

PROYECTO DE GRADO

Marco de Trabajo para el desarrollo de Software

Dousdebes Abraham, José Gabriel

Ingeniería en Informática
Facultad de ingeniería
Universidad Católica de Salta

2017

Marco de Trabajo para el desarrollo de Software

Firmas

Profesor Guía:

Perdiguero, Jorge Alejandro

Profesor Tribunal Evaluador:

Profesor Tribunal Evaluador:

Profesor Tribunal Evaluador:

Fecha de exposición:

Agradecimientos

A mi familia.

Especialmente a mi mamá, Gloria, gracias por tu amor incondicional e infinito. Tu sonrisa es la fuerza que me empuja cada día.

Capítulos

Introducción.....	11
Objetivos.....	12
Marco teórico.....	13
Método de trabajo a utilizar.....	26
Personas y procesos.....	28
Artefactos.....	35
Elementos.	37
Fases.	38
Reuniones.	67
Verificación experimental.	75
Análisis.	113
Conclusiones.....	117
Anexo.	118
Bibliografía.....	119

Índice de contenidos

Marco de Trabajo para el desarrollo de Software.....	11
Introducción.....	11
Objetivos.....	12
Alcance	12
Brief	12
Marco teórico.....	13
Scrum.....	13
Roles	14
Artefactos.....	14
Reuniones	15
Kanban.....	16
Elementos	17
Manifiesto ágil.....	18
Valores.....	18
Principios	18
Cascada.....	19
Etapas	19
Espiral.....	20
Fases	21
UML	22
Proceso unificado	23
Elementos	23
Características.....	24
Fases	24
Resumen	24
Ventajas y desventajas.....	24
Método de trabajo a utilizar.....	26
Roles	26
Investigador	26
Consultor	26
Reuniones	26

Reunión de actualización.....	26
Artefactos.....	27
Ítem.....	27
Lista de ítems.....	27
Solución propuesta	28
Personas y procesos.....	28
Roles.....	28
Organización.....	33
Cultura organizacional general.....	33
Locación del Equipo.....	34
Artefactos.....	35
Lista del Proyecto	35
Lista de ítems.....	35
Ítem.....	35
Tareas.....	36
Error.....	36
Tablero.....	36
Elementos	37
Sistema de gestión de proyectos.....	37
Pizarras	37
Útiles.....	37
Sala de reuniones	37
Comida	37
Fases	38
Definición	41
Diseño.....	47
Desarrollo	58
Implementación	64
Reuniones	67
Definición	67
Planificación	70
Diaria	70

Demostración.....	71
Retrospectiva	72
Semana de innovación	74
Verificación experimental	75
Situación anterior.....	75
Roles, personas y organización	77
Desarrollador	77
Arquitecto	77
Líder del Proyecto	78
Encargado del Producto.....	78
Cliente.....	79
Personas	79
Artefactos y Elementos.....	81
Ítems y tareas	82
Lista del proyecto	85
Lista de ítems.....	86
Iteraciones.....	88
Definición	88
Diseño	96
Desarrollo	99
Implementación	104
Reuniones	105
Definición	105
Planificación	105
Diaria	105
Demostración.....	106
Retrospectiva	106
Análisis de rendimiento del Marco de trabajo.....	107
Tiempo de desarrollo + Soporte	108
Tiempos de desarrollo y bugs	109
Puntos (Valor de dificultad) por iteración	110
Puntos estimados y completados por iteración.....	111

Estadísticos adicionales	112
Análisis	113
Análisis económico	113
Recursos humanos	113
Recursos de Hardware e Insumos.....	113
Recursos de Software	113
Recursos de Infraestructura	114
Resumen	114
Análisis de riesgo	115
Plan de contingencia.....	116
Conclusiones.....	117
Futuras líneas de investigación.....	117
Anexo	118
Glosario	118
Licencia	118
Bibliografía.....	119

Índice de Tablas

Tabla 1 - Recursos Humanos.....	113
Tabla 2 - Recursos de Hardware e Insumos	113
Tabla 3 - Recursos de Software.....	114
Tabla 4 - Recursos de Infraestructura.....	114
Tabla 5 - Resumen de Recursos	114
Tabla 6 - Tabla de riesgos.....	115
Tabla 7 - Plan de contingencia	116

Índice de Ilustraciones

Ilustración 1 - Representación gráfica del proceso de Scrum	13
Ilustración 2 - Tablero de Kanban	16
Ilustración 3 - Flujo de trabajo de la metodología de Cascada.....	19
Ilustración 4 - Ciclo de vida de la metodología Espiral	20
Ilustración 5 - Diagrama de clases.....	22
Ilustración 6 - Ciclo de vida del PU	23
Ilustración 7 - Características de metodologías ágiles y tradicionales	25
Ilustración 8 - Ítem	27
Ilustración 9 - Primera iteración	27
Ilustración 10-Estructura de Equipo	28
Ilustración 11 - Ciclo de vida	39
Ilustración 12 - Objetivos de las fases	40
Ilustración 13 - Diagrama de red: Red estrella	50
Ilustración 14 - Bosquejo.....	51
Ilustración 15 - Diagrama de implementación	52
Ilustración 16-Diagrama Entidad-Relación	53
Ilustración 17 - Caso de uso con especificación.....	54
Ilustración 18- Caso de uso múltiple	55
Ilustración 19 - Diagrama de actividad con responsabilidad.....	56
Ilustración 20 - Lista de tareas - CTSalta	75
Ilustración 21 - Lista de tareas - Dentis.....	76
Ilustración 22 - Equipo PucaráTech	80
Ilustración 23 - Equipo Pucará. De izquierda a derecha, Alberto, José, Pablo y Sebastián	81
Ilustración 24 - Panel de control: Lista del Proyecto	82
Ilustración 25 - Ítem en Taiga	83
Ilustración 26 - Ítem en papel	83
Ilustración 27 - Tarea en Taiga.....	84
Ilustración 28 - Tarea en papel	84
Ilustración 29 - Tablero de corcho.....	85
Ilustración 30 - Tablero en Taiga	85
Ilustración 31 - Extracto de la Lista del Proyecto	86
Ilustración 32 - Iteración 10 de Factura Electrónica.....	86
Ilustración 33 - Representación de un Error en Taiga	87
Ilustración 34 - Error en papel.....	87
Ilustración 35 - Diagrama de red y comunicación. Gentileza de Pucará.....	92
Ilustración 36 - Lista del Proyecto.....	93
Ilustración 37 - Modelo para el Dashboard	97
Ilustración 38 - DER para el módulo de Login	98
Ilustración 39 - Error de PucaráFactura.....	102
Ilustración 40 - Índice del manual de usuario CT Salta	103
Ilustración 41 - Retrospectiva del día 15/07/2016.....	106

Ilustración 42 - Resumen de Retrospectiva del 01/07/2016	107
Ilustración 43 - Resumen de estadísticos.....	108
Ilustración 44 - Tiempo de Desarrollo más soporte de la iteración 6.....	108
Ilustración 45 - Tiempo de Desarrollo/Bugs	109
Ilustración 46 - Puntos por iteración.....	110
Ilustración 47 - Puntos estimados/completados por iteración	111
Ilustración 48 - Velocidad por iteración	112
Ilustración 49 - Estadísticos varios de José para la cuarta iteración.....	112

Marco de Trabajo para el desarrollo de Software

Introducción

En los últimos años se viene desarrollando software tanto con metodologías ágiles como tradicionales, cada una de estas brinda ventajas en ciertos ámbitos; con frecuencia, en base a los requerimientos de software se requieren desarrollos rápidos y cortos, para un software sencillo y a medida del cliente, en estos casos las metodologías ágiles son la mejor herramienta en cuanto a marco de trabajo disponible en la actualidad, sin embargo, esto no significa que ágiles sea una panacea, no lo es para desarrollos pequeños y mucho menos lo es para desarrollos grandes y con requerimientos rígidos; en estos tipos de proyecto, las metodologías tradicionales tienen ventajas sobre las ágiles, principalmente por la rigidez de los requerimientos, pero se presentan problemáticas similares, la principal es visible en cuanto a equipos y RR. HH., por la nula rotación y repetición de tareas.

En base a estos hechos se desarrollará un Marco de Trabajo balanceado en cuanto a prácticas efectivas de metodologías ágiles y tradicionales, proporcionando artefactos, roles, documentos, eventos y un conjunto de lineamientos sobre los cuales actuar, buscando obtener un rendimiento óptimo en un rango amplio de proyectos.

Objetivos

El principal objetivo de esta tesis es proporcionar un Marco de Trabajo para equipos en búsqueda de una herramienta que sea balanceada en cuanto a técnicas ágiles y tradicionales. El Marco integra todos los actores involucrados en el desarrollo de un proyecto de software, permitiendo escalabilidad, y brindando la posibilidad de manejar equipos distribuidos.

Se pretende un Marco de Trabajo que sea eficiente en el complejo entorno moderno abarcando de esta manera una gran variedad de proyectos en los cuales puede ser efectivo. Todo esto con el objeto de mejorar la definición general del proyecto y la previsibilidad de la entrega.

El desarrollo de este trabajo aportará la forma de afrontar todas las etapas de desarrollo. Especificando artefactos referidos a manejo de tareas y procedimientos; proporcionando una serie de lineamientos generales sobre los cuales actuar, influenciados por la aplicación de técnicas de ingeniería del software.

Alcance

El presente proyecto proporcionará un desarrollo teórico del Marco de Trabajo propuesto, definiendo sus características, roles, documentos y eventos. La forma de accionar con respecto a las etapas de un proyecto de desarrollo de software. Se especificará también el alcance teórico recomendado para cada rol y evento definido, estableciendo de esta manera un lineamiento sobre el cual se debe manejar un equipo ante la tarea del desarrollo de Software.

La verificación experimental del Marco de Trabajo se realiza en base a la descripción de su aplicación en PucaráTech, describiendo elementos claves de su implementación y fundamentando resultados en base a estadísticas recolectadas.

Brief

Marco de trabajo para el desarrollo de software basado en metodologías tradicionales y ágiles, enfocado a Equipos pequeños y medianos.

Marco teórico

El conjunto de conocimientos y tecnologías que posibilitarán el desarrollo de este trabajo abarca estudios de Scrum, XP, Kanban, Manifiesto Ágil, y Metodologías tradicionales (Cascada, Espiral y Proceso unificado).

Por lo general podemos encontrar que en todas las metodologías se distinguen Roles, Artefactos y un conjunto de procedimientos que definen como afrontar las diferentes tareas necesarias para la resolución de conflictos.

También se encuentran marcos de trabajo y un conjunto de recomendaciones y buenas prácticas que recomiendan su aplicación para lograr calidad en cuanto al producto, el ambiente organizacional y la previsibilidad del proyecto.

Scrum

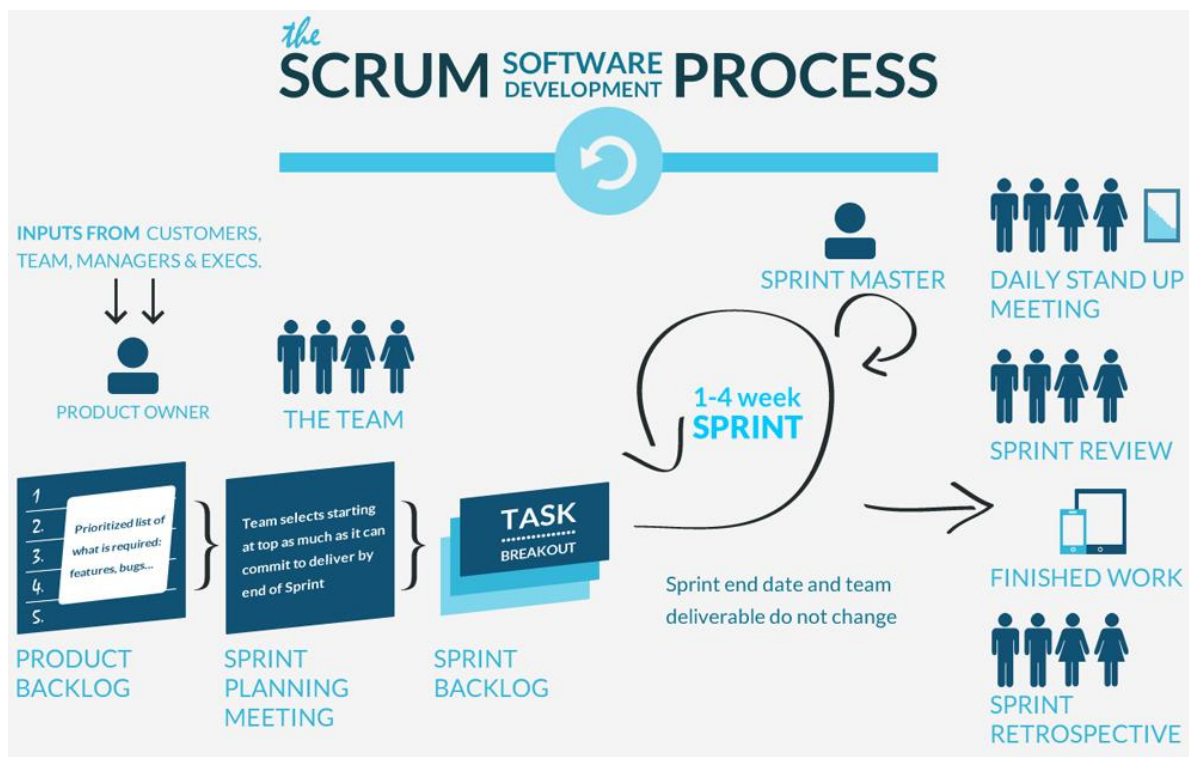


Ilustración 1 - Representación gráfica del proceso de Scrum¹

Scrum es un marco de trabajo ágil destinado a equipos de desarrollo pequeños (< 9 personas), enfocado en proyectos complejos con requerimientos aparentemente ambiguos, donde sea necesaria la comunicación constante con los stakeholders, con el objetivo de proporcionar entregas de software funcional entre periodos cortos y fijos, llamados sprints.

¹ Esta imagen fue descargada de <https://www.augustinfotech.com/scrum-effective-way-implement-agile/> en el 2016

Roles

Stakeholders

El conjunto de personas que se verán afectadas por el producto final son los Stakeholders.

Product Owner

Esta persona es el intermediario entre los Stakeholders y el equipo de programación, esta persona es la encargada de conocer las necesidades del cliente de la mejor forma posible.

Scrum Master

Es un facilitador, es decir, es el encargado de ayudar a resolver cualquier problemática en la cual se encuentre el equipo, como así también de ayudar en la organización de reuniones y la utilización de artefactos.

Equipo

El conjunto de personas multifuncional, experimentado y auto-organizado que tiene como única responsabilidad el generar un entregable a partir del conjunto de requerimientos determinados en el Sprint Backlog.

Artefactos

Scrum tiene un conjunto de artefactos base que se usan para cumplir diferentes objetivos, ellos son, Product Backlog, Sprint Backlog, Burndown chart, Historias de Usuario, Tareas (tasks).

Historia de Usuario

Una Historia de Usuario es un requerimiento escrito en el lenguaje del usuario. Esto permite entender la funcionalidad desde una perspectiva más cercana al usuario. Este requerimiento va acompañado de una estimación de esfuerzo, criterios de aceptación y comentarios adicionales.

Product Backlog

El Product Backlog es una lista priorizada, dinámica y pública de todas las historias de usuario que llevarán al desarrollo completo del software. Esta lista es priorizada y mantenida únicamente por el Product Owner.

Sprint Backlog

El Sprint Backlog es un subconjunto del Product Backlog, y contiene las Historias de Usuario que se van a desarrollar en la iteración actual.

Para cada Historia de Usuario se deberán definir todas las tareas técnicas (Tasks) necesarias para su desarrollo.

Estados

Inicialmente se establece un conjunto básico de estados en los cuales puede estar una tarea o historia de usuario, estos son: Pendiente/Nueva, En curso, Terminado.

Pendiente/Nueva: Las tareas que se encuentran en este estado están todavía en espera de ser ejecutadas.

En curso: Indica que la actividad se está desarrollando, no indica en qué etapa del desarrollo está.

Terminado: Las actividades que se coloquen bajo esta etiqueta indican se han terminado de desarrollar. En algunas organizaciones el terminado indica que la tarea está lista para testing, en otras esta tarea se desarrolla en otro momento mediante Historias de Usuario o Tareas.

Reuniones

Sprint planning

Es una reunión de duración intermedia (2 a 4 horas) que se realiza al comienzo de cada sprint, en ella participa el Scrum Master, el Equipo y el Product Owner. Los principales objetivos de esta reunión son:

- Que el equipo entienda con la mayor precisión posible las necesidades de los Stakeholders.
- Determinación por parte del equipo de los ítems que se desarrollaran en el Sprint, definiendo las Tareas necesarias para completarlos, definiendo de esta manera el Sprint Backlog.
- Realizar una estimación de dificultad y tiempo de cada actividad.

De ser necesario el Product Owner puede pedir que se cambie ligeramente el alcance de Sprint Backlog.

Daily meeting

Son reuniones cortas que se realizan por lo general al comienzo de la jornada laboral. Tiene como principal objetivo determinar las actividades realizadas en el día anterior, qué actividad se va a realizar a continuación y si tiene algún problema o impedimento en el desarrollo del trabajo.

Sprint review

Es una reunión que se realiza una vez terminadas las actividades de desarrollo. En esta reunión el equipo demuestra el avance realizado, mediante la visualización de las funcionalidades desarrolladas el Product Owner (y Stakeholders) puede rechazar o aceptar la entrega; como así también proponer mejoras y modificaciones.

Sprint Retrospective

Es una reunión dinámica, de duración intermedia y representa la última actividad del sprint, en ella participa el Scrum Master y el Equipo, con el objetivo de determinar las oportunidades de mejora y resaltar las fortalezas detectadas por el equipo mediante diversas técnicas.

