FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
ESPECIALIZACIÓN EN ARQUITECTURA EMPRESARIAL DE SOFTWARE
Ingeniería de Software

Parámetros del Proyecto

Startup BilliTel

Objetivo

El objetivo del proyecto del curso es investigar una de las fases descritas en el desarrollo. Cada fase tiene una problemática o un “dolor” que hace que el proceso sea lento o el producto generado sea de baja calidad. Los estudiantes deben analizar dicha problemática y proponer una mejora en el proceso (pain point).

Desarrollo del proyecto

La mejora debe estar estructurada en:

- Prácticas
  - ¿Cuáles van a usar?
  - ¿Cómo van a usar esas prácticas y para qué?
- Herramientas
  - ¿Cuáles van a usar?
  - ¿cómo van a aplicar dichas herramientas
  - ¿Para qué y por qué?
- KPI's (Key Performance Indicators)
  - ¿Cómo van a medir la mejora?
  - ¿Cómo les ayudará a saber si mejoraron en efecto o no?
  - ¿Por qué escogieron dichas métricas?
- Pitch (sustentación)

Estas sugerencias deben ser sustentadas bajo la temática vista en el curso.

Restricciones

- El proyecto se puede realizar en grupos de 4 a 5 estudiantes.
- No se permite divisiones de grupo.
- Todos los grupos de proyecto deben trabajar con la empresa propuesta en la descripción del proyecto.

Contexto

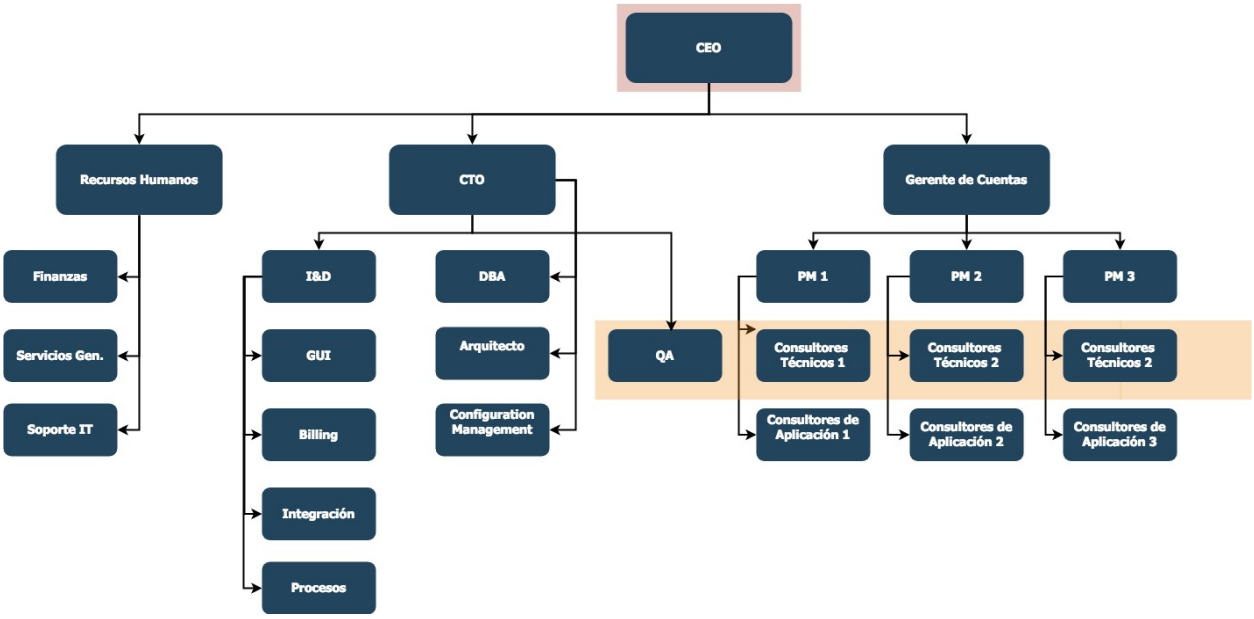
La empresa BilliTel es una Startup en proceso de consolidación cuyo producto base es un sistema de facturación y CRM para empresas de telefonía móvil. La mayoría de sus clientes son operadores virtuales MVNO (Mobile Virtual Network Operators). Las MVNOs son operadores que ofrecen servicios de telefonía celular sin tener infraestructura física, es decir ellos alquilan la infraestructura a un operador real. Usualmente las MVNOs contratan los servicios de CRM, tasación, facturación y logística a terceros. El núcleo del negocio es facturar las llamadas de los subscriptores y ofrecer servicios de activaciones y post venta a los subscriptores.

