
Modelado de una herramienta web para el seguimiento integral del ciclo productivo de aplicaciones bajo el esquema de fábricas de software en entidades bancarias

MODELING WEB TOOL FOR MONITORING INTEGRAL PRODUCTIVE CYCLE UNDER THE SCHEME APPLICATION SOFTWARE FACTORY IN BANKS

Ana María Huayna Dueñas¹, Karina Alexandra Rodríguez Castro²

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática

¹ahuaynad@unmsm.edu.pe, ²krodriquezc@grupobbva.com.pe

ReSUMeN

En el sector bancario existe gran demanda de sistemas de información nuevos o mejora de los ya existentes, los cuales deben ser atendidos en el menor tiempo posible, con una exactitud y eficiencia con las que verdaderamente estos desarrollos se conviertan en herramientas de apoyo. Estos sistemas dan soporte a las operaciones comerciales, a la gestión interna, propician el crecimiento eficiente de sus negocios e incrementan valor para sus clientes. Por ello se requiere de una buena coordinación y regulación cuando estos desarrollos son realizados por terceros.

El presente artículo consiste en realizar una revisión de los modelos de calidad existentes enfocándonos en el proceso de adquisición de servicios de terceros, específicamente aplicado al esquema de fábrica de software, y sobre la base de este modelo, realizar el análisis y modelado de una herramienta web cuyo objetivo es mejorar de forma integral, el proceso de desarrollo y mantenimiento de aplicaciones especializadas, en entidades bancarias, teniendo en cuenta las normas que regulan la producción de software en este sector.

Palabras clave: fábrica de software, proceso de software, modelos de calidad, arquitectura web

AbStRACt

In the banking sector there is great demand for new information systems or upgrading existing ones, which must be addressed as quickly as possible, with such accuracy and efficiency that these developments actually become support tools. These systems support business operations, internal management, foster the efficient growth of their business and increase value for its customers. Therefore requires good coordination and control when these developments are made using outsourcing.

This work consists on conducting a review of existing quality models focusing on the acquisition of third party service, specifically applied to the software factory schema and based on this model, perform the analysis and modeling of a web tool aimed at improving the process of development and maintenance of specialized applications in banking, taking into account the rules governing the production of software in this sector.

Keywords: software factory, software process, quality models, web architecture

1. INTRODUCCIÓN

Actualmente, gran parte de los sectores económicos se encuentran en constante crecimiento y desarrollo. Este crecimiento se debe a la necesidad de las empresas de contar con ventajas competitivas que los posicionen dentro de la concepción de sus clientes.

El sector bancario no es la excepción. Cada vez más, este sector destina gran parte de sus inversiones en el desarrollo y la adquisición de aplicaciones especializadas que proporcionan soporte a las operaciones comerciales, gestión interna, al crecimiento eficiente de sus negocios e incrementen valor a sus clientes.

Nos encontramos con un entorno dinámico y competitivo, en el que la demanda de soluciones tecnológicas se incrementa a gran velocidad. Por ello, la atención a esta demanda de soluciones debe ser ágil y fluida, pero al mismo tiempo, asegurando la calidad de los productos que son entregados a los clientes.

En el presente trabajo pretendemos determinar el modelo de calidad de desarrollo de software que mejor se ajuste a las necesidades del entorno bancario; específicamente a la realidad de PERUBANK¹, empresa líder en el sector Banca y en base a este modelo, realizar el análisis y diseño de una herramienta web que ayude a mejorar la calidad de los productos de software entregados bajo el esquema de fábrica de software. Para la solución al problema se optó en mejorar la Metodología Corporativa de PERUBANK alineándola al Modelo CIMMI y con esto, se logró enriquecer el proceso del Ciclo Productivo de Aplicaciones en la Unidad de Sistemas del Banco.

En la sección 2, describiremos los aspectos más relevantes sobre la Ingeniería de software (procesos de software, modelos de mejora de procesos, modelos de referencia del ciclo de vida del software, metodologías de desarrollo de software, fábricas de software y modelos de gestión de proyectos), en la sección 3 Presentamos la Metodología a seguir es decir el modelo a desarrollar y la herramienta teórica a utilizar, en la sección 4 se presentan los requerimientos y diseño de la aplicación; mientras que la sección 5 las conclusiones, y finalmente en la 6 se presentan las referencias bibliográficas utilizadas.

2. FUNDAMENTACIÓN TEÓRICA

2.1. Ingeniería de Software

La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este; después que se utiliza [1].

La Ingeniería de software es una actividad para la adquisición de conocimiento [2].

2.1.1. Proceso de Software

Un proceso de software es un conjunto de actividades y resultados asociados que producen un producto de software. Existen cuatro actividades fundamentales comunes a todos los procesos de software, que son:

- **Especificación del software**, donde los clientes e ingenieros definen el software a producir y las especificaciones de su operación.
- **Desarrollo de software**, donde el software se diseña y se programa.
- **Validación del software**, donde el software es validado para asegurar que es lo que el cliente requiere.
- **Evolución del software**, donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

2.1.2. Modelos del Proceso de Software

Son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. [1]. Existen tres modelos del Proceso de software:

- **Modelo en cascada**. El modelo de cascada se utiliza sobre todo para grandes proyectos de ingeniería de sistemas donde se desarrolla un sistema en varios lugares [1].
- **Modelo evolutivo**. El desarrollo evolutivo se basa en la idea de desarrollar una implementación inicial exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado.

¹ Debido a las políticas de la empresa, se creó un nombre ficticio para referirnos a ella, por lo tanto, su razón social no será presentada en este documento.

- **Ingeniería de software basado en componentes.** Se basa en la reutilización de componentes de una aplicación base de componentes de software reutilizables y de algunos marcos de trabajo de integración para estos lugares [1].

2.1.3. Modelos de Mejora de Procesos

- **CMMI (Capability Maturity Model Integration)**
CMMI es un modelo que ayuda a integrar las funciones de la organización, conducir la mejora de los procesos, proveer una guía de calidad de los procesos y proporcionar puntos de referencia para la evaluación de estos [3], [7].
- **Modelo de Calidad ISO 9001:2008**
La norma ISO 9001:2008 es un modelo de gestión de la calidad con un enfoque en procesos en el que juega un papel importantísimo, entre otros aspectos, la satisfacción del cliente y la mejora continua. Sus requisitos son flexibles y fácilmente adaptables a las necesidades y características de cada organización [4].
- **Norma ISO/IEC 15504**
La norma ISO 15504 SPICE es una norma abierta e internacional para evaluar y mejorar la capacidad y madurez de los procesos. Junto con la ISO 12207, la norma aplica a la evaluación y mejora de la calidad del proceso de desarrollo y mantenimiento de software [5].