Kanban

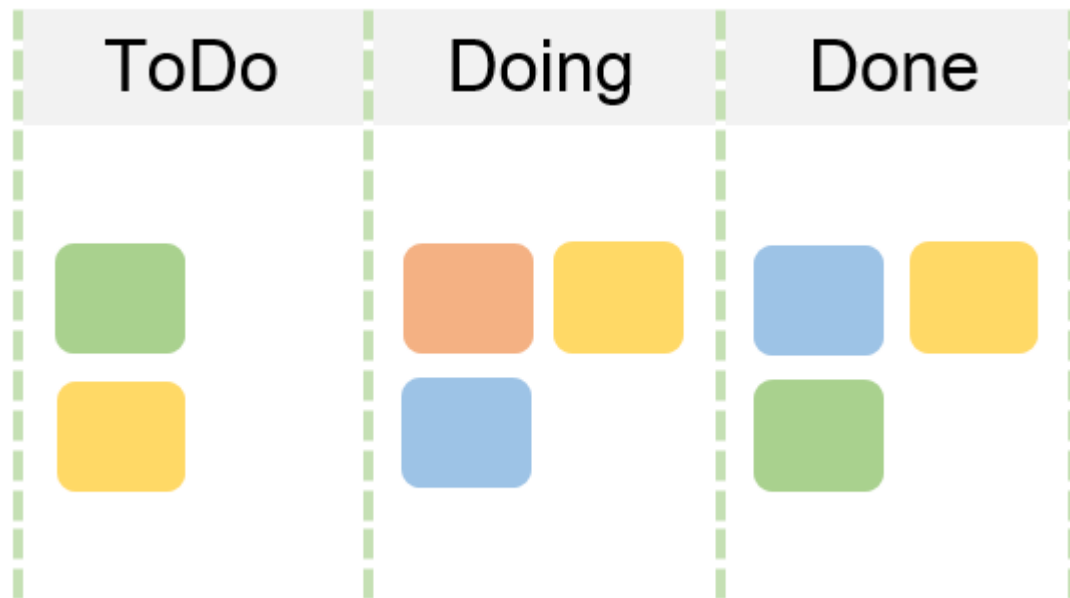


Ilustración 2 - Tablero de Kanban²

Según David Anderson.

Un sistema Kanban es un sistema o proceso diseñado para disparar trabajo cuando hay capacidad para procesarlo. El disparador se representa con una tarjeta. Se pone a disposición una cantidad de tarjetas correspondiente a la capacidad del sistema. Una tarjeta se adjunta a cada ítem de trabajo. La tarjeta sirve como mecanismo visual. Un nuevo ítem de trabajo puede iniciarse únicamente cuando se dispone de una tarjeta libre. Esta tarjeta libre se adjunta al ítem de trabajo para que pueda avanzar a través del sistema. Cuando no hay más tarjetas libres, no se pueden iniciar nuevos trabajos. Cuando el trabajo se termina, la tarjeta es separada del ítem y reutilizada. Este mecanismo es conocido como sistema “Pull”, porque un nuevo trabajo es introducido en el sistema (“Pulled”) únicamente cuando hay disponibilidad para procesarlo, en lugar de ser introducido (“Pushed”) en el sistema en función de la demanda.

Podemos entender que Kanban muestra el trabajo en curso, limita el trabajo en curso (WIP, Work In Progress) optimizando el flujo de actividades.

La forma en la cual Kanban realiza una Generación de Valor (desarrollo de un software a partir de un pedido) consta de varios pasos, estos son:

² Esta imagen fue descargada de <https://www.digite.com/kanban-guide/what-is-a-kanban-board> en el 2016

Elementos

Puntos de control

Los puntos de control son aquellos que se definen en base al área de incumbencia del equipo de desarrollo, estos puntos se colocan de forma visible en un tablero y delimitan el comienzo y fin de una actividad, por ej. Análisis, Diseño, Codificación, Testing.

Tipos de ítems de trabajo

Estos ítems están determinados por las actividades que se realizarán para la Generación de Valor por parte del equipo, los tipos de ítems más comunes son: Funcionalidades, Mejoras/Cambios, Bugs, Refactoring, Mantenimiento.

Tarjetas de ítems de trabajo

Los ítems de trabajo van a estar presentes en tarjetas visuales, las cuales van a contener información referida a fechas de introducción y retiro, responsable, tipo de ítem, prioridad, fecha máxima de terminación y algún comentario referido a la actividad.

Secuencia de actividades

En base a la definición de los puntos de control se determinan las fases por las cuales va a pasar una tarjeta de trabajo, estos estados pueden definirse de forma individual o general para los puntos de control establecidos. Los estados más comunes son *Pendiente*, *En curso* y *Terminado*.

Tablero

Todos estos artefactos son colocados en un mismo tablero que contendrá esta información de forma visual y ordenada.

Limitación del WIP

Un concepto importante en Kanban es la limitación del trabajo con el objetivo de reducir cuellos de botella generados por la acumulación de actividades en puntos de control. Esta limitación se hace a través de una especificación de la cantidad máxima de actividades que pueden estar de forma concurrente en un punto de control.

Optimización del flujo

El lograr un flujo de desarrollo estable, previsible y adecuado a las necesidades del negocio es una de las características a cumplir fundamentales para lograr compromiso y confianza en un sistema de Kanban.³

³ UNIDAD 04 Gestión ágil de Proyecto con Kanban y Extreme Programming UTN

Para tratar de cumplir este objetivo, el equipo cuenta con diversas métricas, las más importantes son: Cycle Time; Throughput y; Diagrama de Flujo Acumulado (CFD).

Manifiesto ágil

El pilar más importante sobre el cual se plantea el presente trabajo son los principios y valores presentados por el manifiesto ágil, estos establecen un marco definiendo características generales de roles y personas, la forma de afrontar el trabajo y la manera en la cual se manejan los artefactos.

Valores

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Principios

Los siguientes principios son aquellos que surgen de forma derivada de los valores anteriormente descritos.

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para la continuación ajustar y perfeccionar su comportamiento en consecuencia.

Cascada

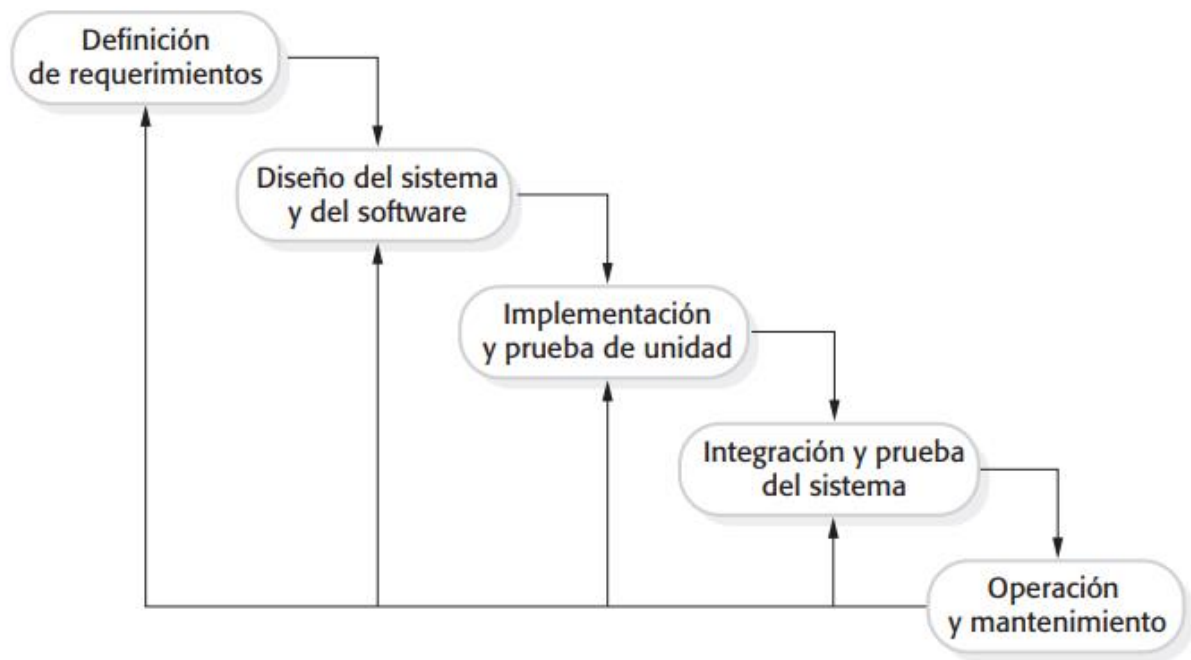


Ilustración 3 - Flujo de trabajo de la metodología de Cascada⁴

La metodología en Cascada es una de las sencillas de entender, y fáciles de implementar, a partir de un acercamiento tradicionalista de ingeniería se definen un conjunto de fases por las cuales debe transitar el proyecto para poder terminar de manera exitosa. Así el modelo en Cascada es un ejemplo de un proceso dirigido por un plan; donde hay que planear y programar todas las actividades del proceso, antes de comenzar a trabajar en ellas.

Etapas

Las etapas del modelo en Cascada reflejan directamente las actividades de desarrollo que involucran, así tenemos:

Análisis y definición de requerimientos

Los servicios, restricciones, y las metas del sistema se establecen mediante consulta a los usuarios del sistema. Luego, se definen con detalle y sirven como una especificación del sistema

La información que es obtenida en esta etapa tiene carácter definitivo y no puede ser cambiada en etapas posteriores.

Diseño del sistema y del software

⁴Imagen obtenida del libro Ingeniería del Software - Ian Sommerville 9° Edición

El proceso de diseño de sistemas asigna los requerimientos necesarios para el proyecto, al establecer una arquitectura de sistema global. El diseño del software implica identificar y describir las abstracciones fundamentales del sistema de software y sus relaciones

Implementación y prueba del sistema

En base a la información obtenida en la etapa de Diseño se procede a la codificación de la solución, como un conjunto de programas o unidades del programa. La prueba de unidad consiste en verificar que cada unidad cumpla con su especificación.

Integración y prueba de sistema

Las unidades del programa, o los programas individuales se integran y prueban como un sistema completo para asegurarse de que se cumplan los requerimientos de software. Después de probarlo, se libera el sistema de software al cliente.

Operación y mantenimiento

En esta etapa el sistema se instala y se pone en práctica. El mantenimiento incluye corregir los errores que no se detectaron en etapas anteriores, mejorar la implementación de las unidades del sistema e incrementar los servicios del sistema conforme se descubran nuevos requerimientos.

Espiral

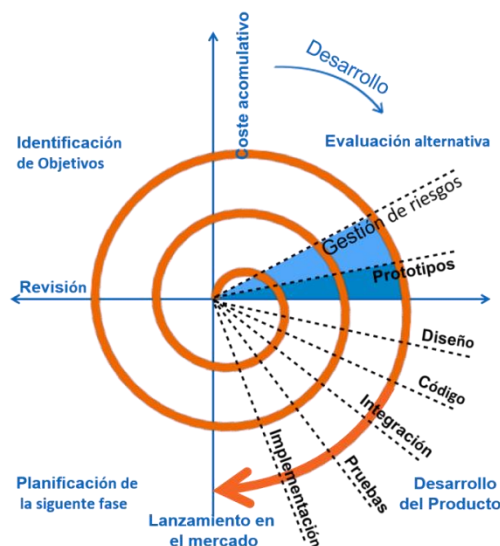


Ilustración 4 - Ciclo de vida de la metodología Espiral⁵

El modelo en espiral es un marco de trabajo que se centra en hacer entregas secuenciales del software teniendo en cuenta un análisis del riesgo al hacer una nueva iteración.

⁵Esta imagen fue descargada de http://www.tutorialspoint.com/sp/software_engineering/software_development_life_cycle.htm en el 2016

Fases

Para lograr el objetivo de entrega secuencial se tienen en cuenta 4 fases, estas son:

Identificación de objetivos

En esta fase es clave la comunicación con el cliente, ya que tiene como objetivo definir los requerimientos del software estableciendo una definición clara y detallada de las funcionalidades, asimismo es necesario definir los riesgos, como así también establecer la gestión del proyecto.

Evaluación alternativa

A partir de los riesgos identificados en la etapa anterior, se procede a analizarlos en detalle y a establecer planes de reducción y de contingencia.

En esta etapa también es necesaria la creación de prototipos y/o modelos de simulación sobre el software a desarrollar para establecer de esta manera un análisis más completo.

Desarrollo del producto

A partir de los requisitos se realizan las tareas de diseño, codificación, prueba e implementación del software.

Planificación de la fase siguiente

Es la última etapa de la iteración, en ella se revisa el proyecto poniendo énfasis en el análisis de riesgos y se toma la decisión de continuar o no.

UML

UML es un lenguaje de modelado con un amplio alcance, ya que permite cubrir una gran cantidad de proyectos de diversas aplicaciones y dominios. El objetivo de UML es proveer a los arquitectos, ingenieros y desarrolladores de aplicaciones con herramientas para el análisis, diseño e implementación, permitiendo modelar los procesos técnicos de software como aquellos orientados al negocio.

Los conceptos de modelado de UML están agrupados en unidades de lenguaje. Una unidad de lenguaje es una colección de modelos relacionados que permiten representar aspectos del sistema en base a un determinado paradigma. La agrupación en unidades de lenguaje hace posible que el usuario diagrame lo que necesite sin conocer UML en su totalidad.

La dimensión de UML tiene como consecuencia que no todas sus herramientas son útiles para todos los dominios y aplicaciones, la elección de las herramientas que maximicen la efectividad en el proyecto es de crítica importancia.

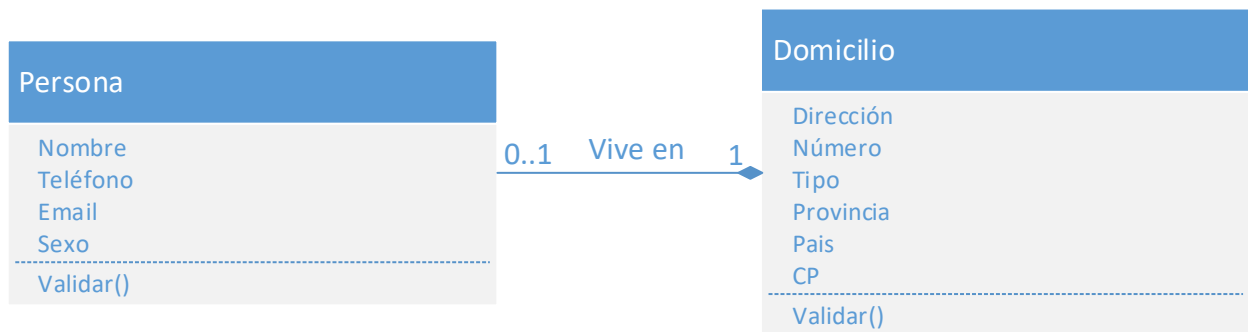


Ilustración 5 - Diagrama de clases

Proceso unificado

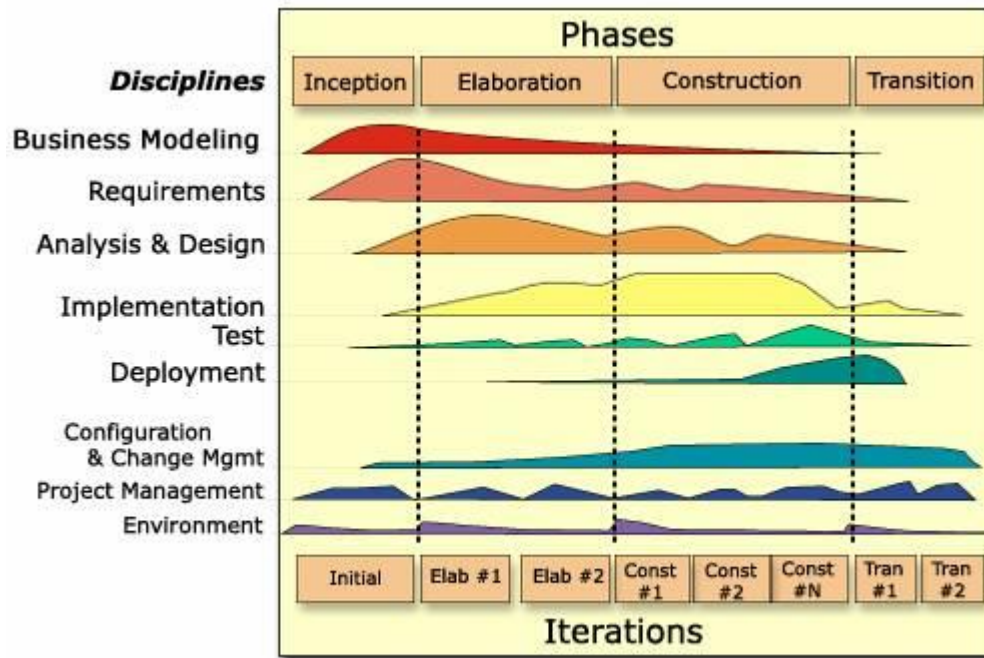


Ilustración 6 - Ciclo de vida del PU⁶

El Proceso unificado es una metodología de desarrollo de software iterativa e incremental, basada en componentes e interfaces. Hay dos implementaciones del Proceso unificado, Open Unified Process (UOP) y Rational Unified Process (RUP) de IBM, esta última es la que describiremos en este marco teórico.

Elementos

Trabajadores o actores: Define el comportamiento y responsabilidades de un individuo, grupo de individuos o sistema.

Actividades: Es una tarea con un propósito claro, la misma manipula artefactos y es realizada por un Actor o Trabajador.

Artefactos: Son elementos del proyecto que han sido producidos, modificados y usados por las Actividades, estos elementos pueden ser modelos, diagramas, código fuente, base de datos, etc.

Flujo de actividades: Es una secuencia de actividades realizada por los Trabajadores y tiene como resultado un incremento de valor.

⁶ Esta imagen fue descargada de <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm> en el 2017

Características

- Dirigido por Casos de Uso: Los Casos de Uso reflejan las necesidades y deseos de los usuarios. Los casos de uso representan los requisitos del sistema y guían el proceso de desarrollo.
- Centrado en la Arquitectura: La arquitectura consiste en los diferentes modelos y vistas que representan la Arquitectura del Sistema.
- Iterativo e incremental: El proceso unificado produce mediante iteraciones, en cada iteración el proyecto pasa por cada una de las 4 fases: Inicio, Elaboración, Construcción y Transición. Es incremental

Fases

Inicio: Aquí se define el dominio del proyecto, su factibilidad, metas, plazos, costos y una visión. El esfuerzo se encuentra enfocado en el modelaje del negocio y el análisis de los requerimientos.

Elaboración: En la fase de Elaboración se completa el análisis de los casos de uso y se define la arquitectura del sistema, obteniendo un ejecutable

Construcción: Representa las actividades evolutivas necesarias para completar los requisitos necesarios para considerar al producto como listo

Transición: En esta etapa el producto se prueba, instala y posteriormente es utilizado por el cliente.

