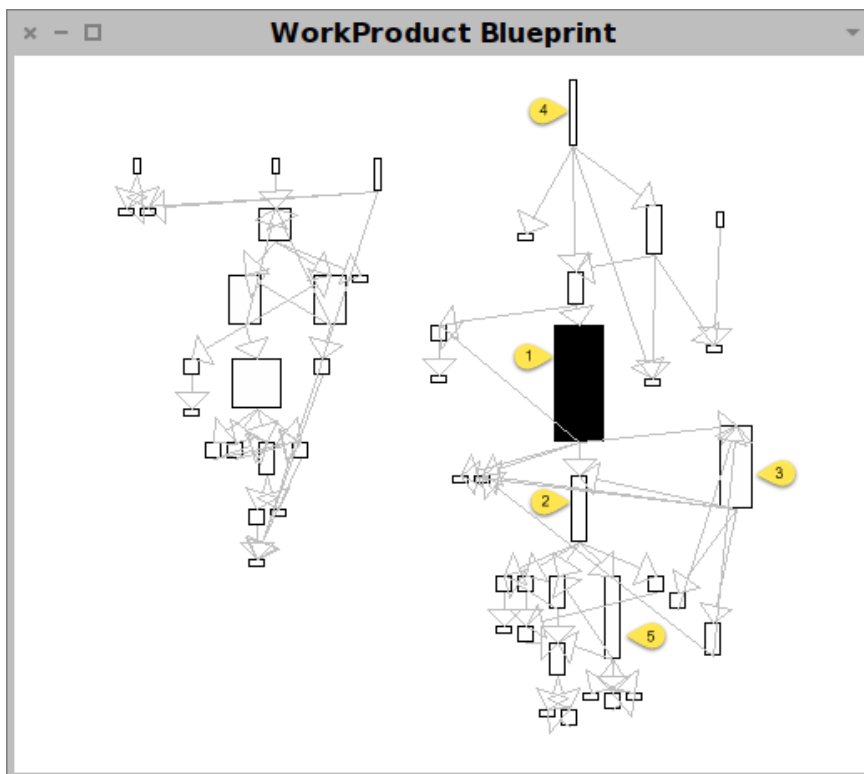


Figura 18–Visualización de los productos de trabajo del proceso en AVISPA.

En la figura número 18 se muestra la visualización de los productos de trabajo del modelo de desarrollo de software de PowerData. Una de las cosas que resalta a primera vista es que sólo uno de los productos de trabajo, “Documento de

requerimientos”, es de tonalidad oscura (la tarea número 1), es decir, sólo uno de ellos tiene definido guías para este producto. Esta guía corresponde a un template para este documento y algunas directrices sobre cómo tomar requerimientos. Idealmente sería mejor que la mayoría de los productos de trabajo tuviesen guías, ya que de esta forma es más fácil formalizar estos artefactos y existiría homogeneidad, independiente de qué persona realice el producto de trabajo y del proyecto.

De este mismo análisis también se puede observar cómo este documento de requerimientos forma una especie de cuello de botella para el resto de los productos de trabajo. Esto se observa porque es el nodo más alto del modelo, lo que por definición significa que mientras más alto, es entrada para un mayor número de tareas, y por lo tanto si se retrasa, causará un retraso en todos sus sucesores.

Al igual que el producto de trabajo número 1, existen otros cinco artefactos que igualmente son entrada para un gran número de tareas, estos productos de trabajo son:

1. **Documento de requerimientos:** Artefacto donde se registran los problemas y solicitudes que tiene el cliente y sobre los cuales se trabajará para resolverlos.
2. **Solución:** Son los entregables, los fuentes del proyecto que satisfacen la necesidad del cliente.
3. **Documento de diseño:** Artefacto que contiene el planteamiento del proyecto, el diseño sobre el cual será construida la solución.
4. **Documento de solicitud de cambio:** Documento formal pero sin template que contiene la solicitud formal de cambio; este documento es generado por el cliente.
5. **Documento de pruebas unitarias:** Documento que contiene las pruebas unitarias realizadas sobre el desarrollo de la solución.