2.1.4. Modelos de Referencia del Ciclo de Vida del Software

- **NTP (Norma Técnica Peruana - ISO/IEC 12207)**
La Norma Técnica Peruana basada en ISO/IEC 12207-2006, presenta los procesos del ciclo que se pueden emplear para adquirir, suministrar, desarrollar, operar y mantener productos de software. Agrupa las actividades que se pueden llevar a cabo en el ciclo de software, en cinco procesos principales, ocho procesos de apoyo y cuatro organizativos. Cada proceso está dividido en un conjunto de actividades; y cada actividad se subdivide a su vez en un conjunto de tareas [6].

2.1.5. Metodologías de desarrollo de Software

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas me-

todológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos las metodologías tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y las notaciones que se usarán [17].

Otras en cambio se centran en el factor humano en el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad [17].

A. Metodologías tradicionales

RUP (Rational Unified Process)

Es un proceso de ingeniería de software. Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales, dentro de un horario predecible y presupuesto [8].

El Proceso Unificado, está basado en componentes y por tanto, según este método un sistema de software está formado por componentes de software interconectados mediante interfaces bien definidas. Se utiliza el Lenguaje Unificado de Modelado UML para preparar todos los esquemas de un sistema de software. Además, RUP se caracteriza por estar guiado por los casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

El RUP describe cómo implementar efectivamente enfoques comercialmente probados de desarrollo de software para equipos de desarrollo de software y proporciona a cada miembro del equipo las directrices, modelos, herramientas necesarias para que todo el equipo aproveche al máximo entre otras, sus mejores prácticas [9].

El RUP divide un ciclo de desarrollo en cuatro fases consecutivas [10]: inicialización, elaboración, construcción y transición.

Métrica Versión 3

Esta metodología contempla el desarrollo de sistemas de información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que aseguran que un proyecto cumple sus objetivos en términos de calidad, coste y plazos. Así mismo, cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de la Configuración Aseguramiento de la Calidad y Seguridad [11]. Métrica 3 divide el ciclo de vida del desarrollo de software en tres procesos: Planificación, Desarrollo y Mantenimiento del Sistema de Información. En cada uno de los procesos se establecen actividades específicas a realizar cuando se utiliza la Métrica 3.

b. Metodologías ágiles

XP (extreme Programming)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

El ciclo de vida ideal de XP consiste de seis fases [12]: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

SCRUM

Scrum es un framework para el desarrollo de productos y sistemas complejos. Se basa en la teoría empírica de control de procesos. Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad y control de riesgos. Dentro de cada iteración, Scrum emplea equipos transversales con capacidad de auto-organización en el desarrollo del trabajo, para optimizar la flexibilidad y la productividad.

El corazón de Scrum es un *Sprint*. Un *Sprint* es una iteración de un mes o menos que es consistente a

lo largo de todo el esfuerzo de desarrollo. Todos los *Sprints* utilizan el mismo marco de Scrum, y todos los sprints finales, con un incremento del producto final que es potencialmente liberable [13].

2.1.6. Fábricas de Software

Fábrica de Software es una organización que produce soluciones industriales de calidad mundial bajo un modelo de rentabilidad, planificación, diseño y gestión. A diferencia de los programas informáticos masivos, este tipo de fábricas ofrece soluciones "a la medida" para las empresas [14].

Su propósito es brindar mejoras de fabricación en el software, reducir errores dentro del ciclo de vida de los productos y contribuir al time to market: el tiempo que va desde la generación de una idea de un nuevo producto o servicio hasta que este alcanza el mercado.

Podemos mencionar como un conjunto de buenas prácticas aplicables a un buen número de fábricas de Software: la integración continua y énfasis en la gestión de la configuración; control de calidad exhaustivo, periódico, automático; énfasis en el diseño basado en el conocimiento; utilizar casos de uso como pieza esencial del desarrollo; ciclo de vida iterativo y evolutivo; estimación basada en puntos de función avanzado y guiar las acciones por valor [14].

2.2. Modelos de Gestión de Proyectos

2.2.1. PMBoK (Project Management base of Knowledge)

La finalidad principal de la Guía del *PMBOK* es identificar el subconjunto de Fundamentos de la Dirección de Proyectos generalmente reconocido como buenas prácticas.

Esta norma describe la naturaleza de los procesos de dirección de proyectos en términos de su integración, las interacciones dentro de ellos, y sus propósitos. Estos procesos se dividen en cinco grupos, definidos como los Grupos de Procesos de la Dirección de Proyectos [15]:

- Grupo de Procesos de Iniciación
- Grupo de Procesos de Planificación
- Grupo de Procesos de Ejecución
- Grupo de Procesos de Seguimiento y Control
- Grupo de Procesos de Cierre.

Asimismo, estos procesos se agrupan en nueve áreas del conocimiento de la Dirección de Proyectos, que son:

- Integración
- Alcance
- Tiempo
- Costo
- Calidad
- Recursos Humanos
- Comunicaciones
- Riesgo
- Adquisiciones

2.2.2. PRINCE (Projects IN Controlled environments)

PRINCE2 es un método de Gestión de Proyectos basado en procesos. Es un estándar usado por el gobierno del Reino Unido y es ampliamente reconocido y usado en el sector privado tanto en el Reino Unido como internacionalmente [16]. Este método es de dominio público y ofrece una guía no propietaria de buenas prácticas para la administración de proyectos.

Los procesos que conforman el modelo del método PRINCE2 [16] son: Dirección de un proyecto, Puesta en marcha de un proyecto, Iniciar un proyecto, Control de una fase, Gestión de los Límites de fase, Gestión de la entrega de productos, Cerrar un proyecto:

3. Metodología

3.1. Definición del proceso de negocio actual

A continuación se presenta el modelo de procesos de alto nivel para el ciclo de vida productivo de las apli-

caciones según la Metodología Corporativa actual del Banco, aplicando BPMN².

Este macro proceso está constituido por seis subprocesos que representan cada etapa del ciclo de vida productivo de las aplicaciones en el Banco. A continuación desplegamos cada uno de estos subprocesos con sus respectivas descripciones.