Problemática General

Debido a que la empresa a crecido de manera rápida y ha ganado varios contratos con distintos clientes ha empezado a tener problemas de calidad y deterioro en la productividad de los desarrolladores. Las entregas son cada vez más dispendiosas y en cada entrega no se le está dando valor al negocio a los clientes por la cantidad de bugs que reportan. Muchos de estos bugs son debido a problemas de configuración y de instalación. La solución inmediata es contratar más desarrolladores. Por otro lado los clientes existentes están generando una serie de requerimientos qué hacer qué el product roadmap avance de manera muy lenta. Los requerimientos levantados pueden ser similares entre los clientes pero no hay una buena comunicación o una centralización de los requerimientos. Esto conlleva a duplicidad de trabajo y duplicación de funcionalidad.

Estructura de la empresa

Organigrama



CTO

Gerente de Tecnología, encargado de dirigir la parte técnica y arquitectónica de la plataforma. El CTO tiene como responsabilidad crear y administrar el product roadmap. De éste departamento se desprende los siguientes cargos.

- I&D Investigación y desarrollo. Es el departamento de crear los componentes núcleo del sistema.
    - GUI. Encargado de administrar los componentes de interfaz gráfica
    - Billing. Encargado de administrar los componentes de facturación y tasación.
    - Integración. Encargado de administrar los componentes de integración de la plataforma con otros sistemas.
    - Procesos de negocio. Equipo encargado de hacer mejoras con respecto al sistema de flujo de trabajo de la plataforma. Este componente se comportaría como capa media.
- DBA. Administrador de base de datos.
- Arquitecto. Trabaja de la mano con el CTO para determinar el product roadmap. Crea

nuevas funcionalidades que son transversales a toda la plataforma.
- Configuration Management. Encargado de administrar el repositorio de versiones y los releases de que hay de la plataforma.
- QA. Encargado de realizar pruebas de calidad a los entregables de la plataforma. Adicionalmente ofrece servicios a las cuentas de clientes para hacer pruebas específicas para cada cliente.

Gerente de cuentas

Es el gerente encargado de manejar las relaciones comerciales con los clientes. Tiene a cargo varios Project Managers que manejan las cuentas individuales de los clientes. Cada Project Manager puede estar a cargo de uno o varios clientes dependiendo el tamaño de la operación. A su vez, tiene a cargo los siguientes roles.

- Consultores Técnicos: Son desarrolladores que implementan nuevos requerimientos. Pueden tener varios roles como tester, puesta en producción, corrección de bugs, configuración del sistema. Deben crear requerimientos en las cuatro áreas de la plataforma (GUI, procesos de negocio, billing e integración). Un equipo de consultores técnicos puede estar entre tres a siete desarrolladores.
- Consultores de aplicación. Son los encargados de levantar requerimientos, administrar y configurar la plataforma. Su perfil no es muy técnico. Un equipo de consultores de aplicación puede conformarse entre uno a tres personas.

Ambiente tecnológico

Lenguajes de programación.

- Java. Toda la plataforma está implementado en Java.
- JavaScript. Utilizado para el front end de la plataforma

Base de datos. La base de datos utilizada es Oracle.

Herramientas de construcción de software.

- Ant. Herramienta para automatizar el compilado y el empaquetado de software. Sin embargo hay problemas de versionamiento y centralizado de librerías de terceros. El manejo de releases se hace más complicado.
- Subversion. Control de versiones centralizado.
- Eclipse. IDE para implementar proyectos en Java. Los desarrolladores pueden usar cualquier IDE sin embargo los proyectos actuales están atados a Eclipse debido a que la herramienta genera la compilación y el empaquetado.