Resumen

Evidentemente hay muchas formas de encarar el desarrollo de software, cada una de las metodologías de desarrollo afrontan el trabajo con una visión diferente, esas diferencias, en algunos casos tienen apariencias similares, como el tratar con iteraciones (espiral, RUP, Scrum). Todas estas metodologías tienen un sector acotado de proyectos en donde realmente son útiles, este rango se ve influenciado por el tiempo, costo, alcance y personas que componen el proyecto.

Ventajas y desventajas

Es difícil determinar qué características de cada filosofía de desarrollo representan ventajas o desventajas, lo que haremos es presentar las características principales de las dos filosofías y el lector decidirá si la misma representa una ventaja o desventaja.

Antes de realizar eso tendremos que realizar una distinción entre los dos grupos principales, las metodologías ágiles y las tradicionales. Una metodología ágil será aquella que se base en el Manifiesto ágil, en esta categoría tendremos a Scrum, XP, y Kanban entre otros.

Agiles	Tradicionales
Preparadas para el cambio	Resistencia al cambio
Basadas en las personas	Basadas en planes y procesos
Proceso de trabajo basado en principios y en pocas reglas	Proceso de trabajo altamente controlado y planificado
El cliente es un miembro más del equipo	El cliente trabaja con el equipo de forma más normalizada
Bajos niveles de documentación	Altos niveles de documentación
Pocos roles y artefactos	Mayor cantidad de roles y artefactos
Eficaces para proyectos del complejo entorno moderno	Ineficaces para proyectos del complejo entorno moderno
Menos eficientes para proyectos restrictivos y sin cambios	Más eficientes para proyectos restrictivos y sin cambios
Centradas en el código	Centradas en la arquitectura
Laxa definición de responsabilidades y roles	Buena definición de responsabilidades y roles
El incremento es tangible por el cliente en periodos cortos	El incremento es tangible por el cliente “de repente”
Los clientes no están familiarizados de antemano con la forma de trabajo	Los clientes reconocen inmediatamente la forma de trabajo.
Para un funcionamiento óptimo requieren contratos ágiles	Contratos tradicionales
Ideado para grupos pequeños de personas	Ideado para grupos medianos o grandes de personas
Menor burocracia	Mayor burocracia
Estructura jerárquica horizontal	Estructura jerárquica vertical

Ilustración 7 - Características de metodologías ágiles y tradicionales

Método de trabajo a utilizar

La forma de trabajo que se utilizará para el desarrollo de este proyecto está conformada por un conjunto de elementos pertenecientes a varios marcos, principalmente aquellos de metodologías ágiles e iterativas, obteniendo así las ventajas referidas a un trabajo incremental y enfocado en la calidad.

Ya que el trabajo teórico tendrá la participación de dos actores, se modifican los roles y procedimientos tradicionales que se pueden encontrar en la mayoría de las directivas de trabajo, ya que estos están apuntados a un conjunto mayor de personas.

Roles

Investigador

El investigador es la persona encargada de realizar todo el trabajo correspondiente para el cumplimiento de los objetivos planteados.

Consultor

El rol de Consultor es desempeñado por el Profesor Tutor, los Profesores de la cátedra “Proyecto de Grado” y profesores consultores; su responsabilidad consiste en responder preguntas que pueda tener el Investigador referidas a: los lineamientos generales del proyecto; el formato y estructura del trabajo; necesidades de conocimiento o recomendaciones teóricas y de información complementaria.

Reuniones

Reunión de actualización

Estas reuniones son de corta duración y tienen como partícipes al Investigador y al Consultor, en ella, el investigador muestra los avances que se hicieron desde la última reunión y determina qué se presentarán en la próxima reunión. Es aquí cuando el Investigador y el Consultor tienen la posibilidad de plantear inquietudes y recomendaciones sobre la estructura, enfoque y lineamientos del proyecto.

Estos ítems que se priorizan de acuerdo al valor que aportan hacia el cumplimiento de los objetivos generales y posteriormente se documentan en una lista denominada Backlog.

Artefactos

Ítem

The screenshot displays a web interface for a software development framework. The main header shows 'MARCO DE TRABAJO PARA EL DESARROLLO DE SOFTWARE' and 'HISTORIA DE USUARIO'. Below this, a task titled '#8 Fases-Definición' is shown, created by 'José Dousdebes'. The task includes a list of sub-tasks: 'Identificación de las necesidades del Cliente', 'Definición de la visión y objetivos', 'Determinación de la Arquitectura y Herramientas de modelado', 'Análisis de riesgos y definición del plan de contingencias', 'Plan de entregas Estimación de costo y Tiempo', and 'Determinación de necesidades de capacitación'. Below the sub-tasks, there is a section for 'Tareas relacionadas' and '0 adjuntos'. The right sidebar contains a 'PANEL DE TAREAS' button, a 'CERRADA' status, a list of users (UX, Diseñador, Front, Back), a 'puntos totales' section, an 'Asignado a José Dousdebes' section, and a '0 Observadores' section with buttons for 'OBSERVAR' and 'AÑADIR OBSERVADORES'. At the bottom of the sidebar, there is a 'PUBLICAR COMO GIG EN LA TRIBU TAIGA' button and a 'Mas información' section with icons for user, document, lock, and trash.

Ilustración 8 - Ítem

Un ítem es un elemento que describe un tema o tarea que ayuda a alcanzar el objetivo del proyecto de investigación. Adicionalmente, cuando sea requerido, el Investigador puede colocar comentarios adicionales que ayuden a clarificar el ítem.

Lista de ítems

La lista de ítems es una lista priorizada de los ítems que el Investigador se comprometió a trabajar en la Reunión de actualización. El contenido de la misma surge del trabajo pendiente, en ocasiones habrá una dependencia de ítems, en dichos casos será el criterio gobernante.

The screenshot shows a dropdown menu with a downward arrow and the number '1'. Below the dropdown, there is a list of items: '#1 Crear índice' and '#2 Marco teórico'. The items are separated by horizontal lines.

Ilustración 9 - Primera iteración

Solución propuesta

Personas y procesos

Las personas son la parte más importante de cualquier organización, esto se amplifica aún más si es en el ámbito de la informática. Esta afirmación es verdadera para todas las metodologías y marcos de desarrollo existentes, desde las formales a las informales, este caso no será la excepción. La definición de cada uno de los elementos de este marco de desarrollo está influenciada por las ventajas de las metodologías analizadas, sumado a conocimientos de otras áreas.

Para que las personas interactúen de forma eficiente en un equipo de trabajo tienen que estar bien definidos tres conceptos principales: los roles y, los derechos y responsabilidades que traen acarreados los mismos.

Roles

En este Marco de trabajo proponemos un Equipo formado por 4 + 1 roles, el Líder del Proyecto, Encargado del Producto, Arquitecto y el Desarrollador conformarán el Equipo en el sentido más tradicional de la palabra, es decir, son aquellas personas que pertenecen a la organización. Decimos 4+1 roles porque se tiene en cuenta al Cliente como un rol muy importante en el ciclo de vida del proyecto, dejarlo afuera sería un error; no decimos directamente 5 roles, porque no participa de forma constante en las actividades del Equipo.

El diagrama siguiente representa la composición de Equipo propuesta por este Marco:

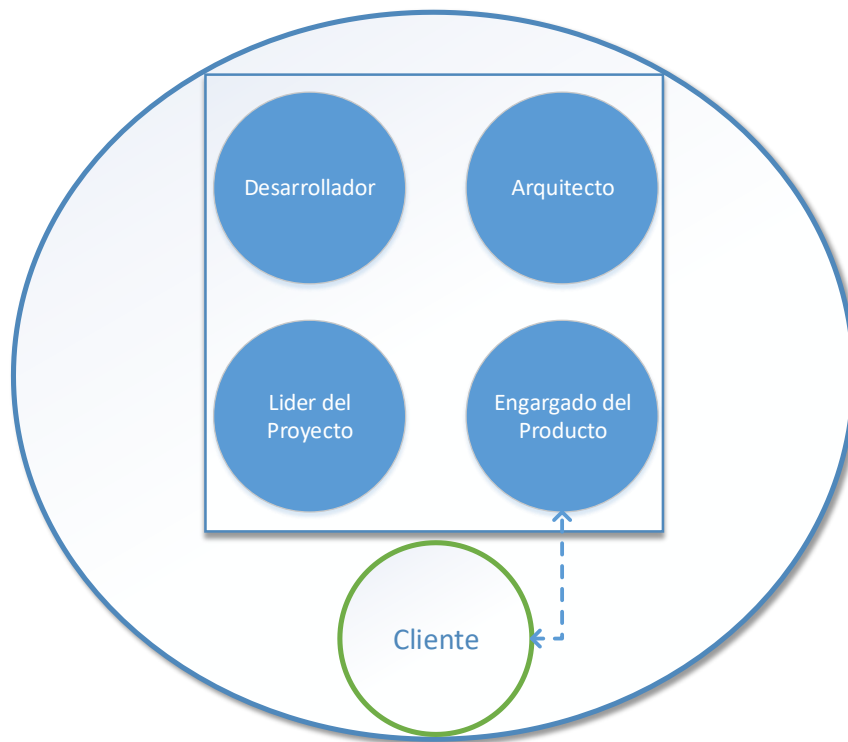


Ilustración 10-Estructura de Equipo

Los roles que propone este Marco están destinados a satisfacer las necesidades de todos tamaño de Equipos, estando especialmente diseñado para el tamaño de una organización promedio, que es de menos de 10 personas⁷. Sin embargo, a través de la separación de responsabilidades en roles se logra una excelente escalabilidad.

Líder de proyecto

El líder de proyecto o LP es una persona que cumple el rol de líder servicial, adoptando cuando sea requerido el rol de líder anfitrión. Ayuda a crear y mantener un ambiente que permita al Equipo cumplir con los objetivos planteados, y optimizar la eficiencia grupal.

Las tareas sobre las cuales tiene que tener responsabilidad son:

- Mantener un ambiente de comunicación y cooperación entre todos los roles.
- Asistir y facilitar la resolución de problemas y consultas que tenga el Equipo.
- Aislar al Equipo de interrupciones externas.
- Mantener al Equipo enfocado en las metas.
- Crear y mantener un espacio de aprendizaje y creatividad dentro del Equipo.
- Asegurar la correcta utilización del Marco de Trabajo y otras prácticas de desarrollo en el Equipo.
- Participar y facilitar todas las reuniones.
- Ayudar y asistir al Encargado de Producto, Arquitecto y Cliente en la definición de la Visión del proyecto.
- Elaborar informes estadísticos y plan de mejoras para el Equipo.
- Determinar las directivas de UX en el proyecto.
- Asegurarse del correcto funcionamiento de los elementos de comunicación y colaboración.
- Guiar a los diferentes roles para la óptima utilización de las herramientas de comunicación y cooperación.
- Elaborar planes de contingencia para los distintos niveles involucrados en el Proyecto.
- Determinar necesidades de documentación especiales para casos específicos.
- Contribuir en la definición y el análisis de riesgos
- Contactar a staff especializado.

Una de sus responsabilidades más importantes es la de ser el encargado de Experiencia de Usuario (UX), esto implica la definición de directivas generales de UX, sobre las cuales se desarrollarán los requerimientos establecidos en los Ítems. Esto es posible porque tiene conocimiento de las mejores prácticas y últimas tendencias referidas al desarrollo y definición de funcionalidades de software que permitan al usuario hacer uso óptimo del mismo.

En ocasiones, el Líder del Proyecto puede llegar a tener tiempo “libre” en una jornada laboral completa, en estos casos tiene permitido tomar otros roles, en este Marco de Trabajo

⁷[QSM - Team Size Can Be the Key to a Successful Software Project](#)

recomendamos que no ocupe el rol de Encargado del Producto, principalmente porque este último tiene que estar al tanto (de la forma más actualizada posible) de las necesidades de los Clientes y esto, sumado a sus otras responsabilidades suele ocupar la totalidad de su tiempo; el Líder del Proyecto tiene que estar constantemente en el Equipo.

En base al modelo de los 5 Grandes de la Psicología podemos concluir que un buen LP tendrá bajos niveles de Neuroticismo, lo que proporciona una mejor habilidad para resolver problemas bajo presión, y alta Amabilidad, una característica definitiva en la colaboración. Esos son los rasgos necesarios, sin embargo es un aditivo contar con buenos niveles de Apertura.⁸

Desarrollador

Un Desarrollador es toda persona que participe en el desarrollo “real” de la solución propuesta. Para lograr este objetivo realiza el análisis, diseño, programación, estimación, testing y todas las tareas que el grupo considere apropiadas para cumplir la meta.

Además de las tareas detalladas anteriormente un Desarrollador tiene como responsabilidad:

- Consultar al Encargado del Producto sobre el dominio del proyecto.
- Identificar tareas sobre los Ítems.
- Realizar una estimación sobre el tiempo y esfuerzo (VD) que llevará completar una Tarea o Ítems.
- Mantener actualizado al equipo sobre el estado de sus actividades, en las reuniones diarias y en el tablero de control.
- Colaborar con el Arquitecto al momento de definir y/o modificar la arquitectura.
- Estar dispuesto a aprender habilidades fuera de su campo de experticia.
- Ser proactivo para detectar conflictos o problemas e identificar soluciones a los mismos.
- Mantener un ambiente de comunicación constante dentro del equipo.
- Contribuir en la definición de riesgos

El objetivo en cuanto al conocimiento y habilidades es lograr un Equipo multifuncional, logrando que todas las personas tengan todas las habilidades presentes en el Equipo en su conjunto, en mayor o menor medida, sin dejar de lado sus especialidades.

En los comienzos del Equipo es poco probable que se de esta situación, por lo tanto, es necesario crear un ambiente de comunicación constante entre los miembros para que, sumado a un ambiente de rotación de actividades se logre un Equipo donde todas las personas tengan conocimientos generales del cúmulo total de habilidades. Con esto se busca llegar a que un miembro, ante un problema fuera de su área de experticia se pregunte “¿Cómo puedo hacer para ayudar?” en vez de “Ese no es mi problema”.

Una consecuencia de este objetivo es que, si en algún momento se ausenta una persona, sus tareas puedan llegar a ser realizadas por otra persona del Equipo.

⁸[Dr. Jordan B. Peterson - Introducción a la Psicometría, 2017](#)

Arquitecto

El Arquitecto es la persona encargada de liderar la definición y evolución de la estructura técnica y de diseño de la solución. Para poder cumplir este objetivo es necesario que esta persona tenga amplios conocimientos técnicos y del dominio del Proyecto.

La estructura técnica del proyecto va a constituir el conjunto de conocimientos y elementos tecnológicos necesarios para cumplir con los requerimientos de la solución. Este conjunto establece la forma en la cual se va a organizar modularmente la solución.

La estructura del proyecto se encarga de sentar las bases sobre las cuales la solución se relaciona y mantiene fiel a los lineamientos tanto de la organización que la desarrolla, como a la que está destinada.

Para lograr los objetivos propuestos es necesario que el Arquitecto lidere la definición de estas estructuras generales, con participación de otros miembros del Equipo.

En base a esto podemos definir a responsabilidades del Arquitecto como:

- Liderar la definición de la Arquitectura de la solución asegurándose de que sea fácil de mantener.
- Colaborar con el Encargado de Producto, Líder del Proyecto y Cliente en la definición de la Visión del proyecto.
- Educar a Desarrolladores en cuestiones referidas a la Arquitectura.
- Mantenerse informado sobre mejores prácticas y asegurarse de que se empleen cuando sea conveniente.
- Liderar el desarrollo del plan de contingencia técnico realizado de forma conjunta con el LP.
- Guiar el desarrollo del alcance del proyecto haciendo foco al esfuerzo requerido por la Arquitectura teniendo en cuenta requerimientos no funcionales.
- Asegurarse de que los activos tecnológicos (como patrones, frameworks, código, etc.) que haya producido la organización con anterioridad sean usados y no se creen nuevos activos redundantes.
- Contribuir en la definición y el análisis de riesgos

En todos los proyectos menos aquellos muy grandes y complejos, las responsabilidades del Arquitecto no cubrirán la totalidad del tiempo de una jornada laboral completa, en estos casos esta persona asume también otros roles, recomendablemente como Desarrollador o Líder del Proyecto, para mantenerse en niveles óptimos de comunicación con las personas que implementan la Arquitectura; por otro lado, no es recomendable que tome el rol de Encargado del Producto, por demandas de tiempo de este último y porque además la priorización del trabajo puede verse demasiado sesgada por cuestiones técnicas.

Cliente

El Cliente representa a todos los interesados que van a ser afectados de alguna forma por el producto, esto incluye, pero no está limitado a: Usuarios finales; Jefes; Departamento de IT interno.

Es importante la participación y colaboración del Cliente con el Equipo y, en particular que mantenga una comunicación constante con el Encargado del Producto, siendo este su representante dentro del Equipo; el hecho de que el EP sea su representante no implica que esté exento de responsabilidad de participación, sino que su participación puede ser no continua ya que se tiene en cuenta que tiene otras responsabilidades.

Encargado del producto

El Encargado del Producto o EP es quien dentro del equipo de trabajo que representa al Cliente, resolviendo dudas que puedan surgir con respecto a los requerimientos de la solución. Al representar al Cliente representa también sus necesidades, esto se ve reflejado en la priorización de los ítems, tratando de lograr el mayor valor posible.

Para cumplir con sus tareas es necesario que mantenga contacto constante con el Cliente, para que ambas partes se mantengan actualizadas respecto a las necesidades y el avance del proyecto.

Las responsabilidades del Encargado del producto son:

- Encontrar respuestas a tiempo para consultas del Cliente o del Equipo
- Priorizar el trabajo, teniendo en cuenta necesidades del Cliente y del Equipo, maximizando el valor para el Cliente.
- Colaborar con el Arquitecto, Líder del Proyecto y Cliente en la definición de la Visión del proyecto.
- Participar activamente en testings de aceptación.
- Explicar el dominio del Cliente al Equipo.
- Demostrar el entregable al Cliente.
- Negociar el presupuesto, alcance y tiempo del Proyecto con el Cliente.
- Establecer canales de comunicación con el Cliente.
- Contribuir en la definición y el análisis de riesgos

El hecho de que represente las necesidades del Cliente acarrea que también represente las capacidades y necesidades del equipo.

Al ser la persona que tiene mayor contacto con el Cliente, tiene la responsabilidad de representar las capacidades y necesidades del Equipo, por esto es necesario que logre que el Cliente entienda la forma de trabajar, como así también que el Equipo logre un entendimiento de las necesidades del Cliente.