Subproceso: Análisis Conceptual

En este subproceso se establecen todas las actividades relacionadas al análisis conceptual de las solicitudes de las distintas áreas usuarias a través del Líder Usuario. El proceso inicia cuando el Líder usuario elabora una solicitud de atención al equipo de sistemas. Luego de una serie de actividades representadas en la figura que a continuación mostramos, el proceso finaliza cuando se otorgan plazos y fechas de atención estimados para la atención de la solicitud.

Subproceso: Análisis Funcional y técnico

En este subproceso se establecen todas las actividades relacionadas al Análisis Funcional y Técnico en el ciclo productivo de las aplicaciones. El proceso inicia cuando el analista BP organiza y lleva a cabo reuniones de trabajo para delimitar los alcances funcionales de la solicitud efectuada por el Líder usuario. Mediante actividades subsecuentes, se llegan a determinar los requerimientos funcionales y técnicos de la solicitud, se cierran los artefactos relacionados a la etapa. El proceso finaliza cuando se otorgan los plazos y fechas para la construcción y pruebas unitarias.

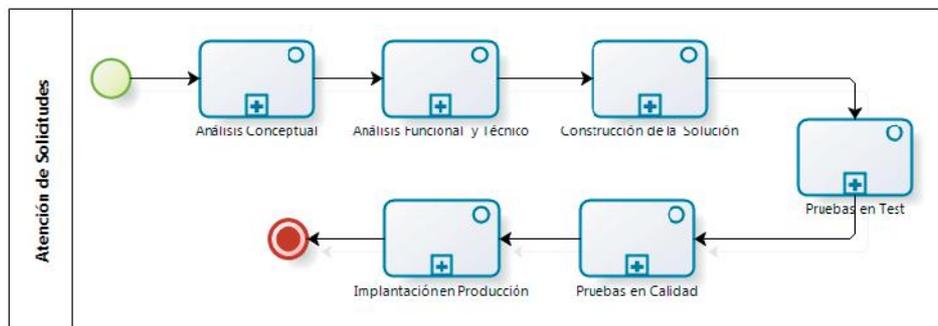


Figura 1. Modelo As-is del Proceso del ciclo productivo de las aplicaciones.

2 BPMN: Business Process Management Notation.

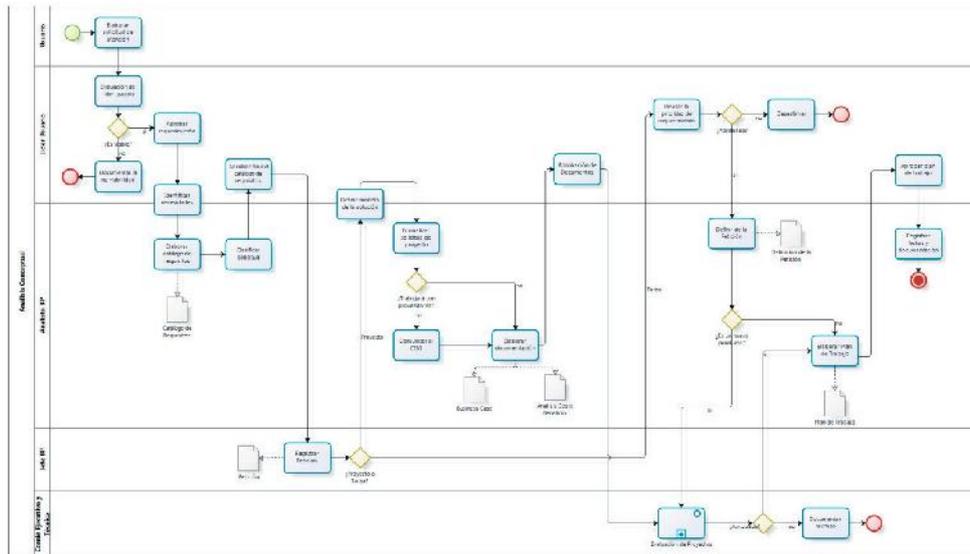


Figura 2. Modelo As-is del Sub-Proceso de Análisis Conceptual.

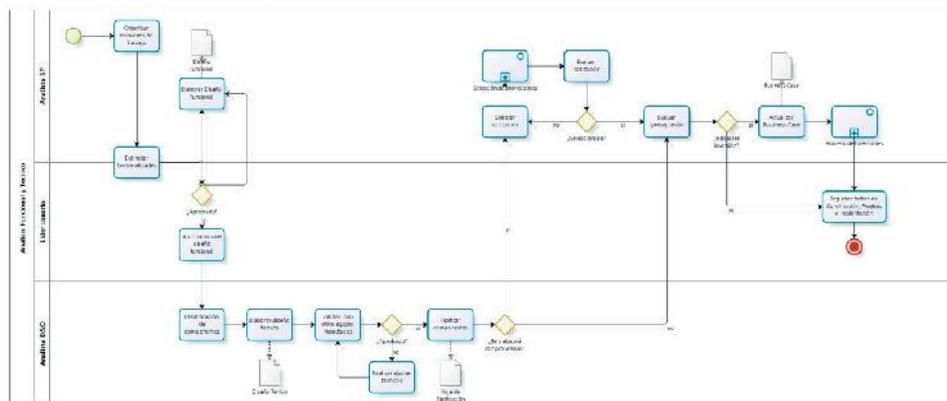


Figura 3. Modelo As-is del Sub-Proceso de Análisis Funcional y Técnico

Subproceso: Construcción

En este subproceso se realiza la construcción y pruebas unitarias de las aplicaciones nuevas o modificadas dados los requerimientos solicitados por los líderes usuarios. El subproceso inicia cuando el analista BP solicita las fechas de atención para la construcción y pruebas unitarias al analista D&D. Si la construcción se ha de realizar haciendo uso de fábrica de software, se solicita estimación de plazos de atención a la fábrica. Se realiza la construcción de los componentes solicitados, se ejecutan y evidencian los casos de prueba solicitados. El subproceso finaliza cuando se entrega la documentación de casos de prueba. Adicionalmente se realizan actividades relacionadas a esta fase, como la generación de manuales de usuario, gestionar facultades,

des, se prepara ambientes para capacitación, pruebas, guías de usuario, casos de prueba en el siguiente entorno de certificación, etc.

Subproceso: Pruebasentest

En el subproceso de pruebas en test se realiza la verificación y validación en el ambiente de Test de los componentes desarrollados. El proceso inicia con el pase de componentes hacia el entorno de test. Se realizan las pruebas, se identifican incidencias encontradas para que sean corregidas. El proceso finaliza con el pase al entorno de calidad de los componentes y la entrega al líder de implantación, de la carpeta física de documentación y firmas asociadas a la petición.