Lo deseable para los nodos sería que fuesen más cuadrados que altos ya que de esta forma existirían menos tareas dependientes de cada uno de estos artefactos, y por lo tanto menor riesgo de retraso en caso de que surjan problemas con la creación de estos productos de trabajo. Adicionalmente se esperaría que existan más nodos oscuros ya que eso significa que tienen guías asociadas, lo que facilitaría de alguna forma la creación de estos artefactos. Esto es uno de los puntos que declara PowerData que desea modificar en su modelo.

Finalmente otra de las cosas que llama la atención del análisis del modelo de productos del proceso es que existen, aquí también, dos grandes grupos de

productos de trabajo que no están conectados entre sí, el de la derecha, donde se muestran las tareas mencionadas anteriormente, y el grupo de la izquierda, que corresponde a artefactos generados por el área de marketing, finanzas y ventas. Entre estos dos grupos no existe ningún producto de trabajo que una ambas áreas de la organización. Esto puede deberse a un sesgo al momento de modelar el proceso o bien que definitivamente este artefacto no exista, al igual como ocurre en el análisis de la vista de tareas, donde se presenta este mismo caso.

4.4 Diagnóstico

Del análisis estático del proceso, establecido por los patrones de error definidos por AVISPA se puede desprender lo siguiente:

En cuanto a los roles, el rol más sobrecargado es el jefe de proyectos: participa directamente en la confección de 3 productos de trabajo, pero está involucrado con 31 de las 45 tareas que componen el modelo. Esta situación genera riesgo dentro del proceso ya que si falla, todas las tareas asociadas a él dentro del proceso también fallarán al igual que los productos de trabajo que genera o modifica. Algo similar ocurre con otros roles que también están a cargo de ejecutar un gran número de tareas (ancho de los nodos) como el Delivery Manager y el Account Manager.

En cuanto a las tareas, existen sólo 4 que generan gran cantidad de productos de trabajo, lo que se puede considerar como bueno ya que múltiples salidas puede revelar que éstas no están especificadas con la granularidad apropiada. Sin embargo, estas tareas podrían ser muy complejas ya que su objetivo no es único.

Por el contrario, la mayoría de las tareas del modelo tienen como input múltiples productos de trabajo. El riesgo que esto puede generar es un potencial retraso de las tareas, ya que son necesarios varios productos de trabajo independientes entre sí para dar comienzo a estas tareas.

En lo que se refiere a la opacidad de los nodos de las tareas no existen demasiados de ellos con estas características. Alrededor de 5 nodos, y algunos de ellos, como la tarea número 11 (“Realizar reunión de lanzamiento”), no son predecesoras de ninguna otra tarea, lo que presumiblemente no afectaría el desarrollo del flujo del proceso.

En relación a las tareas aisladas, el modelo cuenta sólo con 3 tareas que tienen esta característica. Estas tres tareas no generan productos de trabajo que sean utilizados por otras tareas, ya que corresponden a actividades de verificación, notificaciones o recepciones que se marcan como realizados con envíos de correo o vistos buenos informales.

Se puede observar además que existen dos grandes grupos de tareas que no están unidos entre sí, uno corresponde a tareas relacionados con el desarrollo del proyecto, y el otro grupo a las tareas relacionadas con la administración, finanzas y marketing del mismo. Llama la atención que no exista una tarea que relacione ambas áreas del modelo entre ellas, esto puede deberse a una omisión al momento de modelar el proceso con PowerData.

Finalmente, el análisis de los artefactos de trabajo, muestra como ya se ha señalado, que el documento de requerimientos asoma como un posible problema en el modelo del proceso. Se puede ver en la figura 18 como AVISPA grafica este artefacto como el más largo del modelo, lo que quiere decir que muchas tareas lo utilizan como entrada, evidenciando un potencial retraso de esas tareas en caso de que el artefacto se retrase o presente problemas.

Otro hallazgo encontrado en el análisis realizado con AVISPA, es que el único artefacto que tiene una guía asociada es el documento de requerimientos. Esto se expresa con el color del nodo, mientras más oscuro es el nodo, mayor cantidad de guías existentes y viceversa. El hecho de que los otros artefactos del modelo no tengan guías o modelos para su construcción supone una posible falta de alineamiento o desorganización entre los roles responsables de su elaboración.

A continuación, se muestra una tabla resumen de los hallazgos encontrados con AVISPA que representan posibles errores de oportunidades de mejora dentro del modelo de desarrollo de software de la empresa.