Por lo general una sola persona ocupará el puesto de EP de forma exclusiva, sin embargo, en organizaciones donde se manejen varios proyectos de forma simultánea, estas responsabilidades pueden convertirla en un cuello de botella. En estos casos otra de las personas del Equipo se

tendrá que encargarse de algunas tareas que ocupaba el EP, siendo responsabilidad de ambos la coordinación de esfuerzos; teniendo como máxima evitar ambigüedades sobre responsabilidades.

Organización

La clave para producir software de calidad son las personas. El Equipo es un conjunto de individuos auto-organizados, completamente enfocados en la meta, y que tienen presente que cuentan con el respeto y apoyo de sus colegas, no hay mentalidad individualista, es por eso que se fracasa o se triunfa en Equipo.

Una consecuencia de esto es la disminución del uso de pronombres personales singulares en favor de plurales, la evidencia de esto radica en el presente trabajo, que, si bien fue elaborado por una persona, a la misma le cuesta desarraigarse del “nosotros”.

El LP es la persona encargada de motivar a todas las personas hacia la formación de este ambiente, es una tarea demasiado grande y con muchas variables para llegar a estar en control de una sola persona, sin embargo, un buen LP llegará a este estado de forma más rápida.

Desde un punto de vista externo se verá al Equipo trabajar con una mentalidad de colmena, pero la realidad, como fue vista es mucho más profunda.

Cultura organizacional general

La forma en la cual se relacionan las personas y el trabajo dentro de la organización es crítica para que el Equipo pueda entregar una solución de calidad.

No es secreto que las personas producirán un trabajo de calidad si el proyecto en el cual están involucradas les resulta interesante, trabajando como Equipo y en un entorno libre de distracciones. Esto es posible si todos los roles cumplen sus responsabilidades y si el LP y EP encuentran proyectos interesantes.

Un concepto importante y sobre el cual se basa este trabajo es software libre, idealmente todos los equipos producirían este tipo de software, sin embargo, la realidad indica que este no es el caso y para aprovechar de forma máxima los activos es necesario que todo el software producido sea libre dentro de la misma organización. En base a esto logramos que las personas puedan mejorar, usar y por sobre todo aprender cómo se funcionan los procesos, herramientas, modelos, y otros activos de la organización.

La clave para que en el Equipo se logre un clima de aprendizaje, comunicación, y responsabilidad es la confianza y el respeto. Como consecuencia del respeto y la confianza los miembros del Equipo estarán dispuestos a compartir información y ayudarse mutuamente, a ver los errores como aprendizaje, a confiar en que cada una de las personas realizó un esfuerzo perfecto en sus responsabilidades y quedó nada por hacer para cumplir el objetivo con el que el Equipo se comprometió. Como dijo Norman Kerth, “... Entendemos y creemos que todos hacen el mejor

trabajo posible, en base a sus habilidades, conocimientos, recursos disponibles y situación presente.”⁹

Cuando se describió el rol de Desarrollador se mencionó que se intenta lograr un perfil que tenga todas las habilidades, reconociendo especialidades de cada uno. El hecho de que una persona se haya especializado no es producto del azar; esta información debe ser usada a nuestro favor ya que nos indica una preferencia que debe ser aprovechada para producir elementos de mayor calidad, y proveer opciones de aprendizaje a otros miembros del Equipo.

Una de las responsabilidades del Arquitecto es la definición de convenciones, estas juegan un papel incrementalmente importante a medida que el Equipo aumenta en tamaño, sin importar cuáles se elijan es necesario que estén libres de ambigüedades para que se produzca un software fácil de mantener.

Una de los requerimientos organizacionales de este Marco es la eliminación de roles jerárquicos, todos somos iguales en rango, la única diferencia entre miembros radica es la tarea que cumplen, si no existe el EP, nadie sabría qué desarrollar, sin desarrolladores el EP no tendrían qué presentar y el LP no tendría a quien ayudar. Idealmente todas las personas tendrían el mismo sueldo ya que todas cumplen la misma cantidad de horas de trabajo y con igual responsabilidad absoluta.

Locación del Equipo

Lo óptimo es que el Equipo se encuentre bajo un mismo techo, que esto trae acarreadas muchas ventajas en cuanto a las relaciones interpersonales de sus miembros; reduce el tiempo de respuesta ante problemas, mejora la comunicación, la calidad del aprendizaje mutuo es de mayor calidad y más personalizada.

Es posible lograr buenos resultados en un equipo distribuido, en estos casos las tecnologías de comunicación y gestión son clave para un trabajo exitoso; aquí los estándares juegan un papel aún mayor.

⁹<http://retrospectives.com/pages/retroPrimeDirective.html>

Artefactos

Lista del Proyecto

La Lista del Proyecto es una lista priorizada de todos los ítems que componen al Proyecto. La priorización es realizada por el Encargado del Producto con ayuda del Arquitecto, esta priorización está realizada en base a las necesidades del Cliente y de la Solución, el EP buscará maximizar el valor para el Cliente, mientras que el Arquitecto buscará asegurar una demostración temprana del funcionamiento (o no) de la Arquitectura.

Lista de ítems

La lista de ítems es un subconjunto de la Lista del Proyecto, los ítems se seleccionan por el Equipo en su conjunto, la lista resultante comprenderá los ítems que el Equipo se compromete a entregar al finalizar la iteración. La cantidad de ítems está limitada por la Velocidad que haya tenido el Equipo en la iteración anterior, los ítems que componen la lista, si se realizó correctamente la priorización serán aquellos con alta prioridad.

Ítem

Un ítem es una descripción de la funcionalidad o requisito a desarrollar, este ítem tiene una estimación de complejidad y esfuerzo que llamaremos Valor de Dificultad (VD), además contendrá el tiempo necesario; responsables; tareas relacionadas y una descripción adicional en caso de ser necesario aclarar el alcance. Algunos ítems determinados por el Arquitecto también contendrán una lista de criterios de aceptación, los mismos representarán las condiciones que tiene que cumplir el ítem para considerarse como realizado.

Los ítems están restringidos en su granularidad por la cantidad de tiempo estimado para los mismos. Cuando un ítem se estima para un tiempo mayor a 8 horas por participante es necesario desagregarlo, convirtiendo al mismo en un super-ítem.

Es posible asignar dependencias entre ítems ante la necesidad de finalización de uno para el comienzo de otro. Si bien el nivel de encadenamiento es potencialmente infinito, es recomendable adoptar un acercamiento lo más modular posible, minimizando dependencias.

Estados

Un ítem cambia de Estados a medida que avanza el nivel de trabajo sobre el mismo. A continuación, se propone una secuencia que contempla todos los hitos importantes por los que pasa un ítem en su ciclo de vida.

1. Pendiente: Aquel ítem que está dentro de la Lista de ítems de la iteración, pero que todavía no se ha empezado a desarrollar.
2. En progreso: Como indica su nombre, es aquel ítem que se ha empezado a trabajar.
3. Lista para testing: El ítem se terminó de programar y está lista para validación, en algunos casos este estado se puede saltar.
4. Lista o Hecha: Una tarea que ha pasado el testing se puede considerar como lista.

5. Implementada: Representa el último estado, indica que un requerimiento ha sido implementado en producción.

Tareas

Las tareas se desprenden del trabajo técnico necesario para completar un ítem, los desarrolladores son los responsables de definir estas tareas en la reunión de Planificación; el nivel de progreso de las tareas se clasifica según estados, los mismos pueden ser idénticos al de los ítems.

Error

Los errores son un tipo especial de ítems que requiere un trato prioritario. Un error se define con una descripción del inconveniente y la forma de replicar dicho error. El mismo contiene los mismos campos que un ítem, es decisión del Equipo la inclusión del VD. Adicionalmente, un error cuenta con un nivel de Prioridad que refleja la urgencia de su trato.

Tablero

El Tablero es el elemento visual clave del Marco de trabajo, en el tablero se colocan los ítems en base al WIP (Work In Progress) establecido en la Reunión de Planificación. El Tablero tiene representado en columnas las diferentes etapas del ciclo de vida de un ítem de trabajo.

Elementos

Los Elementos que se mencionan a continuación comprenden herramientas y bienes materiales base que permiten el funcionamiento correcto del Marco de trabajo. Es una lista corta, donde se da por sentado la existencia de elementos esenciales para trabajar en cualquier empresa de desarrollo, como por ejemplo: una PC, escritorio, electricidad, herramientas de ofimática y desarrollo, internet.

Sistema de gestión de proyectos

Algo que siempre debe estar presente sin importar si el Equipo es distribuido o no, es un Sistema de gestión de proyectos, la existencia del mismo permitirá mantener un orden que no se podría lograr de otra forma, ahorrará tiempo al momento de elaborar informes estadísticos, facilitará la visualización el estado actual de la iteración. Algunos sistemas proveen de una página wiki que resulta altamente útil para que el Equipo podrá colocar las reglas, convenciones, tutoriales o descripciones de requisitos.

Pizarras

De corcho

Una planchuela de corcho nos va a servir como Tablero si el equipo está reunido en un mismo inmueble, los ítems se escribirán sobre papeles de diferentes colores, el Equipo decidirá si el color representa la persona responsable o el proyecto a que corresponde el ítem, si hay más de uno en simultaneo.

Blancas

Si el equipo está bajo un mismo lugar es imprescindible tener una pizarra blanca grande para realizar esquemas, modelos, problemas y soluciones, las personas tienen la tendencia a formar un enjambre cuando se están tratando estos temas, lo cual es una gran ventaja. Es recomendable tener fibrones de varios colores para una representación más clara.

Útiles

Para algunas personas nada reemplaza a un cuaderno y un conjunto de lapiceras, ayudan muchísimo para tomar todo tipo de anotaciones, “visualizar” un problema o solución, modelar, etc. Es por esto que recomendamos que cada persona se le otorgue un cuaderno y lapiceras.

Sala de reuniones

Una sala de reuniones además de contar con una mesa grande y sillas, va a tener un televisor o un proyector, esto permitirá hacer presentaciones de calidad tanto al Equipo como a Clientes; facilitar al EP explicar necesidades del Cliente, demostrar bugs, realizar video conferencias con staff especializado, visualizar en conjunto el Sistema de gestión de proyectos.

Comida

Aunque constituye un elemento que no contribuye de forma directa al trabajo, es sumamente recomendable tener variedad de alimentos disponibles si se trabaja bajo un mismo techo.

Fases

Las fases representan las estaciones por las que pasa el proyecto durante su ciclo de vida, las mismas están diseñadas para ser transitadas en todas las iteraciones. Al comienzo del proyecto la fase de Definición será la reinante, de forma posterior se transitará por cada una de las 4 fases propuestas, desde las iteraciones 2 a N-1 la fase que ocupará la mayor parte del tiempo será la de Desarrollo, mientras que, en la última iteración, la Implementación será la fase identificadora, pero no exclusiva.

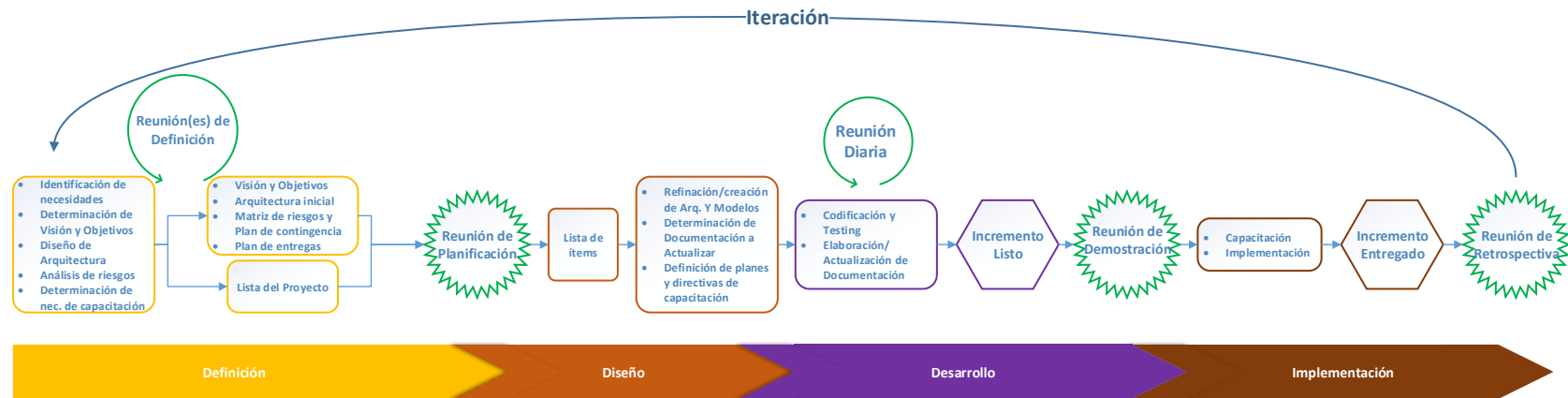


Ilustración 11 - Ciclo de vida

La ilustración11 muestra el Ciclo de Vida del Marco de trabajo.

Definición	Diseño	Desarrollo	Implementación
Identificar las necesidades del Cliente	Elaborar modelos	Producir un incremento entregable de alta calidad	Planificar la implementación
Identificar la Visión y los Objetivos del proyecto	Definir la Arquitectura de la Solución	Manejar necesidades de cambio	Capacitar al Cliente
Establecer lineamientos de Arquitectura	Determinar los ítems a documentar	Realizar tareas de documentación	Desplegar el entregable
Identificar riesgos	Definir planes y directivas de capacitación	Resolver potenciales errores	
Establecer el plan de entregas	Actualizar documentos	Demostrar la arquitectura	
Establecer el contrato			
Crear el Equipo y el ambiente de trabajo			
Determinar las necesidades de capacitación			
Determinar el costo y tiempo del proyecto			
Incrementar y mejorar las habilidades del Equipo Completar los objetivos del Proyecto Manejar riesgos Mejorar procesos			

Ilustración 12 - Objetivos de las fases

En la ilustración 12 se muestran los objetivos de cada fase propuesta, además de estos objetivos individuales, tenemos 4 objetivos comunes a todas las fases.

Definición

En la fase de Definición se establece la planificación general del proyecto, en primer lugar, esto se logra despejando dudas acerca del dominio, para luego poder determinar el alcance, costo y tiempo del proyecto, identificar riesgos y crear planes de contingencia, especificar la arquitectura a emplear, establecer la forma de manejar iteraciones y dejar listo el entorno de trabajo.

El proyecto atraviesa la fase de Definición una o más veces, en ocasiones algunos elementos se realizarán por única vez y no requerirán modificación, sin embargo, hay otros que si las requerirán (como por ejemplo la Lista del Proyecto o la Matriz de Riesgos). Así, podemos identificar que en la Definición conviven elementos que se modifican, ajustan y actualizan de forma continua a lo largo del proyecto, con otros que se definen al comienzo, sirviendo de lineamientos generales y por lo general no se modifican.

Si bien se reconoce que al comienzo del proyecto es el momento donde menos conocimiento se tiene sobre el mismo, es necesario elaborar un plan general ya que nos dará una mejor visión del proyecto y disminuirá nuestras chances de fallar.

En base a estas características, la Definición puede llegar a tomar una iteración para completar todas sus actividades, posteriormente (desde la iteración 2) ocupará una menor parte del ciclo de vida del Proyecto.

Identificación de las necesidades del Cliente

La identificación de las necesidades comienza con un contacto inicial del EP y el Cliente, allí se identificará el tema del proyecto y se establecerán futuras reuniones para identificar necesidades y así poder elaborar la Lista del proyecto.

Una vez que planificadas las reuniones, es necesario determinar las formas en la cual se van a recolectar los requerimientos o necesidades de los Clientes, en todos los casos hay que tener cuidado de no sobre extenderse recolectando requerimientos que estén en un futuro demasiado distante, ya que solo lograremos perder tiempo y malgastar recursos.

Una buena estrategia es recolectar requerimientos de alto nivel y detallarlos en reuniones futuras, esto nos ayudará en la definición de la Visión y Arquitectura, como veremos a continuación.

Al finalizar estas reuniones el EP debería tener una Lista del proyecto priorizada que refleje las necesidades planteadas por el Cliente.

Definición de la visión y objetivos.

Al determinar la visión sobre el proyecto se clarifica el problema que se está tratando de solucionar; entender claramente esto es clave para establecer un punto final al desarrollo y no encontrarse en un ciclo infinito de desarrollo, agobiando de trabajo tanto al Equipo como al Cliente.

Tener en claro la situación a resolver nos permitirá desarrollar 4 conceptos principales en un nivel de granularidad alto, los límites del producto, la Arquitectura de la solución, el Calendario del proyecto y un Presupuesto estimado.

La visión también exige establecer la Forma de Pago, ya que esta gobernará algunos aspectos del desarrollo. Recomendamos el escenario ideal, donde se trabaja con una metodología de pago que tenga en cuenta las iteraciones, tratando de evitar a toda costa los presupuestos iniciales fijos.

Cuando nos referimos al objetivo nos referimos a la determinación de las metas técnicas y humanas que se espera cumpla el producto una vez finalizado, este objetivo es una explicación sencilla escrita en lenguaje de usuario; el objetivo está vinculado estrechamente con el alcance, esta definición se hace a un nivel de granularidad tal que no da lugar a dudas al momento de definir la Lista del Proyecto sobre la cual se determinará el Plan de Entregas.

Determinación de la Arquitectura y herramientas de modelado

A través de la determinación de la Arquitectura se establece la estructura de hardware y software sobre la cual se desarrollará el Sistema. Con la definición de estos elementos se tiene una base sobre la cual se manejará el enfoque y el desarrollo de la solución, manteniéndola acotada y realista en situaciones restrictivas; disminuyendo tiempos de desarrollo y riesgos técnicos; reduciendo problemas de escalamiento y evolución tecnológica; brindando una línea base para todos los tipos de proyectos, en especial aquellos que se pueden clasificar dentro del complejo entorno moderno.

La estructura técnica del proyecto consiste en la determinación del cúmulo de conocimientos y tecnologías que serán necesarias para el desarrollo de la solución; en base a la determinación del conjunto de conocimientos a emplear se establece el paradigma de programación a emplear, la forma de organizar la solución, patrones de diseño y desarrollo, seguridad, forma de implementación, etc. Mediante la definición de la tecnología se definen los elementos necesarios para que el Equipo sea capaz de desarrollar la solución de forma eficiente y sin impedimentos.