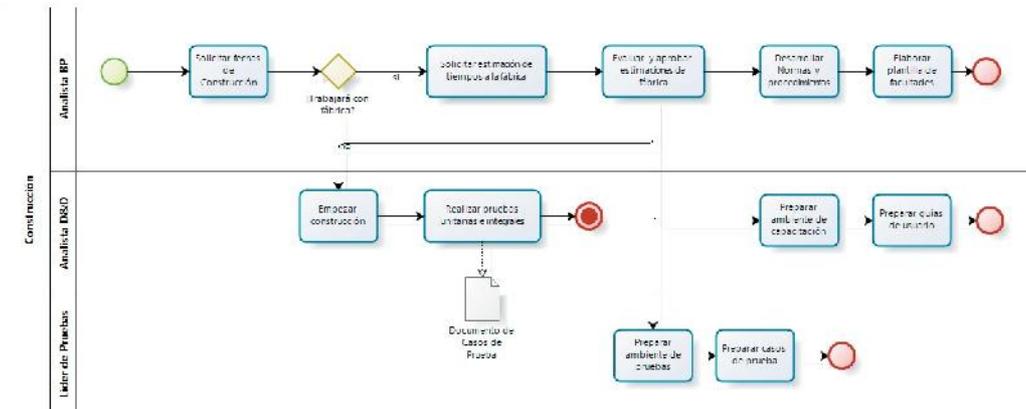


Figura 4. Modelo As-is del Sub-Proceso de Construcción.

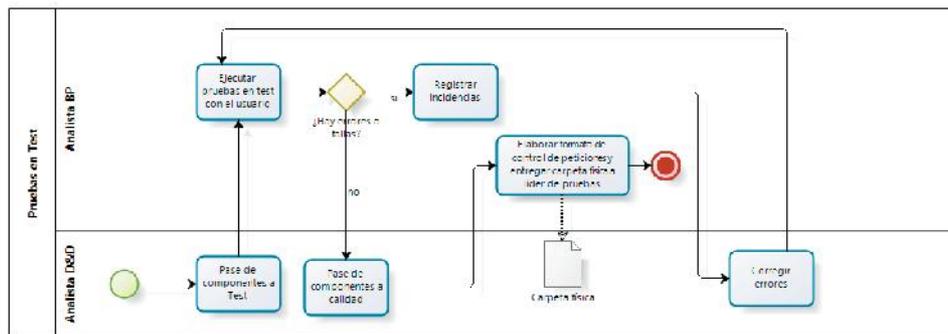


Figura 5. Modelo As-is del Sub-Proceso de Pruebas en Test.

Subproceso: Pruebas en Calidad

En el subproceso de pruebas en calidad se realizan las pruebas de los componentes en el entorno de calidad. Este subproceso es conducido por el líder de pruebas, ejecutado por el analista de pruebas y apoyado por el

analista D&D. El subproceso inicia con la ejecución de pruebas en el entorno de calidad. Los resultados de estas pruebas son validados por el líder usuario, quien da su aceptación. El proceso finaliza con la coordinación del pase a producción de los componentes asociados.

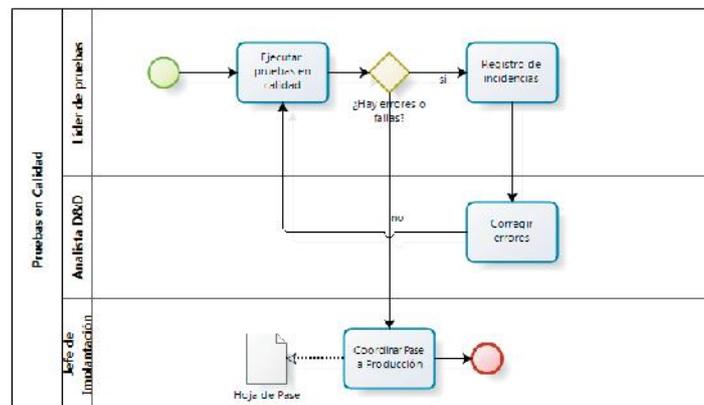


Figura 6. Modelo As-is del Sub-Proceso de Pruebas en Calidad.

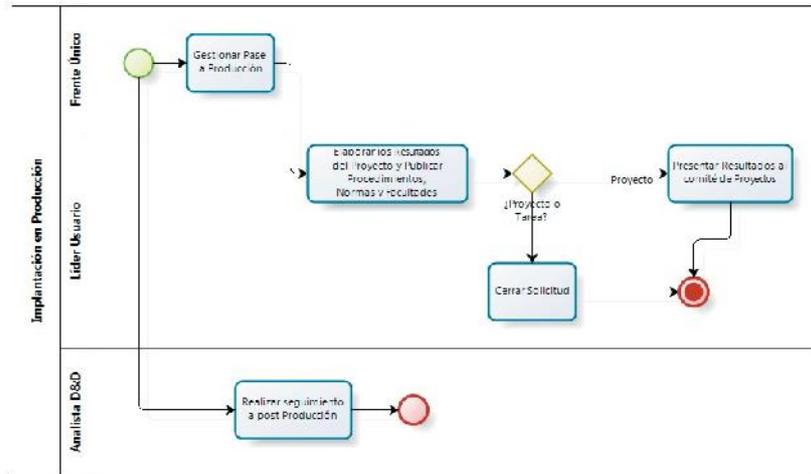


Figura 7. Modelo As-is del Sub-Proceso de Implantación en Producción.

Subproceso: Implantación en Producción

En el subproceso de Implantación en producción se realizan las actividades relacionadas con la coordinación del pase a Producción de los componentes relacionados a la solicitud inicial. El resultado de este subproceso es monitoreado por el analista D&D, y validado por el Líder Usuario.

construcción en el ciclo productivo de las aplicaciones ya sean nuevas o modificaciones de las ya existentes.

Mediante un proceso de evaluación y bajo ciertos criterios definidos por el área de Metodología y la Gerencia de la Unidad, se han seleccionado un conjunto de Fábricas de Software que prestarán el servicio de desarrollo y pruebas unitarias para la atención del ciclo productivo de las aplicaciones. En total y hasta la fecha son cinco las fábricas seleccionadas, las mismas que se encuentran distribuidas por Centro y Sudamérica, es decir nos encontramos bajo el esquema de Outsourcing Offshore³.