Roles	Descripción del posible error
<i>Jefe de proyectos</i>	Es graficado por AVISPA como el nodo más ancho dentro de los roles, lo que significa que es el que participa en más tareas. Esto genera riesgos potenciales de retraso ya que retrasándose las tareas de este rol no podrían comenzar las siguientes.
<i>Delivery manager</i> <i>Account manager</i>	Estos dos roles, luego del jefe de proyectos, son los de mayor ancho en la representación de AVISPA, lo que al igual que el caso anterior pudiesen generar riesgo de retrasos de las tareas en las que participan.
Tareas	Descripción del posible error
<i>Hacer seguimiento comercial de una propuesta</i>	Estas 4 tareas tienen en común que generaran gran cantidad de productos de trabajo, lo que se grafica como el ancho de los nodos en AVISPA,
<i>Validar alcance</i>	
<i>Notificar propuesta aprobada</i>	

<i>Verificar diseño de la solución</i>	múltiples salidas pueden revelar que éstas no están especificadas con la granularidad apropiada. Estas tareas podrían ser muy complejas ya que su objetivo no es único y podrían generar un retraso en el proyecto.
<i>Planificación de proyecto</i>	Lo que tienen en común estas 4 tareas, es que son graficadas en AVISPA como los nodos más largos, lo que significa que requieren de muchos inputs distintos para comenzar dichas tareas, esto representa un potencial riesgo de retraso.
<i>Validar diseño de la solución</i>	
<i>Coordinar trabajo</i>	
<i>Generar documento de cierre</i>	
<i>Validar alcance</i>	
<i>Validar diseño de la solución</i>	El factor común que agrupa a estas 5 tareas, es la opacidad con la que son representadas por AVISPA, siendo ellas las más oscuras, lo que significa que son las tareas donde participan más roles de manera simultánea, esto genera un factor de riesgo debido a la necesidad de coordinar múltiples roles para avanzar la actividad.
<i>Realizar reunión de lanzamiento</i>	
<i>Generar documento de pruebas integrales</i>	
<i>Identificar oportunidad</i>	
<i>Identificar oportunidad</i>	
Artefactos	Descripción del posible error
<i>Documento de requerimientos</i>	Al ser el nodo más largo representado con AVISPA, el principal riesgo con este artefacto es que genere un cuello de botella para las tareas que lo suceden.

Tabla 2, resumen de análisis estático con AVISPA.

5 Aplicación del modelo en proyectos piloto

Un vez que el proceso de desarrollo de software de PowerData fue modelado y analizado, el siguiente paso fue implementarlo en dos proyectos realizados por la compañía. Esto tiene como objetivo probar el modelo definido y obtener métricas de tiempo real y de desempeño del proceso.

Estos proyectos piloto fueron realizados en dos instituciones bancarias: el Banco Central de Chile y el Banco Estado. Los proyectos en estas empresas fueron seleccionados por sus características de duración, ya que se debían desarrollar próximos a este trabajo de tesis. Junto con lo anterior, además se seleccionaron a estos clientes por ser empresas con las que ya se había trabajado anteriormente y de forma exitosa, por lo tanto, la empresa ya conocía el negocio de ambos clientes.

La metodología de implementación del modelo definido y el análisis posterior que se obtuvo con los resultados de los proyectos piloto fue la siguiente:

- Planificación y estimación de los proyectos.
- Ejecución de los proyectos y registro de logs.
- Comparación de logs de ejecución versus la planificación.
- Identificación de elementos problema encontrados en la ejecución de los proyectos piloto.
- Comparación de los elementos problema con los patrones de error encontrados con el análisis de AVISPA.

5.1 Proyecto SIETE, Banco Central de Chile

El proyecto SIETE fue desarrollado en el departamento de estadísticas del Banco Central, y tuvo como objetivo validar los indicadores económicos entregados mensualmente por el Instituto Nacional de Estadísticas (INE).

Específicamente se necesitaba rescatar los datos entregados por el INE en formato Excel y como archivos planos, transformarlos, y validarlos para luego volverlos a cargar en el Datamart del departamento de estadísticas con el fin de que estuviesen disponibles para su consumo por otras aplicaciones del banco.

La tecnología utilizada para realizar esta solución fue la herramienta ETL informática PowerCenter.

5.1.1 Planificación proyecto SIETE

La duración total programada de este proyecto fue de 120 días. En el cronograma se intentó aplicar lo más apegado el modelo de proceso de software definido por la empresa, como se puede ver en la figura 19.