Definir la estructura técnica nos permitirá establecer las convenciones de datos, codificación, interfaz de usuario, experiencia de usuario, reportes estadísticos y plantillas para manuales de usuario, de documentación, y de testing.

Al determinar la Arquitectura tendremos que escoger entre diferentes niveles de granularidad en la especificidad. En un extremo del espectro tendremos una Arquitectura completamente detallista y específica, y en el otro una descripción nula de la arquitectura a seguir, ambos casos tienen ventajas y desventajas que se verán acentuadas de acuerdo al proyecto en cuál se aplique. Sin embargo, como regla general en los proyectos del complejo entorno moderno en el que estamos, podemos decir con seguridad que una arquitectura correcta es aquella que describe el negocio, la tecnología y la navegabilidad de una forma que es apenas suficiente para las necesidades de la solución.

Una vez planteada la arquitectura es necesario documentarla o modelarla, hay numerosas herramientas y técnicas de modelado, como por ejemplo: diagramas de actividad con responsabilidades, diagramas de flujo, etc. La elección de los mismos es responsabilidad del Arquitecto.

Análisis de riesgos y definición del plan de contingencias

Es de importancia crítica el realizar un Análisis de los riesgos a los que puede estar sometido el proyecto.

Este análisis se realiza principalmente por el Equipo mediante sesiones de brainstorming y tienen como objetivo obtener una lista de los potenciales riesgos en los que puede incurrir el proyecto, no hay restricciones en cuanto a la naturaleza de los riesgos, pueden ser referidos al Cliente, a los requisitos no funcionales, a personas, al hardware, al software, etc. En caso de que se esté tratando con un tema nuevo para el Equipo es necesario obtener opiniones de expertos, la contribución de ellos facilitará la elaboración de la lista y reducirá la probabilidad de dejar afuera riesgos de gran impacto.

El análisis se ve reflejado en una matriz donde se identifica el riesgo, describiéndolo brevemente y estableciendo una valoración en cuanto a la probabilidad de ocurrencia y el impacto que puede causar en el proyecto, la multiplicación de estos valores nos dará como resultado la magnitud del riesgo. Con esta información se elaborará un plan de contingencia para los riesgos de más alta magnitud y probabilidad de ocurrencia, este plan indicará la forma en la cual se debe actuar para prevenir la ocurrencia de los mismos y/o disminuir el impacto que pueda llegar a tener en la organización.

A medida que avance el proyecto algunos riesgos se extinguirán y surgirán otros, por lo tanto, es necesario que esta lista se mantenga actualizada.

Por ejemplo:

Código	Riesgo	Impacto	Probabilidad de ocurrencia	Magnitud del riesgo
RRHH1	Perder desarrollador	un 7	20%	1.4
D1	Perder repositorio de Git	9	1%	9
C1	Disminución de la calidad de la solución	8	0.1%	0.8

Plan de entregas

El plan de entregas contempla la duración del proyecto y las iteraciones. Al tratar tiempos es extremadamente importante la participación del Equipo en la elaboración del plan, si participa únicamente el EP, o el LP obtendremos un plan que es irrealista en cuando a las posibilidades del Equipo, sin importar cuán bien estos lo conozcan. Por esto es importante obtener un plan balanceado, que satisfaga las necesidades del Cliente y represente las del Equipo y de la Solución.

En la elaboración del Plan de Entregas tendremos que determinar el nivel de granularidad y escala temporal del mismo. Algunos clientes exigirán planes detallados, en estos casos se suele realizar un WBS y un diagrama de Gantt representando las tareas a realizar, su duración y fecha de entrega, sin embargo, esto no es recomendable. El nivel de granularidad que este Marco de Trabajo recomienda es uno intermedio, enunciando la duración de cada fase (Definición, Diseño, Desarrollo, e Implementación); la fecha estimada de entregables claves; la duración de cada iteración y forma de implementación. Equipos con experiencia en proyectos similares tendrán la ventaja de poder utilizar el registro de valores de Velocidad para mejorar las estimaciones.

Este plan es evolutivo ya que a medida que avance el desarrollo será necesario especificar responsabilidades y tiempos para las Tareas que se desprenden de cada uno de los Ítems de trabajo.

La forma de implementación de entregables depende en gran medida de las necesidades del Cliente, algunos requerirán que se implemente un ítem o tipo de ítem apenas esté Hecho y a otros le será conveniente una implementación en bache del incremento de la iteración.

En los casos en que no sea necesaria una implementación inmediata, es conveniente realizarla al final cada iteración, así se reducirán tiempos de planificación. Si la iteración termina un viernes es altamente desaconsejable que se haga al último momento del último día, ya que en caso de surgir problemas probablemente no contemos con el tiempo suficiente para resolverlos.

Muy relacionado a esto es el tiempo que tomará cada iteración, los tiempos de iteración son fijos una vez que se definen en esta etapa para no introducir otra variable más al plan, mantener al equipo acostumbrado a una escala temporal y no introducir ruido innecesario a las estadísticas. En la gran mayoría de los proyectos del complejo entorno moderno una duración de 2 semanas será la correcta para un equipo de menos de 10 personas, con este tiempo la tasa planificación/desarrollo e implementación/desarrollo será muy buena en comparación con una iteración de menor duración. Algunos proyectos, por la complejidad de sus ítems pueden requerir iteraciones de mayor duración, estos casos implican mayor cantidad de trabajo al LP al tener que realizar un mayor esfuerzo para mantener al Equipo enfocado y obtener trazabilidad del trabajo. En lo que respecta al día de comienzo y fin, se recomienda que la fecha de inicio y fin sea a mitad de semana, esto es por los beneficios que tiene contar con personal después de una

implementación, si algo sale mal en una implementación de día viernes será muy difícil obtener una respuesta rápida.

Es aquí donde tendremos que dejar explícito el primer valor de WiP (Work in Progress) a tomar por el Equipo, el WiP nos indicará la cantidad máxima permitida de ítems que se desarrollarán en simultáneo, para no obtener cuellos de botella. Por lo general el valor del WiP no cambiará, pero en caso de ser necesario se determinará un nuevo valor en la reunión de Planificación.

Tenemos que recordar (nuevamente) que nos encontramos en las primeras etapas del Proyecto, tenemos poca información y elementos realizados, el producto que salga de esta fase deberá contemplar lineativas generales que servirán de guía a la hora de evolucionar la Lista. Como veremos en la etapa de Diseño, al realizar la Arquitectura de forma más detallada, saldrán a la luz los ítems que representan los pilares de la Solución, en algunos casos no serán los ítems que proporcionen mayor valor al Cliente, pero el desarrollo temprano de los mismos será crítico para el éxito del sistema. Es por esto que hay que estar preparado para negociar y hacer entender al Cliente que, a la hora de priorizar y planificar el trabajo se tienen en cuenta varios aspectos, algunos de los cuales no representan la mayor ganancia para ellos.

Estimación de costo y tiempo

Al aún no contar con una Arquitectura para el sistema no es recomendable realizar estimaciones, lo recomendable es hacerle entender este riesgo al Cliente; sin embargo, en algunos casos no se podrá evitar y por esto es que tendremos que tomar medidas para disminuir la incertidumbre. En estos casos procederemos a estimar el costo, tiempo y Valor de Dificultad en base a la Lista del proyecto.

En este marco de trabajo describiremos 3 formas de estimación, las mismas serán capaces de abarcar casi todo el espectro de proyectos con el que se enfrente un equipo de desarrollo tradicional.

Estimación por tiempo

La estimación por tiempo consiste en asignarle a cada ítem la cantidad de tiempo que tomará su desarrollo, para esto el Equipo debatirá sobre el tiempo medio estimado que tomaría desarrollar cada ítem, si hay desacuerdos sustanciales sobre valores será necesario que estas personas expongan sus argumentos y se llegue a un acuerdo.

En proyectos donde el equipo no esté familiarizado con alguno de los temas a tratar se puede realizar una estimación por tres valores utilizando el método PERT, esto nos permitirá eliminar en alguna medida el grado de incertidumbre con respecto a la duración del desarrollo de un ítem. En este método tenemos 3 variables:

- Tiempo más probable (TM) que tomará desarrollar el ítem.
- Tiempo pesimista (TP) que tomará desarrollar el ítem.
- Tiempo optimista (TO) que tomará desarrollar el ítem.

Estas variables se introducen en una fórmula para obtener el Tiempo esperado (TE). Las fórmulas dependen de la distribución con la que se quiera trabajar, así tendremos:

Distribución triangular

$$TE = \frac{(TM + TP + TE)}{3}$$

En otros campos se utiliza esta distribución cuando se tienen limitados elementos muestrales, lo cual la hace una candidata inmediata para situaciones donde se tiene limitada información.

Distribución beta

$$TE = \frac{(4TM + TP + TE)}{6}$$

La distribución beta es la que usa el método PERT y está definida para variables aleatorias continuas en un intervalo. El tiempo es una variable aleatoria continua y el mismo está acotado en una duración finita para el desarrollo de una actividad, esto hace que sea una candidata excelente.

Cuando se use esta última técnica de estimación es necesario tener especial cuidado ya que se triplica la cantidad de tiempo necesario, y si son muchos ítems la reunión se puede extender demasiado, en estos casos es conveniente optar por el método de Juicios de expertos.

Estimación por Valor de Dificultad

La estimación por valor de dificultad consiste en asignarle a cada ítem un valor que contemple la complejidad, el esfuerzo y la escala asociada al mismo. Estos valores siguen la secuencia de Fibonacci, pero pueden tomar valores intermedios cuando la representación tradicional no sea lo suficientemente precisa.

Antes de empezar a estimar estos valores es necesario tener algunos de referencia para que los resultados sean consistentes. Estos valores a su vez tienen una relación con el tiempo que tomará resolverla. Por ejemplo: desarrollar un ABM de productos=2 VD, 2 hs; Mostrar en Gmaps los lugares visitados por vendedores=13 VD, 8 hs; crear tablas en BD para visitas de vendedores =1 VD, 1h.

Una vez que se tienen estos valores de referencia, la estimación se realiza de la misma forma que la estimación por tiempo, es decir, el Equipo se reúne y debate sobre el VD que le asignarían a cada ítem, llegando a un consenso cuando haya discrepancias.

Juicios de expertos

La información con respecto al tiempo de desarrollo que nos pueden llegar a proporcionar los expertos será clave en proyectos con un elevado nivel de incertidumbre. La información que nos proporcionan puede variar en el nivel de granularidad, es decir, puede referirse a ítems, a súper-

ítems o bien a secciones de la Arquitectura. El LP y el Arquitecto son las personas encargadas de determinar cómo se trabajará con esta información.

Sea cual sea el método de estimación terminaremos con un valor de tiempo, este valor deberá ser multiplicado por el salario promedio por hora de las personas involucradas en el proyecto, al tratar con el salario esta tarea la realizará únicamente una persona en la dirección o gerencia.

Determinación de necesidades de capacitación

La capacitación es un proceso educacional orientado hacia los Clientes, a través de ella se transmiten habilidades y conocimientos específicos relativos a la Solución a implementar. Una capacitación efectiva será aquella que logre la integración del Cliente con la solución, un aumento de eficiencia en su puesto laboral, y una mejora en la motivación del Cliente con respecto a su puesto laboral.

Para lograr una capacitación efectiva habrá que hacer un relevamiento de la cultura informática de la organización del Cliente. Cuando lleguemos a esta etapa vamos a tener una noción de la Arquitectura y tecnología de la solución, esta información será clave al hacer el relevamiento de la cultura informática de la organización del Cliente, ya que nos permitirá determinar el nivel de abstracción sobre grupos o individuos, la modalidad y el alcance.

El análisis de las necesidades se hace con las técnicas normales de recolección de información, si se tiene la fortuna de contar con un Cliente cerca de la ubicación del Equipo lo recomendable es hacer entrevistas y en caso de ser necesario, complementarlas con cuestionarios y observaciones.

En algunos casos esto no será posible por lo que tendremos que limitarnos a recolectar información de forma impersonal. El análisis resultante nos permitirá elaborar el plan de capacitación una vez que el producto o un entregable esté listo para implementar.

La definición de este documento es realizada por el EP, LP y el Arquitecto.

Diseño

“El aspecto más importante del desarrollo de software es ser claro sobre lo que se trata de construir” – Bjarne Stroustrup

En la fase de Diseño se crean y actualizan modelos y planos que serán usados en las fases de Desarrollo e Implementación. Decimos que se crean modelos y planos porque se toman como entradas el feedback del Cliente e información proveniente de la fase de Definición para crear documentación que permita al Equipo resolver necesidades de requerimientos funcionales y no funcionales planteados por el Cliente y requeridos por la solución.

Cuando decimos que se actualizan modelos y planos es porque este Marco de trabajo crea un sistema de lazo cerrado con sus iteraciones. Los elementos de arquitectura, Plan de Entregas,

Matriz de Riesgos y Plan de Contingencias creados en la fase de Definición necesitarán actualizaciones a medida que el proyecto evolucione.

Entender que es casi imposible hacer todo bien al comienzo del proyecto nos lleva a buscar formas de minimizar la cantidad de tiempo y dinero invertido "al vicio". No tenemos que capturar cada detalle en los modelos que hagamos, ni hacer uno al comienzo que comprenda al proyecto de pies a cabeza; simplemente tenemos que realizar uno que sea apenas suficiente para el momento en que se lo necesite, posteriormente se lo completará, actualizará o descartará si es necesario.

Elaboración de Modelos y Arquitectura

No es el propósito de esta tesis crear nuevas herramientas y artefactos de modelación, ni explicar todas las existentes **porque no sería una lista exhaustiva**. Sin embargo, para propósitos de enunciar correctamente los conceptos necesarios se utilizarán algunas técnicas de modelado que se aplican de forma óptima a proyectos medianos.

Cuando elaboremos los modelos es necesario tener presente que es muy poco probable que el modelaje sea correcto si se lo hace al comienzo con el objetivo de abarcar todo el sistema. Con esto en mente y sumado al hecho de que trabajamos con iteraciones, hay que trabajar en base al concepto de simplicidad, simplicidad en tipos de modelos, simplicidad en cantidad de modelos y simplicidad en la elaboración de modelos; si fallamos en esto corremos el riesgo de pasar una gran parte de la iteración creando y actualizando modelos.

Para no caer en iteraciones que se centren en modelos es sumamente importante guiarse por el concepto de solamente actualizar cuando sea absolutamente necesario, es decir, cuando el costo de tener un modelo desactualizado es superior al de actualizarlo, con este concepto inevitablemente algunos modelos quedarán desactualizados, porque ya cumplieron su trabajo y actualizarlos no se puede justificar, en estos casos hay que descartarlos y no caer en la tentación de guardarlos, hay dos razones para esto, en primer lugar porque ya no sirven, pero más importante porque alguien podría guiarse por ellos para tomar una decisión.

Cuando se transita por primera vez esta etapa todavía no se contaba con una Lista de ítems basada en información necesaria por la solución, solamente estaba basada en las necesidades del Cliente, es por eso que una vez definida la Arquitectura principal podremos producir una Lista de ítems que realmente contemple las necesidades de todos los sectores involucrados. Para iteraciones posteriores la elaboración de la misma tendrá un lugar específico en el tiempo, y se realizará en la reunión de Planificación.

Tomando como entradas los requerimientos del Cliente y la Arquitectura se realizan 3 tipos de modelos según requiera el proyecto.

Modelado tecnológico

El modelado de la tecnología contempla el hardware y software involucrado en la solución, cuál es su rol en la misma y la relación entre ellos. En la mayoría de los proyectos pequeños (de corta duración y de baja complejidad), el modelo tecnológico proporcionado por la Definición de la Arquitectura es suficiente, sin embargo, este es un paso necesario para proyectos donde su evolución demande una revisión o modificación de los mismos.

Hay tres herramientas de modelado tecnológico principal, el Modelado de Red, el Modelado de Arquitectura y el Modelado de Negocio.

Modelado de red

El modelado de red se usa para representar tanto la infraestructura técnica existente como la propuesta, representa el hardware y sus conexiones. Con frecuencia este modelo no requiere actualización.

El modelado de red es realizado por el Arquitecto en forma conjunta con algún Desarrollador de ser necesario. En el caso de que se requiera modelar una infraestructura existente será de vital importancia la comunicación con el Equipo técnico de los Clientes (si es que lo hubiese), en dicho caso el LP también jugará un papel importante para facilitar y establecer medios de comunicación.

En este tipo de modelaje tendremos más chances de reutilizar que en otros, esto para mantenernos fieles a la máxima de que hay que reutilizar siempre que sea posible. Estas chances se verán incrementadas si trabajamos con Clientes que tengan un departamento de IT; si tenemos tal suerte tendremos que librarnos de la mentalidad de que si no fue hecho por nosotros no es útil.

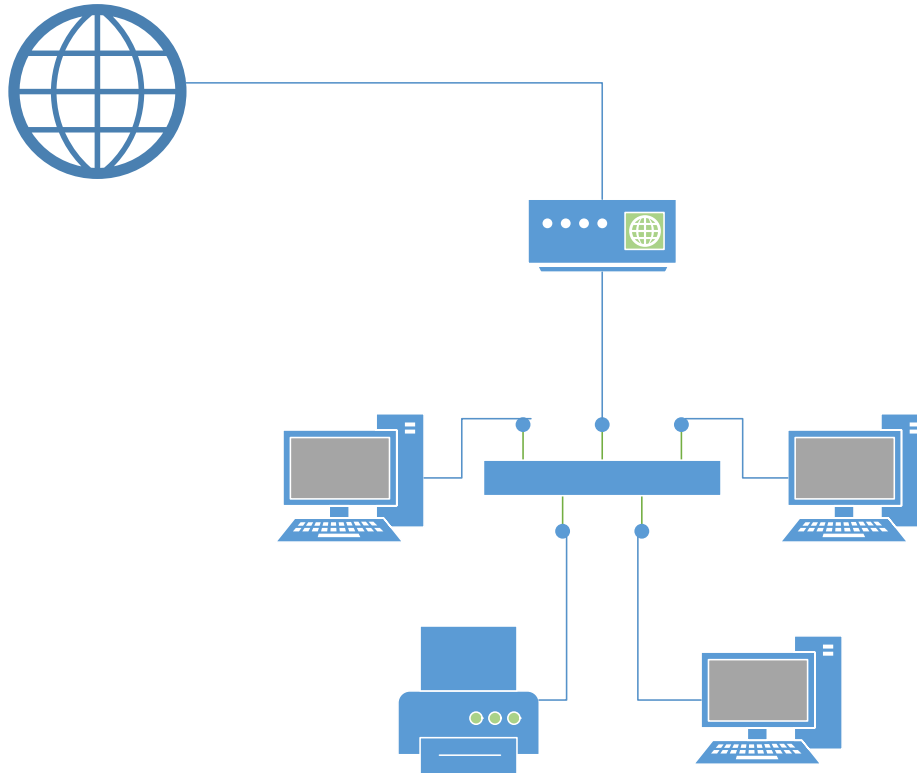


Ilustración 13 - Diagrama de red: Red estrella

Modelado de la arquitectura

Los modelos de la arquitectura representan la organización de los componentes del proyecto y nos dan una idea de cómo funciona la solución, de la misma forma que un mapa de las calles de Salta indica cómo está organizada la ciudad.