3.2. Propuesta del proceso de negocio mejorado

Actualmente, el Banco ha incorporado dentro de su metodología, el uso de Fábricas de Software para la atención de requerimientos de TI, durante la etapa de

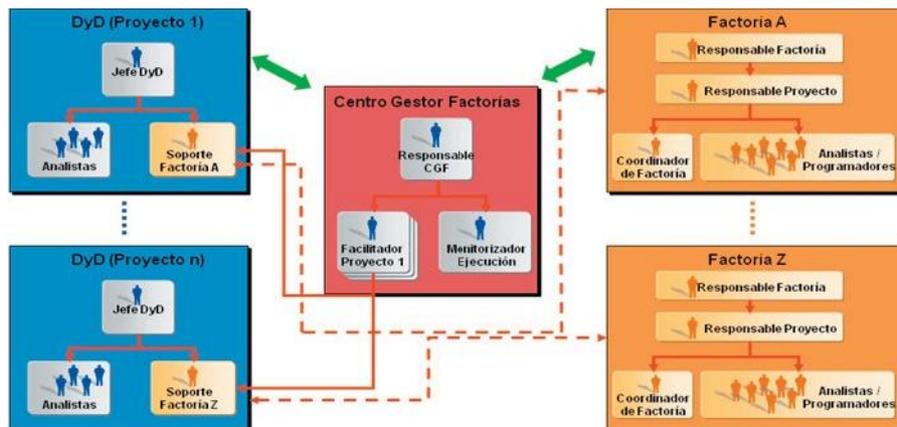


Figura 8. Esquema del modelo de fábrica de software adoptado por PERUBANK.

3 Este modelo de Outsourcing plantea la utilización de equipos de trabajo ubicados geográficamente en otro país al de la empresa que subcontrata estos servicios.

Para ello, se ha planteado el modelo que se muestra en la Figura 8, en el que se describen las relaciones y entidades que intervienen en esta etapa dentro del ciclo productivo: Cada equipo de Diseño y Desarrollo (en adelante D&D), conformado por el jefe D&D, los analistas técnicos y un encargado de soporte a las factorías, interactúa con las fábricas de software a través del Centro Gestor de Factorías (en adelante CGF) y a su vez el CGF interactúa con las distintas Fábricas de Software disponibles con las que se ha realizado la contratación del servicio.

Lo que se pretende con la utilización de este modelo de fábricas de software es disminuir la carga de trabajo de la unidad de Gestión de la Demanda, en el proceso de atención al ciclo de vida de las aplicaciones, para dedicar mayor atención al análisis y generación de iniciativas en beneficio de la organización.

Modelo de Mejoras de Procesos

Para la mejora del proceso de software en el *Banco* hemos tomado como referencia el modelo CMMI para desarrollo. De este modelo se han elegido tres áreas

tabla 1. Mapeo de Áreas de Proceso seleccionadas del modelo CMMI, según niveles de capacidad (situación actual y proyectada)

	Nivel de Capacidad Actual	Nivel de Capacidad Proyectado
Gestión de Acuerdos con Proveedores (SAM)		
SG 1: establecer los acuerdos con los proveedores		
SP 1.1: Determinar el tipo de compra	0	0
SP 1.2: Seleccionar los proveedores	1	2
SP 1.3: Establecer acuerdos con el proveedor	1	2
SG 2: Satisfacer los acuerdos con el proveedor		
SP 2.1: Realizar el acuerdo del proveedor	1	2
SP 2.2: Monitorizar los procesos seleccionados del proveedor	1	2
SP 2.3: Evaluar los productos de trabajo seleccionados del proveedor	1	2
SP 2.4: Aceptar los Productos adquiridos	1	2
SP 2.5: Transferir los productos	1	2
Verificación (VER)		
SG 1: Preparación de la verificación: La preparación de la verificación es conducida		
SP 1.1: Seleccionar los artefactos para verificación	1	2
SP 1.2: Establecer el ambiente de verificación	1	2
SP 1.3: Establecer procedimientos y criterios de verificación	1	2
SG 2: Revisar la revisión de pares		
SP 2.1: Preparar la revisión de pares	0	2
SP 2.2: Conducir la revisión de pares	0	2
SP 2.3: Analizar los datos de la revisión de pares	0	2
SG 3: Verificar artefactos selecciones		
SP 3.1: Realizar la verificación	1	2
SP 3.2: Analizar resultados de la verificación identificando acciones correctivas	1	2
Validación (VAL)		
SG 1: Preparar la validación		
SP 1.1: Seleccionar productos para la validación	1	2
SP 1.2: Establecer el ambiente para la validación	1	2
SP 1.3: Establecer procedimientos y criterios de validación	1	2
SG 2: Validar productos o componentes del producto		
SP 2.1: Realizar la validación	1	2
SP 2.2: Analizar los resultados de la validación	1	2

de procesos para su representación continua, las mismas que se encuentran directamente involucradas en la tercerización de la construcción y pruebas unitarias a través de fábricas de software.

En la Tabla 1 mostramos un mapeo entre las áreas de proceso seleccionadas del modelo CMMI cada una de ellas con sus respectivas metas específicas y prácticas específicas, y su nivel de capacidad actual, de acuerdo al modelo adoptado por el banco basado en la Metodología Corporativa, para identificar los puntos débiles que deberán ser tomados en el análisis de nuestra aplicación.

Según las áreas de proceso seleccionadas del modelo CMMI – DEV 1.2 se ha definido el proceso de Atención de Envíos a Fábrica de Software, aplicando algunas mejoras según el modelo. A continuación describimos las particularidades del proceso:

Nombre del Proceso: Atención de envíos a fábrica de software

Meta del Proceso: Agilizar y mantener controlado el proceso de envío de componentes para su construcción y pruebas unitarias a las fábricas de software.