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		[-] Cronograma Proyecto	119.8 días	lun 08/12/14	vie 22/05/15	
2		Firma e inicio Proyecto	1 día	lun 08/12/14	lun 08/12/14	
3		[-] Actividades	63.8 días	lun 08/12/14	jue 05/03/15	
4		Confección Plan de Pruebas Detallado	5 días	lun 08/12/14	vie 12/12/14	2
5		Revisión Plan de Pruebas Detallado	2 días	lun 15/12/14	mié 17/12/14	4
6		[-] Análisis	17 días	mié 17/12/14	vie 09/01/15	
7		[-] Análisis Detallado de Requerimiento	17 días	mié 17/12/14	vie 09/01/15	
8		Analizar requisitos de implementación	5 días	mié 17/12/14	mié 24/12/14	5
9		Verificación de requisitos	3 días	mié 24/12/14	lun 29/12/14	8
10		Validación de requisitos	8 días	lun 29/12/14	jue 08/01/15	9
11		Revisión Libro de Requerimiento ETL	1 día	jue 08/01/15	vie 09/01/15	10
12		HITO Aprobación Libro de Requerimiento	0 días	vie 09/01/15	vie 09/01/15	11
13		[-] Diseño	7 días	vie 09/01/15	mar 20/01/15	
14		Diseño de la solución	3 días	vie 09/01/15	mié 14/01/15	12
15		Verificación de la solución	2 días	mié 14/01/15	vie 16/01/15	14
16		Validación de la solución	1 día	vie 16/01/15	lun 19/01/15	15
17		HITO - Aprobación del BCCh del diseño Propuesto	1 día	lun 19/01/15	mar 20/01/15	16
18		Hito 1: Requerimientos y Diseño	0 días	mar 20/01/15	mar 20/01/15	17
19		[-] Preparación Ambiente Desarrollo	7 días	mié 17/12/14	vie 26/12/14	
20		Habilitación de Ambientes de Desarrollo	5 días	mié 17/12/14	mié 24/12/14	5
21		Análisis/Revisión de Configuración de los ambientes	2 días	mié 24/12/14	vie 26/12/14	20
22		[+] Construcción ETL	21 días	mar 20/01/15	mié 18/02/15	
72		[-] Pruebas Proveedor	11 días	mié 18/02/15	jue 05/03/15	
73		Pruebas Integrales	5 días	mié 18/02/15	mié 25/02/15	71
74		Pruebas de Stress	2 días	mié 25/02/15	vie 27/02/15	73
75		Documentación Técnica Procesos ETL	4 días	vie 27/02/15	jue 05/03/15	74
76		HITO: Término Pruebas Proveedor	0 días	jue 05/03/15	jue 05/03/15	75
77		[+] Certificación BCCh	34 días	jue 05/03/15	mié 22/04/15	
96		[+] Puesta en Producción	22 días	mié 22/04/15	vie 22/05/15	
103		Hito 3: Pruebas, Pruebas QA y Puesta en Producción	0 días	mié 20/05/15	mié 20/05/15	101
104		[+] Marcha Blanca	44 días	vie 22/05/15	jue 23/07/15	
106		[+] Periodo de Garantía	87 días	jue 23/07/15	lun 23/11/15	

Figura 19 – Cronograma proyecto SIETE.

5.1.2 Logs proyecto SIETE

A continuación, en la tabla 2, se presentan las métricas dinámicas obtenidas del modelo de desarrollo aplicado a este proyecto. En la primera columna se muestra la actividad, en la segunda la duración programada de acuerdo al cronograma y finalmente en la tercera columna se muestra el tiempo real de la actividad, medido directamente durante la realización del proyecto.