Bosquejo

El modelo de arquitectura más común es un diagrama de alto nivel de los componentes principales de hardware, software, datos y la comunicación entre estos, el objetivo es que nos dé una muy buena idea de cómo funciona la solución sin entrar en muchos detalles; este tipo de diagrama se puede considerar como una instancia del Diagrama de Implementación (UML) ya que toma algunas notaciones del mismo.

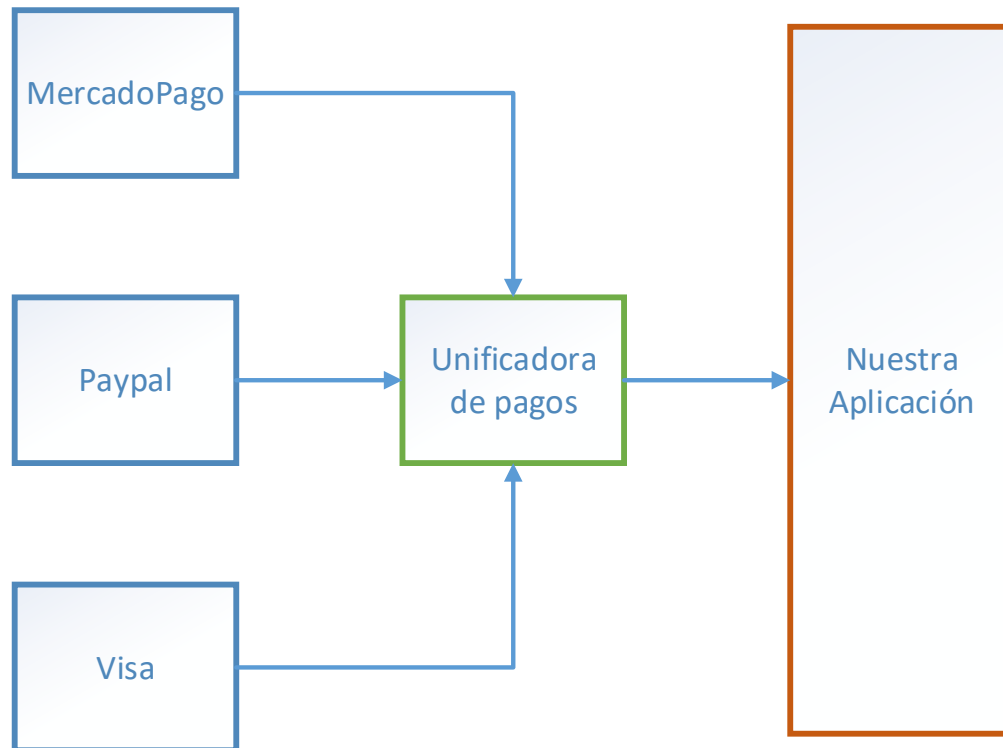


Ilustración 14 - Bosquejo

Implementar uno o más de estos diagramas será suficiente para casi todos los casos que nos encontremos, sin embargo, en proyectos en donde se justifiquen diagramas más complejos y específicos, las 4+1 Vistas de Krutchen resultan ideales.

4+1 Vistas

El modelo de 4+1 Vistas fue creado para *describir la arquitectura de sistemas de software, basándose en el uso de vistas múltiples y concurrentes* ¹⁰. Al usar múltiples vistas vamos a poder afrontar la necesidad de modelar toda la arquitectura de la solución desde diferentes aspectos, cada uno de estos acercamientos representará una de las 5 vistas planteadas por este modelo.

Vista Lógica: Modela los requerimientos funcionales de la solución.

Vista de Procesos: Modela los requerimientos no funcionales de la solución.

Vista de Implementación: Representa la organización de los módulos de software.

Vista Física: Modela cómo será implementado el sistema, a menudo incluye sub-vistas que involucran a entornos de desarrollo, testing y producción.

¹⁰Kruchten, Philippe (1995, Noviembre). Architectural Blueprints — The “4+1” View Model of Software Architecture

Vista de Escenarios: La demostración de que los cuatro elementos anteriores funcionan correctamente se realiza mediante Escenarios (casos de uso), esta vista se crea para un subconjunto de la arquitectura.

Diagrama de implementación

El diagrama de implementación es una herramienta de UML que modela la distribución de elementos de software mediante nodos y cómo se da la comunicación entre ellos.

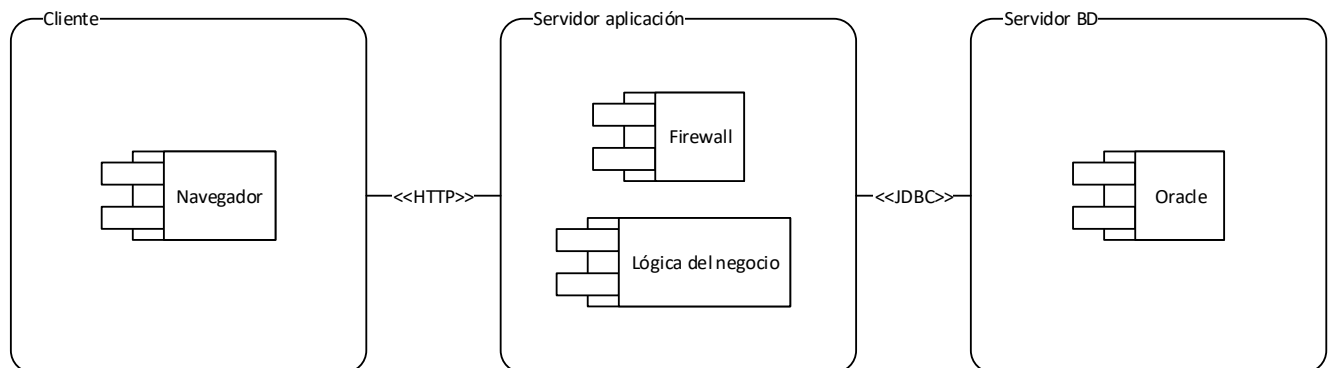


Ilustración 15 - Diagrama de implementación

Dentro del modelado de Arquitectura tenemos el modelado de datos, y junto con los bosquejos forman los dos tipos de diagramas de Arquitectura más comunes, esto no es por azar. El modelado de datos consiste en plantear los datos que va a necesitar el proyecto y cuál es la relación entre ellos, el realizar este tipo de modelado de forma temprana y con una visión de escalabilidad nos facilitará en gran medida el resto del desarrollo, viéndose acentuada su ayuda en entornos que utilicen bases de datos relacionales y programación orientada a objetos.

La herramienta más común de modelado de datos es el Diagrama Entidad-Relación, que es una representación estructurada de las entidades y sus relaciones en un sistema informático. Un DER nos servirá de forma directa al implementar la base de datos relacional, a diferencia de otros modelos, el DER general es de fácil mantenimiento ya que una vez implementada la Base de Datos, su actualización es automática, en todos los SGBD.

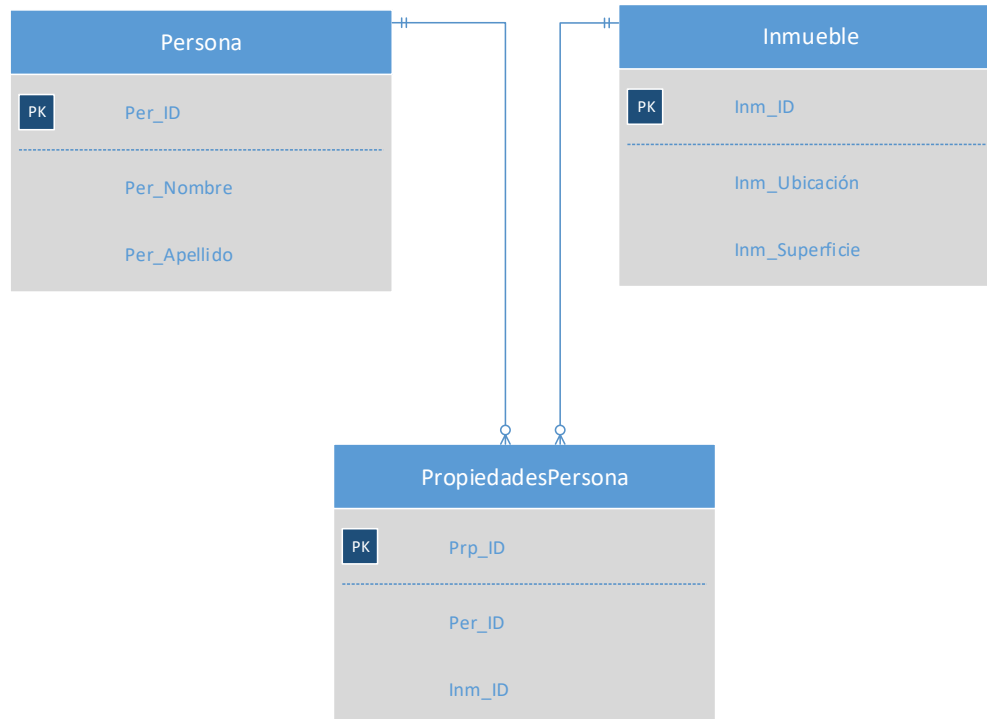


Ilustración 16-Diagrama Entidad-Relación

Modelado del negocio

Se modela el entorno de la organización en la cual operará el sistema informático para explorar y entender mejor el ambiente. Técnicamente los requisitos que recopilamos en la fase de Definición cuentan como una herramienta de modelado, así que de alguna forma el modelado del negocio comienza desde el primer contacto con el cliente, de hecho, la lista de requisitos será la herramienta de modelado del negocio más usada durante el Ciclo de Vida del Proyecto.

Para poder explorar y entender el ambiente organizacional en el cuál se desempeñará el Sistema tenemos que entender de que el mismo será operado por personas, el Sistema tiene que ayudar a cumplir el trabajo de estas personas, así que tendremos que aprender qué es lo que estas personas hacen actualmente y cómo se modificará este comportamiento con el sistema en funcionamiento.

El primer paso para entender el ambiente es obtener una estructura de las jerarquías, roles y relaciones presentes en la organización, por ello es que obtener un **Organigrama** será esencial y crítico ya que nos despejará de muchas inquietudes a la hora de entender procesos o elaborar otros modelos; el Organigrama es la base, pero necesitamos complementarlo con otras herramientas que nos permitan entender cómo funciona la organización, por esto tendremos que elaborar Diagramas de caso de uso y Diagramas de actividad con responsabilidad.

Diagrama de casos de uso

Los diagramas de caso de uso nos van a servir para modelar las interacciones que existen entre un actor y el Sistema. La naturaleza misma de este tipo de diagramas hace que sea punto de

partida para la definición de requerimientos funcionales y no funcionales del sistema. Cuando se requiera mayor detalle en la definición de un requerimiento se opta por un caso de uso con especificación, en donde se detalla el paso a paso, sin embargo, como se explicará más adelante, esto sólo será necesario en muy pocas ocasiones.

ID Caso de uso:1. Loguearse	
Actor principal: Usuario	Actor secundario:
Pre-Condiciones: Usuarios disponibles; datos de usuarios	Post-Condiciones: Usuario logueado
Escenario principal de éxito	
1. Ingresar usuario y contraseña	
	2. Validar usuario y contraseña
	3. Loguear al usuario
	4. Fin del caso de uso
Escenarios alternativos (Usuario o contraseña incorrecta)	
	2.1 Mostrar mensaje “Credenciales inválidas”
	2.1 Volver al paso 1

Ilustración 17 - Caso de uso con especificación

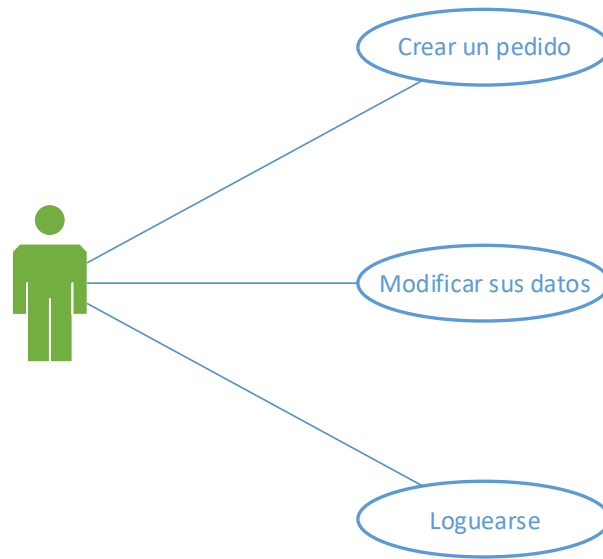


Ilustración 18- Caso de uso múltiple

Diagrama de actividad con responsabilidad

Un diagrama de actividad con responsabilidades se realiza para representar las acciones necesarias por cada rol para cumplir un objetivo, mediante este tipo de diagramas se puede modelar claramente el funcionamiento de la organización y ver cómo reacciona el sistema ante la ocurrencia de determinadas acciones o eventos.

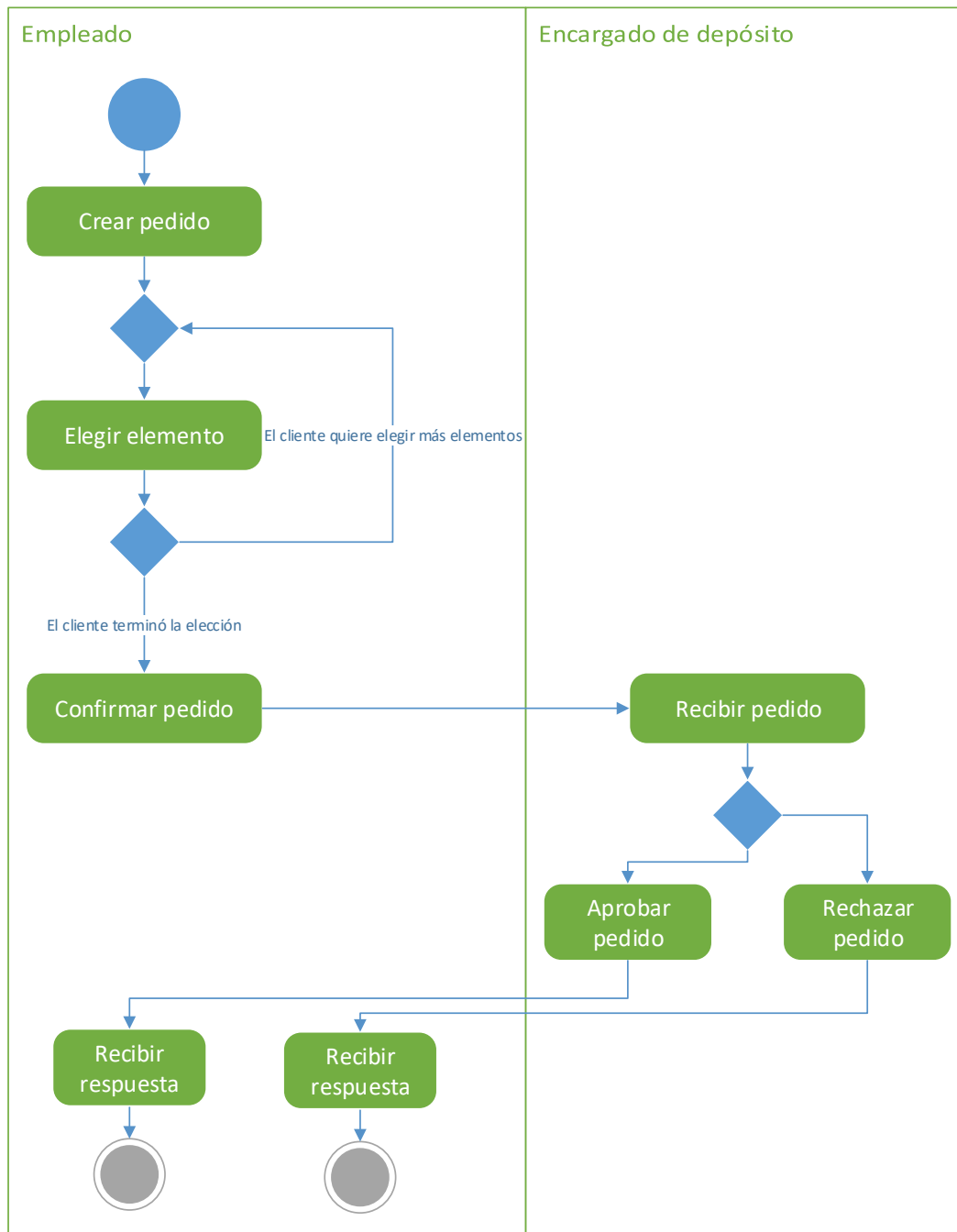


Ilustración 19 - Diagrama de actividad con responsabilidad

Análisis de la cultura informática y de integración con el sistema

Es un análisis que se realiza al Cliente para ver cuál es el estado de su capacidad de manejo de herramientas informáticas relevantes a la Sistema que se está desarrollando, a medida que se progrese en las iteraciones del Proyecto podremos medir cuál es el nivel de integración y

satisfacción con respecto al Sistema. El análisis es realizado por el EP, es muy breve e idealmente será escrito para ver el progreso en la siguiente iteración.

Actualización de documentos

Para no caer en iteraciones que se centren en modelos, es sumamente importante guiarse por la máxima de actualizar cuando sea absolutamente necesario, es decir, cuando el costo de tener un modelo desactualizado es superior al de actualizarlo, bajo este concepto, inevitablemente algunos modelos quedarán desactualizados porque ya cumplieron su trabajo y su actualización es injustificable. Aquí determinaremos qué documentación será actualizada en la siguiente fase.