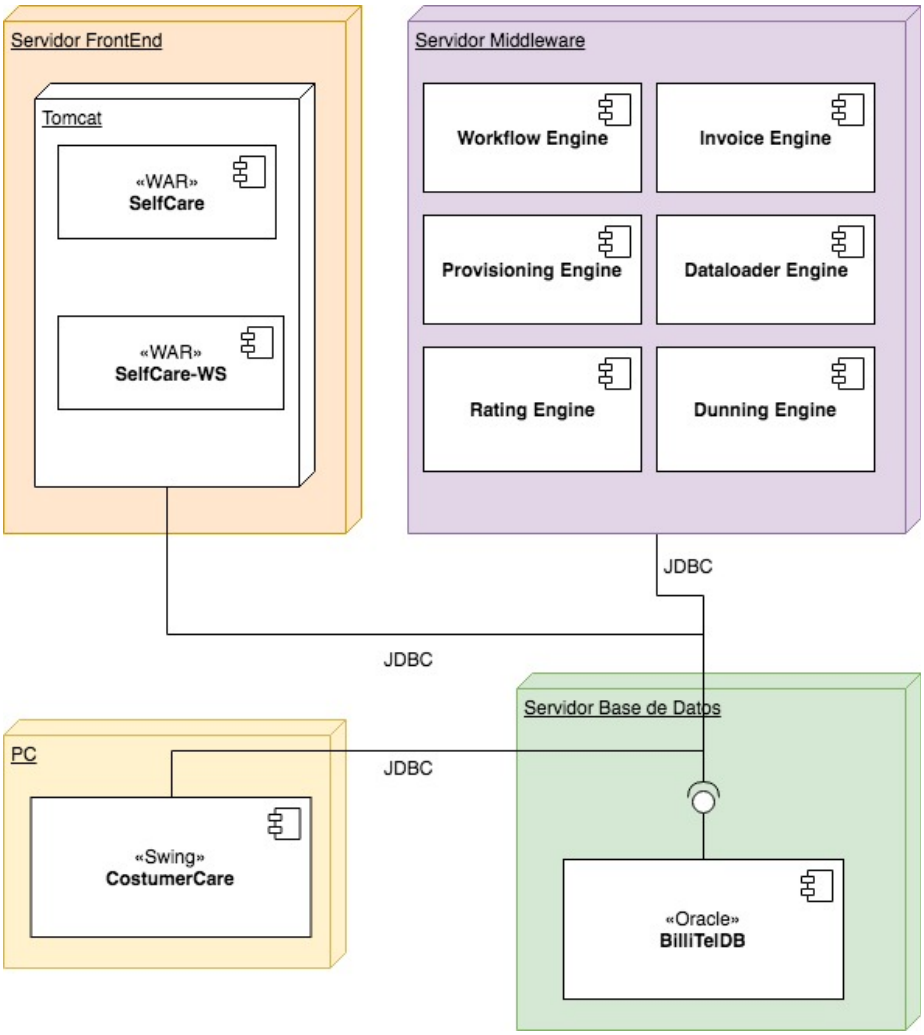
Herramientas de gestión de proyectos

- Jira. Es una herramienta para registrar requerimientos y rastreo de bugs. Tiene funcionalidades de métodos ágiles.

- Confluence. Herramienta de documentación similar a un wiki.

Arquitectura general

El siguiente diagrama representa el diagrama de despliegue general que utiliza la compañía en una instalación típica.



- **Servidor FrondEnd**. Este servidor ofrece los servicios de acceso a la aplicación por web. Los componentes corren dentro de un servidor Tomcat.
    - **SelfCare.** Este componente ofrece todos los servicios básicos para que un subscriptor administre su cuenta.
    - **BilliTel-WS.** Este componente ofrece una API para que otros proveedores puedan integrarse a la plataforma. Utiliza SOAP WebServices para ello.
- **Servidor Middleware.** Este servidor contiene todos los componentes que corren en la capa media y realizan las tareas core del negocio. Los componentes se apellidan

"Engine" debido a que son procesos Java que son administrados por el sistema operativo, son autoejecutables y no dependen de un servidor de aplicaciones. Por consiguiente la lógica no-funcional debe ser implementada por ellos mismos.

- **Workflow Engine**. Este componente como su nombre lo dice maneja el flujo de trabajo de los procesos de negocio. Estos procesos pueden ser de larga duración.
- **Provisioning Engine**. Este componente se encarga de comunicarse con servidores externos para aprovisionar elementos en la red de telecomunicaciones. Por ejemplo activa subscriptores en la red.
- **Rating Engine**. Este componente se encarga de la tasación de las llamadas hechas por los subscriptores.
- **Invoice Engine**. Este componente se encarga de ejecutar los ciclos de facturación generando así la factura para cada uno de los subscriptores.
- **Dataloader Engine**. Este componente tiene como responsabilidad de subir datos de archivos externos a la base de datos.
- **Dunning Engine**. Este componente se encarga de realizar el proceso de recaudo de las facturas. Si un suscriptor no ha pagado éste componente inicia el proceso de cobro.