descripción del Proceso

ACTIVIDAD	PROCEDIMIENTO
Abrir solicitud de atención	El analista D&D realiza la apertura de una solicitud de atención para fábrica de software, en la que registra un formato de planificación para notificar el volumen de componentes que serán enviados.
Seleccionar Fábrica	El CGF selecciona la fábrica que podría atender la solicitud de acuerdo a su disponibilidad. Esta actividad está basada en el modelo CMMI, Área de proceso: Gestión de Acuerdos con Proveedores; Meta específica: Establecer Acuerdos con Proveedores (SG1); Práctica Específica: Seleccionar los proveedores (SP 1.2).
Asignar Analista	El líder de fábrica asignará al analista que va atender la solicitud, registrando sus datos básicos para coordinaciones posteriores si se requiriera.
Registrar Documentación Técnica	El analista D&D registra el formato de tipificación de componentes para ser categorizados según su complejidad, lo que ayudará posteriormente a contrastar la facturación del servicio y estimar el tiempo de atención. Adicionalmente registra la documentación técnica que aplica para el tipo de componentes a ser desarrollados. Los componentes enviados a fábrica de software pueden ser de plataforma HOST o Distribuida.
Verificar Documentación Técnica	El CGF verifica que la documentación ingresada esté completa y sea consistente. Además verifica que la tipificación sea correcta. Se da visto bueno y se traslada a la fábrica asignada.
Revisar Documentación Técnica	El líder de fábrica valida la documentación técnica proporcionada por el analista D&D e identifica las dudas en cuanto a los requerimientos técnicos que han sido solicitados.
Registrar Dudas	Si el líder de pruebas considera que existen puntos ambiguos o que falten especificar, deberá registrar las dudas para que estas sean resueltas por el analista D&D.
Registrar Respuesta	El analista D&D deberá registrar la respuesta a las dudas de la fábrica.
Construcción	El desarrollador asignado realiza la construcción de los componentes solicitados.
Registrar Entregables	Una vez finalizada la construcción y pruebas unitarias, el Líder de fábrica registra la documentación de pruebas unitarias y las evidencias correspondientes, así como el estatus de los componentes desarrollados.
Verificar Entrega	El CGF verifica que la documentación entregada este completa, es decir que haya evidencia de todo lo solicitado
Validar Entrega	El analista D&D valida los componentes entregados y compara con los casos de prueba entregados.
Registrar Error	De encontrar algún error en la entrega, el analista D&D registra el error para que sea atendido y resuelto por el desarrollador de la fábrica correspondiente.
Corregir Error	El desarrollador de la fábrica corrige el error notificado por el analista D&D.
Validar Corrección	El analista D&D valida que el error haya sido corregido
Cerrar Solicitud de Atención	El analista D&D cierra la solicitud dando visto bueno a al circuito de atención

tabla 2. Actividades del proceso de Atención de envíos a Fábrica de Software.

Alcance del proceso:

Inicio del Proceso: El proceso inicia cuando el analista D&D registra la solicitud de Atención de Fábrica, llenando los campos informativos solicitados para ser usados durante el proceso de atención.

Fin del Proceso: El proceso finaliza con el cierre de la solicitud de atención, luego del visto bueno por parte del analista D&D.

Roles:

Dueño del Proceso: CGF (Centro Gestor de Factorías)

Equipo del proceso: Analista D&D, CGF, Líder de Fábrica, Desarrollador

Clientes del Proceso: Analista D&D

Proveedores del Proceso: Analista D&D

Indicadores claves de Rendimiento:

1. Tiempo promedio de resolución de Dudas
2. Tiempo promedio de resolución de Errores
3. Tiempo promedio de construcción de componentes según complejidad

En la Figura 9 se muestra el modelo del proceso representado mediante la notación BPMN.

4. ReQUeRIMieNtoSy dISeño de LAAPLICACIóN

4.1 Requerimientos

Los requerimientos funcionales para el modelamiento de la herramienta web para el seguimiento integral del ciclo productivo de aplicaciones para PERUBANK serían los siguientes:

[ReQ1] Gestionar Cartera de Solicitudes

1. Registrar, actualizar y cancelar solicitudes, ya sean iniciativas, propuestas, proyectos o tareas que la organización requiere, y la documentación correspondiente a la solicitud.
2. Categorizar solicitudes según criterios definidos en las reglas del negocio.
3. Priorizar solicitud según sea requerido por la unidad solicitante para su atención.
4. Estimar o desestimar solicitudes, registrando justificación y anexos relevantes.
5. La información manejada a nivel de cartera de solicitudes deberá ser retroalimentada por los módulos que desarrollen la solicitud, para mantener el estado de la solicitud actualizado.
6. Consultar información de las solicitudes ingresadas.
7. Establecer línea base de atención a las unidades usuarias, según priorizaciones.

[ReQ2] Gestionar Proyectos

1. Registrar datos generales, formatos y documentación relevante para el proceso de gestión de proyectos.
2. La planificación de los proyectos deberá manejarse a través de una plantilla en formato MS Project 2007 y actualizada según el avance del proyecto.
3. Seleccionar ciclo de vida del producto, según se requiera.

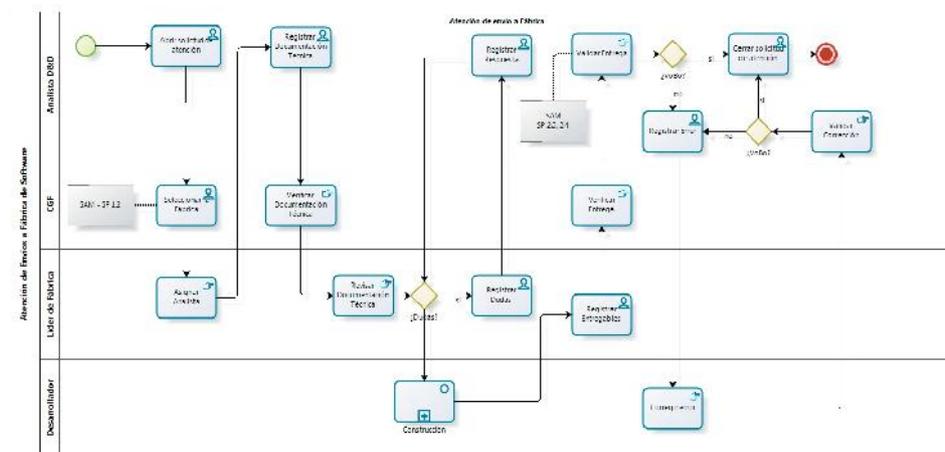


Figura 9. Modelo To-be del Proceso de Atención de Envíos a Fábrica.

[ReQ3] Gestionar Peticiones

1. Esquematizar las etapas del ciclo de vida del producto.
2. Registro de datos generales de la petición, como clasificación, equipo responsable, objetivos, etc.
3. Registro de fechas planificadas de inicio – fin de cada etapa.
4. Registro de fechas reales de inicio – fin de cada etapa.
5. Consulta y seguimiento de peticiones.