Actividad	Días programados	Tiempo real (días)
PLANIFICACIÓN DEL PROYECTO	8	8
Planificación del proyecto	5	5
Verificación de la planificación	2	2
Informar a ADM y Finanzas	1	1
Notificar involucrados	1	1
CONFECCIÓN DE PLANES DE PRUEBA	7	9
Confección Plan de Pruebas Detallado	5	7
Revisión Plan de Pruebas Detallado	2	2
ANÁLISIS	17	37
Analizar requisitos de implementación	5	12
Verificación de requisitos	3	8
Validación de requisitos	8	15
Revisión Libro de Requerimiento ETL	1	2
DISEÑO	14	20
Diseño de la solución	3	5
Verificar diseño de la solución	2	4
Validar diseño de la solución	1	3
HITO - Aprobación del diseño Propuesto	1	1
Habilitación de Ambientes de Desarrollo	5	7
Análisis/Revisión de Configuración de los ambientes	2	1
DESARROLLO	40	80
Coordinar trabajo	1	1
Webservice de llamado a procesos ETL PowerCenter	4	7
Proceso estándar de lectura archivos Excel	4	8
Proceso estándar de lectura archivos CSV	4	4
Proceso estándar de lectura archivos Access	4	6
Proceso especial de lectura archivos IPC	4	12
Proceso especial de lectura archivos IPM	4	10
Reporte de procesos de carga exitosa	4	5
Reportes de Aprobación	4	5
Reporte procesos de publicación	4	6
Generación Informe estado datos	4	7
Envíos de correos automáticos	4	4

Reporte de Auditoría de Eventos	4	5
PRUEBAS Y AMBIENTACIÓN	45	49
Pruebas Integrales - proveedor	5	5
Pruebas de stress - proveedor	2	2
Documentación técnica de los procesos - proveedor	4	3
Traspaso a Ambiente de Prueba - Banco	6	9
Pruebas Usuario - Banco	5	7
Pruebas QA - Banco	23	23
PUESTA EN PRODUCCIÓN	22	25
Documentación Traspaso de Ambientes	7	5
Comités de Certificación y Cambios	5	8
Traspaso de componentes desarrollados	5	5
Generar documento de cierre	3	3
Capacitación Técnica-Funcional	2	4
TOTAL	162	225

Tabla 2, Log proyecto SIETE Banco central de Chile.

5.1.3 Desviaciones proyecto SIETE

Como se puede ver en el log del proyecto SIETE, existió una clara desviación respecto a lo planificado. En este análisis se dividió el proyecto en fases, y la medición del tiempo real se hizo en función de lo mismo.

Se puede ver que las mayores desviaciones se presentan en la etapa de análisis y desarrollo, concretamente en las tareas de diseño de la implementación y desarrollo de la solución. Durante la ejecución del proyecto se pudo observar además que en la etapa de análisis, específicamente durante la creación del documento de requerimientos se tuvieron 5 iteraciones. En relación a la etapa de desarrollo de la implementación, esta tuvo un retraso concreto de 31 días hábiles y lo que se observó durante la toma de métricas en terreno fue que existieron muchos problemas de definiciones confusas o interpretables, obtenidas desde el mismo documento de requerimientos.

5.2 Proyecto Datamart Multicanal, Banco Estado de Chile

El proyecto Datamart Multicanal, llevado a cabo en el Banco Estado de Chile, tuvo como propósito extraer información del Data Warehouse corporativo de la institución, trabajar en la integridad de los datos extraídos y aplicar sobre ellos reglas de negocios definidas por el área de multicanales del banco para luego ser cargados en el Datamart que lleva el mismo nombre del proyecto. Este Datamart

sería explotado diariamente con el fin de obtener información del comportamiento de los clientes que utilizan las plataformas web del banco.

La tecnología utilizada para realizar esta solución fue la herramienta ETL informática PowerCenter.