Proceso de Desarrollo

Administración de Requerimientos

Los requerimientos se originan de cuatro fuentes principales.

1. **Funcionalidades específicas de los clientes**. Estos son el 70% de los requerimientos levantados por la compañía. Los requerimientos son elicitados en conjunto por los clientes, Consultores de Aplicación y Project Managers. No hay mucha comunicación entre las cuentas de clientes ni el área de I&D para saber si es posible reutilizar funcionalidades ya implementadas.
2. **Restricciones legales de los gobiernos donde el cliente tiene operaciones**. Por ejemplo, reglas de funcionamiento de portabilidad numérica, reglas fiscales o restricciones a los operadores. Estos requerimientos tienen alta prioridad ya que el costo lo asume la empresa y puede generar sanciones a los clientes por la no implementación. No son muy frecuentes.
3. **Internos**. Son requerimientos funcionales o no-funcionales que solicita el Product Owner que se incluya en la plataforma. Estos requerimientos son para el mejoramiento de la plataforma y darle nuevas funcionalidades a la misma. Son atendidos por el equipo de I&D. Usualmente estos requerimientos tardan en ser desarrollados y requieren un tratamiento especial ya que afecta el comportamiento de la plataforma afectando a todos los clientes si hay un bug. Por consiguiente deben pasar por QA para ser aprobados.

4. **Necesidades del mercado.** Si varios clientes empiezan a solicitar las mismas necesidades se convierte en una necesidad del mercado. Si el Product Owner detecta que hay una necesidad de mercado, especifica el requerimiento y lo coloca en el product backlog. Usualmente estos requerimientos son implementados por el equipo de I&D. Desafortunadamente estos requerimientos tienen baja prioridad, pero la no implementación de estos genera mucha duplicidad.

Todos los requerimientos son documentados en Jira.

Debido a que las cuentas de los clientes tienen la mayoría de los requerimientos este nos enfocaremos en describir su proceso.

Análisis y Diseño

Una vez levantado el requerimiento y priorizado es asignado a un Consultor Técnico. El desarrollador tiene la responsabilidad de analizar, re estimar, diseñar e implementar el requerimiento. El proceso de análisis se hace con base en la experiencia de desarrolladores senior y así también es su estimación. No hay métricas, ni técnicas bien definidas para la estimación de requerimientos. Han habido muchas quejas por parte del cliente con respecto a este punto. Muchos requerimientos son subestimados.

Desarrollo

El desarrollo de la plataforma tiene varios desafíos. Quizás el mayor problema que tiene la plataforma fue la velocidad de creación de nuevas características. El producto empezó de la nada y creció demasiado rápido. Como consecuencia no hubo un diseño flexible para la reutilización de funcionalidades. Otro desafío son las malas prácticas en el manejo de Subversion. No hay un claro uso de esta herramienta y existen actualmente muchos problemas de colisiones entre desarrolladores.

Entrega (Delivery)

Uno de los mayores desafíos para la empresa es la entrega e instalación de los componentes en el cliente. Cada cliente puede tener una versión de la plataforma. Cada versión puede tener diferentes dependencias de librerías de terceros.

El compilado de los entregables se hacer algunas veces en las máquinas de los desarrolladores y no hay versiones claras de la entrega. Para el cliente y para los consultores en general es muy difícil determinar cuál versión está instalada en cada cliente. Incluso se ha divergido la versión de la plataforma en una rama única para algunos clientes, complicando así la incorporación de nuevas funcionalidades.

Manejo de fallos (Bugs) de alta y baja prioridad

Los bugs que son detectados por los clientes pueden ser de alta, media o baja prioridad. Los

bugs siempre afectan la planeación del ciclo de desarrollo. Los bugs de prioridad alta deben ser resueltos incluso antes de la entrega oficial del ciclo. Esto se llaman hot fixes. El problema es que al no haber una clara metodología de la entrega y manejo de “configuration management” hace difícil este tipo de correcciones.

Los bugs de prioridad media o baja deben ser priorizados por el Project Manager para que sean entregados dentro del ciclo de desarrollo. Esto afecta la planeación de nuevos requerimientos.

Propuestas de Mejora

El equipo puede hacer propuestas de mejora en una fase en particular. Sin embargo para mejorar dicha problemática puede hacer propuestas más profundas, como por ejemplo cambio en la arquitectura o en la estructura organizacional.