[ReQ4] Gestionar Fábricas de Software

1. Registrar los paquetes de componentes enviados a fábrica de software, datos generales y documentación relacionada, junto con el formato de solicitud.
2. Administrar fábricas de software que dan servicios de desarrollo y pruebas unitarias *al banco*.
3. Definir tarifas según tipología de componentes y complejidad de los mismos.
4. Registrar escala de penalización por incidentes registrados por parte de fábrica.
5. Generar reportes de uso de fábricas según los criterios definidos por el usuario.

[ReQ5] Gestionar documentación

1. Implementar un esquema de permisos y flujo de aprobaciones para automatizar el manejo de carpetas físicas de la documentación de las peticiones.
2. Deberá centralizarse el manejo de documentación concerniente al proyecto, aplicando control de versiones para mantener la información actualizada.
3. Deberá manejarse un catálogo de formatos aplicables según la tipología del requerimiento y controlar la carga de aquellos que sean restrictivos según los controles SOX definidos por Auditoría de Sistemas.

[ReQ6] Gestionar Seguridad

1. Deberá controlarse el acceso a la aplicación a través de autenticación de los usuarios y password del Sistema Central.
2. Deberá manejarse un esquema de perfiles para autorizar acciones sobre el sistema según se requiera.
3. Si un usuario de la aplicación que tiene rol de jefe se ausenta por vacaciones o algún otro motivo, de-

berá considerarse la posibilidad de derivar dicho rol a algún colaborador de su equipo temporalmente.

Y para los requerimientos no funcionales serían los siguientes:

1. Respaldo periódicamente de la documentación almacenada en el servidor de archivos para mitigar la pérdida de información por posibles eventualidades.
2. Considerar la independencia del browser para la visualización de los formularios de la aplicación, ya que serán accedidos a través de intranet y extranet.
3. Se utilizará estándares para la generación de nombres de carpetas en el servidor de archivos, los versionados de la documentación.
4. Las carpetas y archivos correspondientes a los proyectos, solicitudes, peticiones, etc., serán manejadas en el sistema de archivos según una organización jerárquica.
5. Notificación de eventos relevantes en la atención de proyectos o tareas a través de correo electrónico automático a los involucrados.

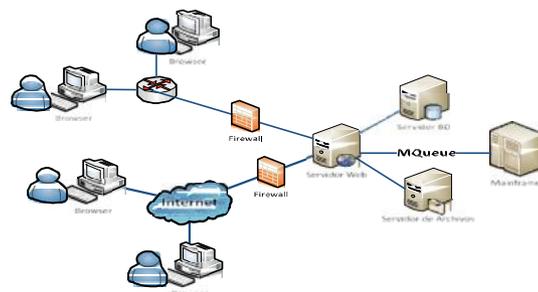
4.2 determinación del entorno tecnológico

Figura 10. Entorno Tecnológico de la Aplicación Web Propuesta

4.3 Especificación de estándares y normas

[bR1] El líder usuario registrará la solicitud, de acuerdo a una necesidad de su unidad, previamente revisada en cuanto a viabilidad.

[bR2] A partir de haber registrado la solicitud, el Jefe BP tiene cinco días útiles para revisar, evaluar y categorizar la solicitud, que puede ser proyecto o tarea. De ser un nuevo producto, deberá ser incluido en alguna de estas dos categorías.

[bR3] La priorización de la solicitud será coordinada entre el usuario y el Jefe BP, de acuerdo a un análisis

macro que haga el Jefe BP y la necesidad del Líder Usuario.

[bR4] Cualquier nueva solicitud, será registrada y se le asignará la prioridad correspondiente según sea convenido por ambas partes.

[bR5] El Jefe BP será el responsable de registrar la documentación referente a la gestión de los proyectos.

[bR6] La planificación será definida mediante un archivo de MS Project por el Analista BP, y estará sujeto al control de versiones a implementarse en la herramienta.

[bR7] Una solicitud será clasificada como proyecto cuando esta consuma más de 500 horas/hombre de esfuerzo o 10,000 US\$ en presupuesto.

[BR8] Todo proyecto deberá ser evaluado por un comité de proyectos para revisión de su viabilidad, y por un comité técnico que evaluará su implicancia técnica.

[BR9] Una solicitud será clasificada como tarea cuando esta consuma menos de 500 horas/hombre de esfuerzo o 10,000 US\$ de presupuesto.

[BR10] Existe la posibilidad de definir peticiones madres e hijas, cuando se requiera, cuando todas las pe-

ticiones hijas se hayan cerrado, se procederá a cerrar la petición madre.

[bR11] Se establece que el plazo de respuesta para proporcionar fechas de Diseño Funcional, por parte del analista BP es de 3 a 7 días útiles dependiendo de la complejidad de la petición.

[bR12] Se establece que el plazo de respuesta para proporcionar fechas de Diseño Técnico, por parte del analista D&D es de 3 a 7 días útiles, dependiendo de la complejidad de la petición.