5.2.1 Planificación del proyecto Datamart Multicanal

La duración total programada de este proyecto fue de 60 días, sin embargo, el tiempo total real del proyecto fue de 113 días. En el cronograma se intentó aplicar lo más apegado el modelo definido por la empresa.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	PILOTO	60.6 días	lun 12/01/15	lun 06/04/15	
2	BATCH	60.6 días	lun 12/01/15	lun 06/04/15	
3	Actividades BACTH	45.6 días	lun 12/01/15	lun 16/03/15	
4	Análisis	4 días	lun 12/01/15	jue 15/01/15	
5	Análisis Detallado de Requerimiento	4 días	lun 12/01/15	jue 15/01/15	
6	Analizar requisitos de implementación	1 día	lun 12/01/15	lun 12/01/15	
7	Verificación de requisitos	1 día	mar 13/01/15	mar 13/01/15	6
8	Validación de requisitos	2 días	mié 14/01/15	jue 15/01/15	7
9	HITO Aprobación Libro de Requerimiento	0 días	jue 15/01/15	jue 15/01/15	8
10	Casos de Pruebas	3 días	vie 16/01/15	mar 20/01/15	
11	Confección Plan de Pruebas Detallado	2 días	vie 16/01/15	lun 19/01/15	8
12	Revisión Plan de Pruebas Detallado	1 día	mar 20/01/15	mar 20/01/15	11
13	Diseño	2.5 días	vie 16/01/15	mar 20/01/15	
14	Diseño de la solución	1 día	vie 16/01/15	vie 16/01/15	9
15	Verificar diseño de la solución	0.5 días	lun 19/01/15	lun 19/01/15	14
16	Validar diseño de la solución	1 día	lun 19/01/15	mar 20/01/15	15
17	HITO - Aprobación del BcoEstado del diseño Propuesto	0 días	mar 20/01/15	mar 20/01/15	16
18	Hito: Requerimientos y Diseño	0 días	mar 20/01/15	mar 20/01/15	17
19	Preparación Ambiente Desarrollo	0.5 días	lun 19/01/15	lun 19/01/15	
20	Habilitación y Configuración Ambientes de Desarrollo	0.5 días	lun 19/01/15	lun 19/01/15	15
21	Construcción ETL	34.6 días	mar 20/01/15	lun 09/03/15	
22	Desarrollos Etapa I - BATCH	34.6 días	mar 20/01/15	lun 09/03/15	
23	1 - Mapeo TDTC a TDMT_SGG	12.6 días	mar 20/01/15	jue 05/02/15	
87	2 - Mapeo TDMT_SGG TDMT_TRH	22 días	jue 05/02/15	lun 09/03/15	
152	3 - Mapeo Log tech a TDMT_SGG	1 día	vie 06/03/15	lun 09/03/15	
157	4 - Mapeo planos LT a TDMT_SGG	1 día	vie 06/03/15	lun 09/03/15	
162	Hito: Término Construcción y Pruebas Unitarias	0 días	lun 09/03/15	lun 09/03/15	161
163	Pruebas Proveedor	5 días	lun 09/03/15	lun 16/03/15	
167	HITO: Término Pruebas Proveedor	0 días	lun 16/03/15	lun 16/03/15	163
168	Certificación BcoEstado	6 días	lun 16/03/15	mar 24/03/15	
187	Puesta en Producción	8 días	vie 13/03/15	mié 25/03/15	
193	Hito: Puesta en Producción	0 días	lun 23/03/15	lun 23/03/15	191
194	Marcha Blanca	10 días	lun 23/03/15	lun 06/04/15	
196	Periodo de Acompañamiento	132 días	lun 06/04/15	mié 07/10/15	194

Figura 20 – Cronograma proyecto Datamart Multicanal.

5.2.2 Logs proyecto Datamart Multicanal

A continuación, se presentan las métricas dinámicas obtenidas del modelo de desarrollo aplicado a este proyecto. En la primera columna se muestra la actividad, en la segunda la duración programada de acuerdo al cronograma y finalmente en la tercera columna se muestra el tiempo real de la actividad, medido directamente durante la realización del proyecto.

Actividad	Tiempo programado (días)	Tiempo real (días)
PLANIFICACIÓN DEL PROYECTO	8	8
Planificación del proyecto	5	5
Verificación de la planificación	2	2
Informar a ADM y Finanzas	1	1
Notificar involucrados	1	1
CONFECCIÓN DE PLANES DE PRUEBA	3	4
Confección plan de pruebas detallado	2	3
Revisión plan de pruebas detallado	1	1
ANÁLISIS	5	12
Analizar requisitos de implementación	1	3
Verificación de requisitos	1	2
Validación de requisitos	2	4
Preparación de ambiente de desarrollo	1	3
DISEÑO	3	11
Diseño de la solución	1	1
Verificar diseño de la solución	1	3
Validar diseño de la solución	1	6
Revisión comité arquitectura	1	1
DESARROLLO	37	44
Coordinar trabajo	1	1
Mapeo de datos desde DWH a DMT Sybase	12	11
Mapeo de archivos de datos planos a DMT Sybase	1	2
Mapeo de archivos de log a DMT Sybase	1	6
Mapeo de datos desde DMT Sybase a DMT SQL Server	22	24
PRUEBAS Y CERTIFICACIÓN	27	32
Pruebas integrales - proveedor	5	9
Pruebas de estrés - proveedor	2	1