En algunas ocasiones los modelos serán temporales, como por ejemplo una lista de requisitos, o un bosquejo de Arquitectura, en estos casos hay que ser fieles al propósito temporal de estos modelos y descartarlos cuando se haya terminado su periodo de utilidad. Hay varias razones para esto, en primer lugar, porque probablemente contengan información desactualizada; esto acarrea otro inconveniente aún más grave ya que alguien podría guiarse por ellos para tomar una decisión; conservar modelos temporales es altamente perjudicial para la organización de documentos del proyecto; y por último disminuye la tentación de actualizarlos.

Determinación de Ítems a documentar

Antes de hablar sobre cuales Ítems hay que documentar tenemos que ponernos de acuerdo en qué entenderemos como documentación. Hay dos tipos de documentación, la documentación que está en el código fuente y la que está en documentos como manuales de usuario, modelos, etc. Nosotros nos referiremos como documentos a esto último, es decir a cualquier elemento fuera del código fuente que proporcione información sobre un determinado tema.

La documentación es una parte inherente a todos los sistemas informáticos, la determinación de cuándo será necesario documentar estará dada por 3 razones principales.

1. Requerimiento del Cliente: El Equipo tiene que cumplir con su obligación de brindar información acerca de cuándo y cómo documentar, pero el cliente es el que tiene la última palabra en el tema, si él requiere que en el proyecto exista determinada documentación lo tendremos que hacer. Después de todo, tenemos que recordar que es su dinero el que se está invirtiendo para el desarrollo de la solución.
2. Requerimiento del Equipo: En la mayoría de los casos el mismo Equipo será quien determinará qué ítems serán documentados, salvo situaciones extraordinarias (que deberán ser revisadas por el LP) se documentará un ítem cuando
 - a. Se trate de un requerimiento ambiguo: A lo largo de un proyecto se presentarán situaciones donde la descripción tradicional usada para un Ítem no será suficiente, por lo general esto ocurrirá cuando se trate de un requerimiento complejo que no se pueda desagregar en sub-ítems.
 - b. Se trabaje con una nueva tecnología: En situaciones el equipo deberá lidiar con una tecnología nueva para realizar alguna tarea, estos casos son candidatos a ser

modelados ya que así aumentaremos las posibilidades de éxito al disminuir las variables en juego.

- c. Nuevo integrante del Equipo: Cada nuevo integrante del tendrá que pasar por un periodo de adaptación hasta que se adecúe a la forma de resolver problemas del Equipo. Tanto el nuevo integrante como el Equipo darán por sentados ciertos elementos a la hora de desarrollar y rara vez coincidirán estas expectativas. Es por esto que cuando el nuevo miembro tome una tarea, incrementarán las chances de que esta tarea requiera ser modelada; sin embargo, esto formará una tendencia con pendiente negativa hasta que termine el periodo de adaptación y la nueva persona y el Equipo se integren completamente.
3. Requerimiento por un Equipo externo: En ocasiones tendremos que trabajar con un equipo de desarrollo externo, aquí se llegará a un consenso entre los dos Equipos para determinar qué tipo y qué Ítems requerirán documentación, es importante remarcar la participación del LP y del Arquitecto, para facilitar vías de comunicación y determinación de tipo y cantidad de documentación respectivamente.

Definición de planes y directivas de capacitación

La definición de planes y directivas de capacitación se nutrirá del Análisis de la cultura informática y de integración con el sistema. Este análisis, en forma conjunta con la documentación requerida por el Cliente determinará qué información y la forma en la que será planteada para lograr una integración óptima entre el Cliente y la Solución.

El Arquitecto, LP y EP serán los encargados de determinar la metodología de exposición, planes de capacitación y organización del evento de ser necesario uno. La Capacitación involucra recursos del Cliente y nuestros, es por eso que es necesaria una comunicación fluida para coordinar de forma conjunta el tiempo y lugar.

Es ideal capacitar al final de la iteración y antes de la implementación, para que el Cliente esté preparado para recibir el incremento, la capacitación tiene que apuntar a preparar al usuario final a hacer uso óptimo del Sistema, es por esto que son convenientes exposiciones con ejemplos reales y con el sistema corriendo en el entorno de Testing (o Desarrollo en su defecto). Las restricciones determinarán si es posible realizar una capacitación a todas las personas o bien a un grupo reducido, de cualquier manera, la capacitación tiene que evacuar todas las dudas que puedan llegar a surgir.

Una forma que prueba ser eficiente la de una Capacitación presencial, demostrativa y participativa. La misma consiste en una exposición de situaciones “cuotidianas” con el Sistema por parte de un orador, los asistentes podrán participar planteando dudas sobre los temas planteados (y no sobre funcionalidades fuera de la capacitación).

Desarrollo

“Hablar es gratis, muéstrame el código” – Linus Torvalds

La fase Desarrollo concentra todos los recursos posibles hacia la meta de producir un incremento, evolucionando el Producto en cada iteración, guiándonos por la Visión, planeación y objetivos establecidos. Es por esto que partimos de los planos y modelos provistos por la fase anterior, ellos actúan como una hipótesis que ha sido planteada en base a evidencias/hechos, y que requiere ser demostrada mediante codificación y testing.

Nos encontramos en la etapa Desarrollo la mayoría de los días, por esta razón la Reunión Diaria de coordinación toma un papel protagónico para ayudar al Equipo a desempeñar tareas que permitirán demostrar que la Arquitectura, planos y modelos propuestos son correctos (o no); todos los miembros del equipo tienen que mantener una actitud proactiva y abierta a cambios, cuando se están transitando las primeras iteraciones, si el Plan de Entregas se negoció correctamente, además de entregarle una solución con el máximo valor al Cliente, se construirán los elementos clave de la Arquitectura planteada, el detectar de forma temprana errores en estos pilares salvará el Proyecto.

Codificación y testing

Todas las tareas que realizamos con anterioridad tienen un punto de confluencia en este momento, la mayoría de los artefactos han sido elaborados para que en esta etapa se facilite la labor de crear un producto de la máxima calidad posible, ya que, si no se cumple con la máxima que tiene que tener toda organización, realizar software de calidad, no podemos entregar el incremento al final de la iteración.

Un software de calidad se puede lograr únicamente con testing intensivo (a menos que el equipo esté formado por clones de Linus Torvalds)¹¹, es por eso que la programación no puede estar separada del testing, de hecho, tienen que estar lo más cerca posible.

A medida que incrementa la complejidad de la Solución y las iteraciones transitadas veremos que es necesario tener test de regresión para garantizar la calidad, la razón se da porque: a medida que los incrementos producidos se van integrando con la Solución, aumentan las chances de que algo salga mal; eventualmente se modificarán/refactorizarán secciones de código “antiguas” y es aquí donde hay mayor peligro, ya que hay mayor riesgo de dependencia.

El método de escribir código tiene que estar apuntado a maximizar la mantenibilidad y relevancia del código en el tiempo. La única forma de conseguir esto es con un desarrollo soportado por el testing y basado en la reusabilidad. Así, identificamos 2 formas de escribir código: aquella donde el testing ocurre después de la codificación, o cuando el testing ocurre “mientras” se codifica.

¹¹ Tech Talk: Linus Torvalds on GIT, 2007.

<https://www.youtube.com/watch?v=4XpnKHJAok8>. Vigente al 16/08/2016

Para elementos de complejidad y dependencia múltiple es recomendable tomar un acercamiento TDD porque ayudará mucho a producir código de calidad, ya que demandará encapsulamiento, garantizará el funcionamiento esperado, y facilitará la refactorización de código porque los errores serán detectados. Estos ítems habrán sido debidamente identificados por el Arquitecto/Equipo y contendrán una lista de criterios de aceptación. Bajo este método, primero y bajo la guía de esta Lista, se crearán las pruebas de aceptación que tendrá que pasar el ítem, posteriormente se realizará la codificación necesaria para que el software satisfaga esas condiciones.

Independientemente del método empleado, al codificar el Desarrollador realiza modelos específicos del ítem a desarrollar, al hacer esto es normal que surjan dudas, por lo tanto, la comunicación con el EP es crucial para evitar errores.

Los tests no tienen que estar limitados a determinar el cumplimiento de los requisitos funcionales, sino que también hay que testear el funcionamiento del diseño de UX, GUI, y requerimientos no funcionales. Es de larga data el conocimiento de que los testeos son más efectivos cuando no son realizados por la persona codificó la funcionalidad, aquí resulta relevante nuevamente el EP, ya que al ser la persona que mejor conoce al Cliente, resulta la óptima para testear funcionalidades siempre que sea posible, siendo indispensable su participación en aquellos ítems que representen un alto valor para el Cliente.

Cualquiera sea la forma de encarar la programación, es de importancia crítica que la persona encargada de la UX (normalmente el LP) esté involucrada de forma constante para asegurarse de que las interfaces sean eficientes y eficaces en la transmisión de información y cuenten con buena usabilidad; esta tarea se extiende a la documentación requerida por los Clientes, ya que una documentación sin adecuación contendrá datos y no información.

En lo que respecta al Equipo, el LP tiene que hacer uso de sus habilidades para crear un ambiente de transparencia, comunicación y responsabilidad, alentando la afrontación de tareas fuera del campo de experticia siempre que sea posible, promover la responsabilidad tendrá un efecto directo en su tarea de elaborar informes estadísticos sobre el desarrollo, esto es porque si se comprende que el desarrollo también involucra completar tiempos y valores asociados a los ítems y tareas que se afrontan, se contarán con los elementos para realizar estos informes, la implementación continua de esto tendrá como consecuencias una serie de datos libre de inexactitudes y de lapsos que puedan dañar la integridad de la misma. Se confía en las personas por sobre todas las cosas, pero también se presta atención a los números que se desprenden de ellas.

Estabilización del trabajo

Todo el trabajo que sea commiteado hacia el repositorio tiene que garantizar estar libre de errores, es por eso que cerca del final de la jornada laboral los miembros del Equipo tienen que asegurarse de estabilizar los ítems sobre los cuales estuvieron trabajando.

Temas desconocidos

A veces el Equipo no contará con alguna habilidad necesaria en un tema particular, el Equipo decidirá rápidamente si es conveniente conseguir ayuda fuera del Equipo o bien investigar por cuenta propia. Estas necesidades se hacen evidentes en aquellos temas que salen de la norma de los temas comúnmente enseñados en las Universidades, algunos ejemplos son Seguridad informática en el aspecto práctico, licenciamiento, y versionamiento de bases de datos.

Errores

En el día a día se trabaja para producir nuevas funcionalidades y modificar existentes, sin embargo, hay ocasiones donde saldrán a la luz errores en el sistema ya implementado, todos los casos deberán ser atendidos con rapidez, aunque algunos serán de mayor emergencia que otros.

La ocurrencia de cualquier tipo de error nos indica que hay algo que se hizo mal en alguna de las etapas, y el tener que atenderlos implica que ese tiempo no se usó para los ítems que nos comprometimos a entregar en la iteración, lo cual es un doble llamado de atención.

Ante cualquier error el equipo deberá dialogar para identificar cuál fue la razón de su ocurrencia, la aparición reiterada de errores tiene que actuar como un llamado de atención al Equipo y requerirá hacer un análisis más profundo para detectar las causas y poder elaborar un plan de acción.

Como vimos, cuando surge un error, el calendario se altera por la introducción de una variable inesperada, el LP necesitará evaluar la situación rápidamente para determinar si (además de solucionar el problema) es necesario investigar inmediatamente (o no) la fuente y las condiciones que llevaron a ocasionar ese error, esto solamente será necesario en situaciones donde haya riesgo de re-ocurrencia o aparición de nuevos por dependencia con el original. Reconociendo que cada proyecto, Equipo y errores son únicos, el LP tendrá que determinar cuándo es el momento apropiado para debatir sobre el tema; la única recomendación que se puede dar es con respecto a errores pequeños, es decir aquellos de baja complejidad, sin dependencias ni riesgo de re-ocurrencia, en estos casos el diálogo tendrá lugar en la reunión de Retrospectiva, que se realiza al final de la iteración, esto es para no “quitar” más tiempo a una agenda que ya fue ajustada.

Para el análisis de errores, la mejor forma de detectar la raíz del problema es a través de una revisión en Equipo de:

1. La descripción del Ítem. Los errores más frecuentes ocurren por ambigüedades o una interpretación incompleta o deficiente de la descripción de la funcionalidad requerida. Por esta razón es la primera línea de investigación; si el análisis del Ítem no describe la causa del error tendremos que seguir con la Arquitectura y Modelos.
2. Análisis de la Arquitectura y Modelos relacionados al Ítem. Aquí podremos determinar si se produjo por un error teórico (modelado) o práctico (código fuente/hardware), e

incluso si fue causado por unos defectos del mismo módulo o de otros que actúan como entradas.

En el caso de tratarse de ítems de baja complejidad, este análisis puede ser realizado por el mismo desarrollador, en caso contrario deberá realizarse de forma conjunta con algún otro Desarrollador; la comunicación es clave para que la persona no se “auto-condene” a una tarea de investigación en solitario, recordando que todas las decisiones que llevaron a esta situación fueron tomadas en equipo.

Documentación

La documentación consiste en elementos que son parte del Sistema y actúan como soporte pasivo de información sobre el mismo; la realización de la misma debe ser realizada únicamente cuando el costo de no tenerla supere al de su creación y actualización. En esta definición queda excluida aquella documentación creada de forma transitoria, como por ejemplo los modelos parciales que realizan los desarrolladores como soporte al codificar, esto significa que para esta documentación no se realiza el análisis de si conviene o no realizarla. Un corolario de esto nos indica que la documentación a entregar está limitada a contener información estable y concisa, que maximice el valor para el Cliente y resulte de utilidad para el mismo; esto está en concordancia con una de las características distintivas del Marco, crear lo apenas suficiente.

La experiencia nos indica que, aunque la documentación sea precisa, planteada adecuadamente, fácil de interpretar, y breve, el usuario con frecuencia no la leerá, optando por la alternativa de establecer una comunicación personal con alguna persona del Equipo. La elaboración de documentación es un proceso que demanda recursos del Equipo y del Cliente, cuando se negocian las necesidades de documentación hay que dejar en claro que la documentación es un proceso costoso, y que los resultados de la misma deben ser aprovechados al máximo. Habiendo planteado esto, por las particularidades del proyecto en algunos casos la negociación desembocará en tener mínima documentación, pero si un soporte activo.

Todo texto que vaya a ser leído por otra persona tiene que estar escrito en una forma ordenada y coherente, brindando información eficientemente, estas necesidades se ven potenciadas cuando el texto esté referido a un tema de ingeniería. No todas las personas tienen la capacidad inmediata de ser buenos escritores técnicos, es responsabilidad del LP identificar a los desarrolladores que tengan la capacidad de producir documentos técnicos de calidad para que inicialmente actúen como redactores y tutores del Equipo sobre el tema. Este Marco de trabajo promueve el aprendizaje, principalmente mediante la comunicación y el trabajo fuera del área de experticia, eventualmente todas las personas que componen al Equipo tendrán la capacidad de escribir documentación de calidad.

De arquitectura

Las documentaciones relevantes a los planos del software en esta etapa requerirán menos trabajo que aquella destinada al usuario final, esto es porque la información que la compone ya está

presente desde que se transitó la fase de Diseño, estos modelos serán la documentación o bien formarán una de la parte de la misma.

Del usuario

La documentación destinada al usuario final comprende Manuales de Usuario, Manuales de Uso, y material de capacitación, en todos esos casos la información también provendrá de la fase de Diseño, específicamente de los diagramas de Casos de Uso, Análisis de la cultura informática, y Plan de capacitación, con ellos tendremos la división en sectores/grupos, módulos de capacitación, nivel de granularidad y lenguaje a utilizar.

La documentación para el usuario puede realizarse en la misma iteración (o en la iteración inmediatamente posterior), o bien en una fecha futura. El LP y Arquitecto, en base a las negociaciones con el Cliente y particularidades del proyecto decidirán sobre cuál es el aproche correcto.

En el primer caso la documentación tiene lugar una vez que se han terminado de desarrollar los ítems de la iteración, las ventajas de esto es que la persona todavía tiene en mente toda la información que fue necesaria para el ítem desarrollado, y al crear la documentación en iteraciones el trabajo se encuentra distribuido y por lo tanto se aplican los beneficios del trabajo iterativo.

En el segundo caso la documentación se realiza una vez que el desarrollo se haya terminado y antes de la implementación final. Las ventajas de esto es que la información es mucho más estable, pero se corre el riesgo de pérdida de información al trabajar con funcionalidades que en algunos casos fueron desarrolladas hace varias iteraciones.

Técnica

En lo que respecta a la documentación del código fuente, la pauta de la organización deberá ser: Documentar solamente para esclarecer cosas que no son obvias con simplemente leer el código. Esto se resultará evidente al desarrollar y no antes; sin embargo, habrá situaciones y necesidades excepcionales que serán identificadas por el Arquitecto, o aquellas que hayan sido acordadas previamente.

En el caso de que desarrollemos una API, interfaz o algún recurso que vaya a ser usado por un desarrollador o Equipo ajeno a la organización tendremos que hacerlo en base a estándares de la industria, el Arquitecto es el encargado de guiar al equipo mediante el establecimiento de pautas. Siguiendo la misma metodología que en la situación del código fuente, evitando la verbosidad para ahorrar tiempo, tanto a nosotros como a la persona que vaya a leer la documentación.

Modificaciones

La organización toda tiene que asimilar la idea de que el cambio es posible en cualquier momento, los Clientes tienen que saber que el Equipo responde a sus necesidades, no hay mejor forma de hacerlo notar que mediante un proceso de cambios no burocrático; si ellos requieren un

cambio solo basta con pedirlo y nosotros nos encargaremos de que ese pedido pase por las etapas correspondientes.

Si no adaptamos una actitud positiva de aceptación del cambio, eventualmente produciremos funcionalidades que no le sirven a nadie, ya que las necesidades de los clientes cambian y se refinan con el tiempo, si pasadas varias iteraciones desarrollamos un ítem que fue especificado en la fase de Definición el riesgo de que esto pase incrementa ya que era el momento en que menos información se tenía; para hacer esto aún peor, se invertirán recursos en funcionalidades que nadie necesita, cuando podrían estar enfocados a aquellas que si son requeridas.