4.4. diagramas de paquetes y subsistemas

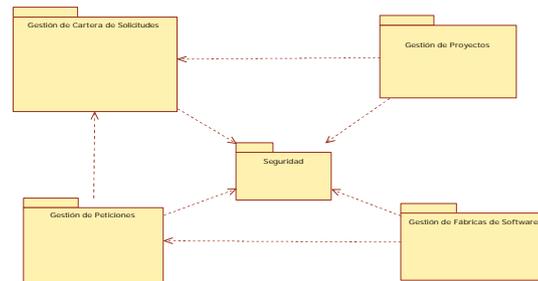


Figura 11. Diagrama de Paquetes y sus Relaciones

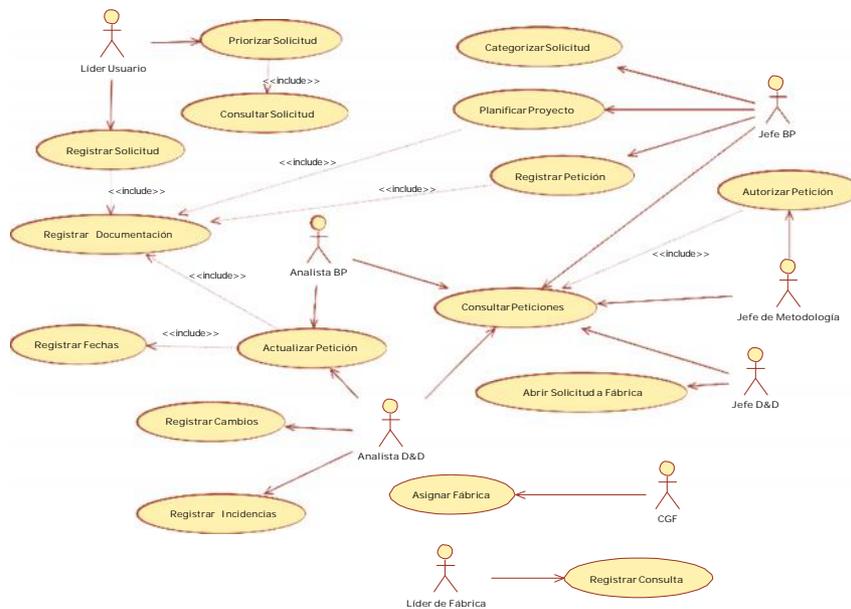


Figura 12. Diagrama de Casos de Uso Propuesto

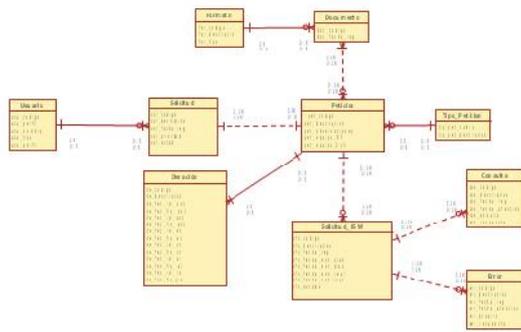


Figura 13. Diagrama E-R

Vista Física

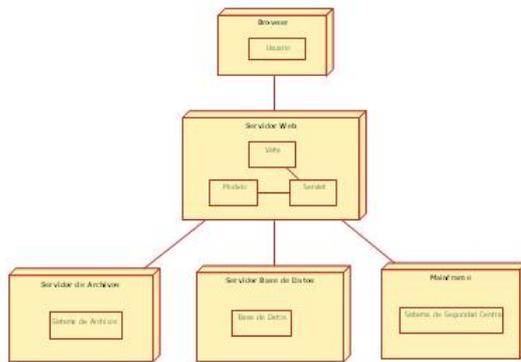


Figura 14. Diagrama de Despliegue

Vista de Componentes

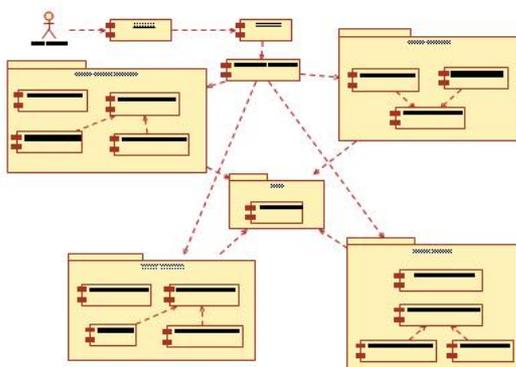


Figura 15. Diagrama de Componentes

5. CONCLUSIONES

- A partir del modelado de una herramienta de Gestión bajo la plataforma web, se ha podido integrar el Proceso del Ciclo Productivo de las Aplicaciones para la unidad de Sistemas en PERUBANK, junto con las actividades de Gestión de Proyectos, basándonos en lineamientos del modelo de mejora de procesos CMMI, el cual fue seleccionado, según criterios extraídos de un trabajo de investigación y ponderaciones según los requerimientos de la Organización.
- Se ha logrado abarcar un 40% de las actividades vinculadas al Proceso de Ciclo de Vida de las Aplicaciones. El resto de actividades seguirá siendo manual, ya que la naturaleza del proceso así lo requiere, y deberá conducirse un proceso de mejora para estandarizar e institucionalizar estas actividades manuales, según el modelo aplicado.
- Se ha logrado definir el proceso de negocio “mejorado”, del cual se basa nuestro enfoque, esto es, el proceso de Interacción con las fábricas de software, modelo adoptado recientemente por la organización.
- Para la implantación del sistema que ha sido modelado, es importante planificar un periodo de concientización de la utilización de la herramienta, así como el uso de la metodología definida por la organización.

6. REFERENCIAS BIBLIOGRAFICAS

- [1] I. Sommerville. 2005 *Ingeniería de Software*. 7.ª ed. Madrid: Pearson Education.
- [2] Bernd Bruegge, Allen H. Dutoit. 2002. *Ingeniería de Software Orientada a Objetos*. México, Pearson Education.
- [3] www.sei.cmu.edu/. Software Engineering Institute.
- [4] International Organization of Standardization. 2008. *Norma Internacional ISO:9001:2008 - Sistemas de Gestión de la Calidad - Requisitos*. 4.ª ed. Suiza.
- [5] <http://www.iso15504.es/>. Portal de la Norma ISO 15504
- [6] Comisión de Reglamentos Técnicos y Comerciales - INDECOPI 2006. “Norma Técnica Peruana – ISO/IEC 12207”. Lima.
- [7] Chrissy Mary Beth, Mike Konrad, Sandy Shrum 2009. *CMMI: Guía para la integración de procesos*

- y la mejora de productos. 3ª ed. Madrid. Addison Wesley.
- [8] Jacobson Ivar, Grady Booch, Jim Rumbaugh 1999. "Unified Software Development Process". Addison-Wesley.
- [9] Rational Software Corporation 1998. "Rational Unified Process – Best Practices for Software Development Team". EEUU.
- [10] Kruchten Philippe. *A Rational Development Process: An Introduction*. RUP, 3ª ed. pp.11-16.
- [11] Ministerio de Administraciones Públicas 2009. "Métrica versión 3 – Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información". Gobierno de España.
- [12] Beck, K. 1999 "Extreme Programming Explained". Embrace Change, Pearson Education. Traducido al español como: Una explicación de la programación extrema. Addison Wesley.
- [13] Palacio Juan 2007. "Flexibilidad con Scrum: Principios de diseño e implementación de campos con SCRUM".
- [14] Plattini Velthuis Mario, Javier Garzás 2007. *Fábrica de Software: Experiencias, Tecnologías y Organización*. México, Ramma.
- [15] PMI 2004. "Guía de los Fundamentos de la Dirección de Proyectos". Project Management Institute, Inc. EE.UU.
- [16] <http://www.prince2.com/>. Método Prince2 – Projects in Controlled Environments.
- [17] S. Pressman Roger 2001. "Software Engineering: A Practitioners Approach". 5ª ed. Mc Graw Hill series in Computer Science.