Documentación técnica de los procesos - proveedor	4	4
Traspaso a ambiente de QA - Banco	2	2
Pruebas usuario - Banco	4	6
Traspaso a ambiente de pre-producción - Banco	5	5
Pruebas técnicas QA - Banco	5	5
PUESTA EN PRODUCCIÓN	13	16
Documentación traspaso de Ambientes - proveedor	7	5
Traspaso de Componentes	1	1
Generar documento de cierre	3	5
Capacitación técnica-Funcional	2	5
TOTAL	96	126

Tabla 3, Log proyecto Datamart Multicanal, Banco Estado.

5.2.3 Desviación proyecto Datamart Multicanal

En el caso del proyecto Datamart multicanal, también existió una desviación entre el cronograma programado versus la medición de tiempo real tomada en terreno, y al igual que en el piloto anterior, las desviaciones más importantes se presentan en las tareas de análisis de requisitos de implementación y desarrollo de la solución, tomando en el caso de análisis, más del doble del tiempo programado.

6 Análisis y Discusión de Resultados

Luego del estudio de las métricas estáticas, la revisión de logs y el análisis de las desviaciones de los proyectos pilotos, se pueden proponer algunas oportunidades de mejora en relación al proceso definido.

6.1 Roles sobrecargados

Al analizar estáticamente el proceso, específicamente los roles involucrados en él, llama la atención lo sobrecargado que se encuentra el rol de jefe de proyecto, que participa activamente en el 70% de las tareas que componen el modelo.

Lo anterior queda de manifiesto en el análisis empírico del modelo que se comprobó en los proyectos piloto, donde se vio que el jefe de proyectos estaba muy sobre demandado durante todo el ciclo de vida de ambas iniciativas, sobre todo durante la etapa de desarrollo donde tenía que participar activamente en la definición y revisión de los documentos de diseño y requisitos que fueron los artefactos que más se retrasaron, ya que a su vez son salidas de las tareas más retrasadas.

Sería una buena idea revisar la carga de trabajo que tiene este rol y estudiar la real necesidad de que esté presente en todas las actividades en las que hoy está.

6.2 Productos de trabajo que son input de muchas tareas

Otro de los hallazgos encontrados, es el cuello de botella que genera el documento de requerimientos. Esto es comprobable en la práctica ya que en ambos proyectos piloto el retraso de este artefacto significó un retraso considerable en el cronograma del proyecto en general.

Quedó en manifiesto en los pilotos, que además no existían criterios unificados para generar el artefacto. Sería recomendable evaluar la posibilidad de construir el documento en incrementos, y aquellos incrementos que sean validados por el cliente puedan ser liberados para que se continúe con el flujo de desarrollo. Lo mismo puede aplicar para la generación de otros artefactos como el documento de diseño.

Al revisar el guidance que tiene la empresa para la construcción del documento de requerimientos, se aprecia que es poco específico y ambiguo en la definición del contenido que debería tener. Además no cuenta con un template que sirva de modelo para el desarrollo del artefacto.

6.3 Tareas con muchos inputs

Siguiendo con las tareas, otro de los puntos relevantes son aquellas que tienen como input muchos productos de trabajo, es decir, aquellos nodos largos. Existen muchas tareas con esta característica, destacando entre ellas “Planificación de proyecto”, “Validar diseño de la solución”, “Coordinar trabajo” y “Generar documento de cierre”. Se debería revisar estas tareas para validar que efectivamente son necesarios todos los artefactos de entrada para cada una de ellas, ya que existen artefactos que pudiesen no ser relevantes para el inicio de una actividad.

En el análisis empírico se descubrió que efectivamente una de las tareas con más desviación fue la de “Validar diseño de la solución” y “Analizar requisitos de implementación”. Ambas tareas ya habían sido identificadas en el análisis estático como potencialmente problemáticas, pertenecen a la etapa de desarrollo del proyecto, y las dos sufrieron retrasos en los proyectos piloto.

6.4 Otros

Al analizar la manera en que la empresa desarrolla software se puede ver que siguen por instinto o no, un modelo de desarrollo en cascada. Dada la naturaleza de las soluciones que generan y las múltiples iteraciones en las que se incurre,

sería válido el evaluar integrar otro modelo de desarrollo, como un desarrollo incremental o por prototipos, y evaluar su impacto en proyectos similares.

7 Conclusiones

Desde los últimos 3 años, la empresa PowerData Chile, ha experimentado constantes pérdidas en su área de consultoría, lo que ha venido de la mano con menor productividad, pérdida de clientes y fuga de colaboradores. Lo anterior se especulaba era por un proceso de desarrollo de software poco adecuado y roles sobrecargados, lo que se fue comprobando a medida que se desarrolló este trabajo.

Para comprobar estos supuestos se trabajó definiendo el proceso actual de desarrollo de software de la empresa, a partir de entrevistas con los diferentes involucrados en cada etapa del ciclo de vida de los proyectos, y se modeló este proceso utilizando la herramienta EPF Composer para luego poder medir estáticamente el modelo y aplicarlo en dos proyectos pilotos.

Durante el análisis estático, se pudo comprobar que existen roles como el jefe de proyectos que se ve sobrepasado en sus responsabilidades. En los dos proyectos pilotos este rol sufrió atrasos en su gestión como parte responsable de tareas y artefactos como “Planificación de proyecto”, “Validar diseño de la solución” y “Coordinar trabajo”, consistentemente con lo obtenido del análisis estático. Por otro lado, los consultores, en ambos proyectos se vieron retrasados principal y notoriamente en la generación del “Documento de requerimientos”, hito clave para comenzar con el desarrollo de la solución, esto produjo retraso en ambos pilotos. Este documento se había encontrado, como parte del análisis estático, como un potencial cuello de botella, y así fue.

Otro descubrimiento fueron aquellas tareas que tienen como input muchos productos de trabajo, es decir, aquellos nodos representados de manera larga en AVISPA. Existen muchas tareas con esta característica, destacando entre ellas “Planificación de proyecto”, “Validar diseño de la solución”, “Coordinar trabajo” y “Generar documento de cierre”. Durante la aplicación del modelo en los proyectos piloto se comprobó que estas tareas efectivamente sufrieron los mayores retrasos.

Para mejorar el proceso de desarrollo de software de PowerData se podría revisar aquellos hallazgos que surgieron en la revisión estática y dinámica. Dada la naturaleza de las soluciones en las que trabaja la compañía, una buena idea sería evaluar un modelo de desarrollo de software incremental o por prototipos, y evaluar su impacto en proyectos similares, ya que el actual modelo en cascada

que utiliza la empresa, propicia cuellos de botella como lo es, por ejemplo, el artefacto “documento de requerimientos”. Adicionalmente se podría revisar el rol de jefe de proyectos, ya que se comprobó en la práctica que también ocasionó retrasos durante la ejecución de los proyectos piloto debido a atrasos en su gestión.

Al analizar los resultados de las mediciones estáticas y compararlos con los datos obtenidos de las mediciones empíricas, se puede concluir que los patrones de posibles errores y los predictores obtenidos con AVISPA, coinciden muy bien con los resultados de la aplicación del modelo en los proyectos piloto. Al entregar un asertivo análisis del modelo, sería posible extrapolar esta experiencia a otras empresas de tecnologías con características similares a PowerData, y a partir de este análisis, evaluar su proceso de desarrollo de software enfocándose en los patrones que entrega AVISPA.

8 Bibliografía

1. **Piattini Velthuis, M. G., García Rubio, F. Ó., Garzás Parra, J., Genero Bocco, M. F. 2008.**Medición y estimación del software.
2. **Fenton, N. E., Pfleeger, S. L., - Boston, MA, USA, 1997.**Software Metrics: A Rigorous and Practical Approach.
3. **Ruiz, Francisco y Verdugo, Javier, 2008.** Guía de Uso de SPEM 2 con EPF Composer v3.0. La Mancha, España :Grupo Alarcos, 2008.
4. **J. A. Hurtado, A. Bergel, and M. C. Bastarrica, 2011.** Analyzing Software Process Models with AVISPA. Proceedings of the 2011 International Conference on Software and Systems Process. págs. 23-32.
5. **Silvia T. Acuña, Angélica de Antonio, Xavier Ferré, Luis Maté y Marta López, 2001.** **The Software Process: Modelling, Evaluation and Improvement.** [ed.] **S. K. Chang. New Jersey, EE.UU.** : Handbook of Software Engineering and Knowledge Engineering, 2001, págs. 193-237.