Implementación

Cuando se finalice una funcionalidad tenemos que analizar en qué entorno se va a implementar, tendremos tantas opciones como entornos haya en nuestra organización, como mínimo, cuando una funcionalidad está clasificada como Lista debe ser implementada en el entorno de Desarrollo/Testing, esto es por dos razones: realizar los últimos testings de integración general, y posteriormente, mediante la reunión de Demostración, la exposición de los avances a los Clientes, trataremos este último tema en profundidad cuando se describa la Reunión de Demostración.

Una funcionalidad no puede ser clasificada como terminada a menos que esté implementada en Producción y haya sido aceptada por el Cliente. Anteriormente vimos que diferentes ítems (o Clientes) requieren un trato diferente, habrá algunos que necesiten una implementación inmediata, al final de la iteración, o cuando se alcanza un hito. Teniendo en cuenta estas restricciones, debemos gravitar hacia implementaciones frecuentes y de baja demanda temporal.

Planificación

Una implementación no involucra simplemente desplegar el incremento en Producción, si bien esa es la tarea principal, la misma trae acarreadas otras, como la coordinación con el Cliente, capacitación al Cliente, marketing (si corresponde), instalación de hardware de ser necesario e incluso contactar con Staff especializado.

Podemos tomar a la implementación (de ahora en más nos referiremos a implementación en Producción) como una función de 3 variables que nos indicará cuándo implementar, las variables serán:

1. Requerimientos del Cliente: El modelo de negocio del Cliente es el que dictará las reglas para la implementación. Como el objetivo es maximizar el valor para el Cliente del incremento, el momento estará destinado a ser fuera de la jornada laboral del Cliente. Cuando definimos el Plan de Entregas en la fase de Definición quedó establecido el lapso para realizar implementaciones. Antes de realizar una implementación, o cuando sea necesario alterar el lapso establecido tendremos que comunicarnos con el Cliente para plantear la situación, de esta forma sabremos si está listo o no para la implementación. En

la comunicación indicaremos las lineativas y requerimientos del Plan de Capacitación con el objetivo de establecer un lugar y fecha, como así también darle tiempo para organizarse.

2. **Requerimientos de la Solución:** En ocasiones las funcionalidades incluidas en el incremento tienen necesidades especiales, cuando se haya necesitado de la ayuda de Staff para completar alguna tarea en la fase de Desarrollo y Testing las probabilidades de que necesitemos Staff para realizar la implementación incrementan linealmente. Cuando se necesite de personal especializado ajeno a la organización tendremos que avisarle con antelación para disponer de su tiempo cuando sea necesario. La necesidad de instalación de hardware pertinente a la solución requiere coordinación de recursos por parte del Equipo (y/o Staff) y del Cliente.
3. **Requerimientos del Equipo:** El realizar implementaciones previo a un fin de semana es altamente desaconsejable, las razones deberían ser evidentes. Es por eso que deben ser realizadas entre el comienzo y la mitad de la semana, en el lapso establecido por el Cliente. El Equipo deberá coordinarse para determinar quién implementará el incremento y estar preparado para posibles inconvenientes. A nadie le gusta trabajar después de hora, por esta razón, siempre que se pueda, la labor de implementación debería que ser de carácter rotativo.

Cuando se está desarrollando una solución que representa una propiedad del Equipo, las tareas de Marketing deben ser planificadas para que coincidan con el lanzamiento del incremento, las mismas incluyen, pero no están limitadas a:

- Diseñar posters.
- Crear avisos publicitarios.
- Difusión en redes sociales.
- Contactar Staff (diseñadores gráficos, administradores de publicidad de Google, administradores de redes sociales)

Capacitación

El mejor momento para realizar una capacitación es antes de la implementación del sistema, de manera que estén preparados para recibir las nuevas funcionalidades, la capacitación se realizará de acuerdo al Plan establecido previamente. Es importante recopilar la mayor cantidad de información posible con respecto a la forma en la cual el usuario final hace uso del incremento, detectando posibles falencias en la interfaz, UX e incluso en funcionalidades.

Implementación

Una vez que se hayan detectado las necesidades de implementación, y establecido cuándo realizarla, se procede a responder el cómo.

Una tarea de implementación siempre deberá seguir una secuencia de pasos para evitar confusiones y errores, el seguir una “receta” que contemple la seguridad no significa que sea una

tarea burocrática ni lenta, cualquier persona que haya pasado por una tarea de implementación sabe cuán necesario es tener un plan y estar preparado para inconvenientes.

Como vamos a tocar un sistema que está en producción, tendremos que dejarlo listo para recibir el incremento. Esto significa que, en primer lugar, tiene que dejar de recibir peticiones y terminar de procesar aquellas transacciones pendientes para evitar inconsistencia de datos. Una vez que el Sistema se encuentra en un estado consistente es necesario realizar un backup, para poder recuperar el mismo ante un error.

Para la implementación propiamente dicha primero tendremos que configurar los servidores para recibir las actualizaciones, posteriormente habrá que implementar las modificaciones a la base de datos, contemplando tanto las estructuras de datos (DML), como los datos mismos (DDL), en ocasiones una parte de la lógica del sistema será realizada por la base de datos, la misma será clasificada como DML. Cuando todos los scripts de DML y DDL hayan sido ejecutados y compilados correctamente podremos proceder con la implementación de las funcionalidades.

Una vez que la implementación se haya realizado tendremos que comprobar que funcione correctamente, para testear en Producción tendremos que conocer a la perfección qué datos están involucrados ya que corremos el riesgo de dejar la base de datos en un estado inconsistente si no sabemos sobre qué hacer un rollback. Aunque un testing a estas alturas parezca arriesgado, no es innecesario ya que es mejor descubrir un error en esta fase que descubrirlo mediante un reclamo del Cliente; de todas formas, los errores que se puedan descubrir en esta etapa seguramente habrán sido causados por una implementación defectuosa, porque el testeo de las funcionalidades se realizó de forma intensiva en la fase anterior.

Ahora tenemos un sistema estable, que funciona correctamente, lo correcto es realizar un backup, hacer “público” el sistema, y avisar al Cliente y Equipo que la implementación fue exitosa.

La “receta” anterior indica los pasos generales por los que tiene que pasar un incremento tradicional, excepciones requerirán que el Equipo dialogue sobre qué pasos tomar para asegurar una implementación sin problemas.

No tenemos que olvidar que el Cliente es a quien están destinadas todas estas tareas, es por esto que no podremos decir que una implementación fue exitosa hasta que el EP nos diga que todas las partes involucradas están satisfechas con el incremento.

Reuniones

Las reuniones asumen un papel crítico en este Marco de trabajo, son las que nos permiten planificar, coordinar, actualizar y demostrar. Son un arma de doble filo, por un lado, son extremadamente útiles y por otro, si no se implementan y manejan correctamente se convierten en una actividad frustrante, donde no se consigue ningún avance y que todos esperan que termine lo más rápido posible.

Es por esto que las reuniones tienen que ser específicas, tratar solamente temas relacionados a su objetivo y lo más cortas posible. Es altamente desaconsejable mezclar temáticas de varias reuniones en una sola, ya que el riesgo de perder el hilo incrementa con el número de variables a tratar, esto se ve potenciado aún más si las variables pertenecen a dominios distintos.

Siempre que la situación lo permita se tiene que adoptar un tono informal, no es obligación que las reuniones se realicen en la oficina, lo informal contribuye a que una reunión tenga un tono más ameno.

Definición

Las reuniones de Definición son las que le dan inicio al proyecto, y tienen lugar en la fase de Definición. Decimos reuniones, en plural, porque cuando se da inicio al proyecto con frecuencia son necesarias varias reuniones, cada una enfocada a un objetivo distinto y no siempre con el mismo público; a continuación, se detallan los tipos, funciones y características de cada una.

Definición de Necesidades

La reunión de Definición de Necesidades preferentemente es de carácter presencial, y es la primera reunión que se lleva a cabo en el ciclo de vida del proyecto, tiene lugar después de un contacto del EP con el Cliente para establecer tiempo, lugar, tema y participantes. La reunión es de una duración mediana, es decir no más de 3 horas por día, y tiene como objetivo la recolección de Requerimientos, establecer los Objetivos del Proyecto, identificar y definir las necesidades clave del Cliente.

La experiencia del EP y Arquitecto juega un papel vital para guiar la recolección hacia elementos que permitan delinear primeras versiones de Arquitectura y Visión, y no gastar tiempo en recopilar requerimientos que no sean necesarios o que estén muy a futuro en la línea temporal.

Los participantes de la reunión son el LP, EP, Arquitecto y los Clientes. Puede haber una o más reuniones de este tipo y se recomienda que tengan lugar en la primera mitad de la jornada laboral, para que en la segunda el Equipo pueda enfocarse en trabajar con el contenido producido por esta.

Definición de Visión

La reunión de Definición de Visión y Arquitectura tiene como entradas lo producido por la reunión de Definición de Necesidades. Como vimos, puede haber 1 o más de esas reuniones y por esta misma razón puede haber 1 o más reuniones de este tipo. Aquí nos enfocaremos en:

- Identificar los límites del producto.
- Crear y refinar primeras versiones de la Arquitectura de la solución, entendiendo que la misma no necesariamente tiene que representar la Arquitectura final, pero si tiene que servir como guía fidedigna para su detalle en la fase de Diseño.
- Establecer objetivos técnicos y humanos a cumplir con el Proyecto, con esto nos referimos a las metas para los miembros de nuestra organización.
- Crear, actualizar, priorizar y estimar la Lista del Proyecto.
- Elaborar Plan de entregas, maximizando el valor para el Cliente y necesidades críticas de Arquitectura.

Idealmente esta reunión tiene lugar en la segunda mitad de la jornada laboral para aprovechar el conocimiento reciente adquirido sobre las necesidades del Cliente. La misma es exclusiva para las personas de nuestra organización y, es conveniente que no participen miembros adicionales a los que estuvieron en la reunión de Definición de Necesidades, para no gastar tiempo en explicarles la situación, aquí veremos un claro ejemplo donde se cumple la Ley de Brooks¹².

Definición de Entregas

Una vez que hayan culminado las reuniones anteriores, deberíamos tener un panorama claro de qué pretende el Cliente y cuál es la solución planteada. Lo que queda es negociar cuál será el Plan de entregas inicial y el Plan de pagos a adoptar.

Al tener debidamente preparados todos los elementos con los cuales trabajar, esta reunión es de corta duración, no más de 2.5 horas, en ella participarán el EP, Arquitecto y el Cliente, con el objetivo de:

- Negociar y refinar el Plan de entregas. En la reunión de Definición de Visión se elaboró un plan de entregas tentativo, en principio el mismo tiene carácter final ya que para su elaboración se contemplaron las necesidades de la Solución y del Cliente, sin embargo, el mismo puede llegar a negociarse y cambiar siempre y cuando no ponga en riesgo la Arquitectura; dejando de lado la cronología, refinaremos el plan a través de la identificación de ventanas de implementación, que serán críticas para completar la fase de Implementación.
- Negociar y refinar el Plan de pagos. Si se realizó la estimación de la Lista del Proyecto o Lista de ítems deberíamos tener un presupuesto estimado, el flujo de dinero idealmente

¹² “Añadir más efectivos a un proyecto de software en retraso, lo retrasará más.”

seguirá el patrón de iteraciones que sigue el Proyecto, desafortunadamente este no siempre será el caso, es por esto que debemos negociar el plan que favorezca a ambas partes. Una forma de inclinar la balanza hacia un plan por iteraciones es hacerle notar al Cliente las desventajas monetarias de desarrollar funcionalidades que en un principio aparentaban ser importantes, pero la evolución de necesidades hace disminuir su valor.

Definición de riesgos

Al contar con un panorama de los elementos y condiciones del proyecto, tenemos que empezar a definir los riesgos que trae aparejado el desarrollo, como así también el plan de contingencias para los mismos. Es recomendable que este análisis involucre al Equipo en su conjunto siempre que sea posible, para abarcar la mayor cantidad de aspectos, puntos de vista y especialidades posible.

La definición se hace en una reunión de corta-media duración, con modalidad de lluvia de ideas o Brainstorming. Al comienzo del proyecto, naturalmente quedan por definir todos los riesgos, esta reunión intenta dejar el número de riesgos relevantes por definir en el mínimo aceptable. El primer elemento resultante de esta reunión es una Matriz de Riesgos, donde se describe, clasifica y pondera cada riesgo identificado; el Plan de Contingencias tiene como herramienta de entrada esta matriz, y se aplica únicamente a los elementos con mayor magnitud de riesgo, este plan se realiza mediante la misma técnica de Brainstorming, dejando en claro cómo prevenir el riesgo y/o disminuir el impacto que pueda tener en la organización. En la primera reunión, al tener numerosos elementos por definir, la habilidad como facilitador del LP será de vital importancia para mantener un clima productivo.

Una vez transitada la primera reunión de Definición de Riesgos, la Matriz resultante puede ser actualizada en cualquier momento de la iteración, por cualquier miembro del Equipo, si este es el caso, los otros miembros del Equipo también tienen que tener conocimiento de esta alteración, demoras en la identificación de riesgos pueden resultar catastróficas. Así, esta reunión puede tener un lugar en el tiempo especial, diferente a todas las otras reuniones, determinado por el tipo de riesgo y su momento de aparición, por ejemplo, si se detecta un riesgo con un alto impacto y elevada probabilidad de ocurrencia en medio de la iteración, sería imprudente esperar a que termine para registrarlo y tratarlo; como así también es poco conveniente realizar una reunión solamente para tratar un error con un impacto muy bajo y de baja probabilidad de ocurrencia, cuando un aviso en la Reunión Diaria sería suficiente.

Todos los elementos producidos hasta el momento sirven para poder tomar decisiones empresariales enfocadas en el riesgo, antes de cada comienzo "formal" de iteración, el LP, EP y Arquitecto analizarán los riesgos que implica continuar con el desarrollo del proyecto, y tomarán una decisión acerca de continuar o no.

Planificación

Una vez terminadas las reuniones de diseño se realizará una reunión de duración media (alrededor de 2 horas), donde el Equipo trabajará para cumplir los objetivos de:

- Establecer la Lista de ítems correspondiente a la iteración. Los ítems que la componen corresponden a la parte superior (los ítems con mayor prioridad) de Lista del Proyecto y por lo tanto representan aquellos que proporcionan mayor valor al Cliente y son necesarios para la Arquitectura. La cantidad de ítems que se realizarán en la iteración está limitada por la Velocidad del Equipo, en ocasiones es conveniente sobrepasar levemente este valor si el Equipo está preparado para el desafío; cuando se trabaja elementos nuevos (personas, temas, herramientas) esto puede ser beneficioso para encontrar el valor de Velocidad, previamente desconocido.
- Identificar tareas. Para aquellos ítems que no sean atómicos y de complejidad mínima, será necesario identificar las tareas técnicas necesarias para completar la funcionalidad. Una técnica efectiva de identificación es la lluvia de ideas o brainstorming.
- Asignar responsabilidades. Es conveniente que todos los miembros del Equipo tengan algo que hacer al primer día de la iteración, este será el limitante en casi todas las organizaciones; sin embargo, en aquellas en donde la libertad esté siendo abusada (esto es claramente visible por la falta de compromiso, tareas inconclusas y sobrecarga de trabajo en un individuo) el LP tomará la decisión de si es necesaria una asignación inicial de todos los ítems.
- Estimar. En Equipo se procede a la estimación, la estimación realizada de forma individual no cuenta ya que invalida estadísticas y no representa el esfuerzo real que le toma al Equipo. En la etapa de Definición se explicaron técnicas y recomendaciones de estimación.
- Una vez que contemos con una Lista de ítems completa deberemos cargarla en el Tablero y/o Sistema de gestión de proyectos.
- Determinar el valor de WiP para la iteración. El valor del Work in Progress indicará la cantidad máxima de VD o Ítems a trabajar en paralelo, el mismo estará influenciado por la Velocidad mediante la fórmula: $WIP = \text{Velocidad} / \text{Días de trabajo}$.
- Implementar plan de mejoras. El plan de mejoras será producto de la reunión de retrospectiva, está referido a cosas que se podrían haber realizado mejor en la iteración anterior, el mismo está destinado a solventar estos inconvenientes.

Diaria

La reunión diaria es muy breve, y cumple la función de actualización y coordinación del trabajo, en ella, cada Desarrollador responderá 3 preguntas:

1. ¿En qué trabajaste ayer?
2. ¿Qué vas a hacer hoy?
3. ¿Algún inconveniente?

Estas preguntas son simples, pero en su simpleza se esconde un riesgo, el de convertir la reunión en una actividad tediosa que solamente sirve como actualización de estados. El LP tiene que hacer uso de sus habilidades para motivar a los desarrolladores a ir preparados a la reunión, preparados para responder cada una de las preguntas contemplando no solamente las implicaciones del trabajo propio, sino el de sus compañeros. El estar consciente del trabajo de los demás permitirá saber si tiene algún impacto en el trabajo propio, este conocimiento es **indispensable** para responder la segunda y tercera pregunta, ya que es ahí donde ocurre la coordinación. Por ejemplo, si José trabajará sobre tests de regresión que involucran tareas de Enzo, José podría decir “Hoy voy a trabajar con los tests de regresión, los mismos involucran las tareas que hiciste Enzo, ¿quieres que veamos un rato esto?”

El mismo criterio se aplica para responder la primera pregunta, hay que contemplar el contexto para responder de forma correcta, hay que considerar que se trabaja en Equipo y sobre un mismo Sistema, todas nuestras acciones de desarrollo impactarán sobre el trabajo de otro. Informar sobre el trabajo realizado es más que decir, “ayer trabajé en X”, siempre habrá algo más que informar, esa información les servirá a nuestros compañeros de Equipo. Por ejemplo: “Ayer trabajé en la página de contacto, disminuí el tiempo de carga a un nivel razonable”; “Estuve trabajando en la página de facturación, modifiqué el comportamiento del CAE ya que la lógica que habíamos planteado era incorrecta”; “Hoy voy seguir trabajando en XYZ, tengo algunas dudas con los requerimientos, LP, podrías conseguir al EP, ayer no me pude comunicar con el”

La reunión Diaria tiene un fin tácito, generar compromiso. La falta de trabajo, o el abuso de las libertades propuestas por este Marco de trabajo saldrán a la luz cuando no haya (o haya una pobre) respuesta a la primer y segunda pregunta.

Demostración

La reunión de Demostración tiene lugar al finalizar la fase de Desarrollo. Es una reunión breve, de menos de una hora y media. La Demostración tiene dos objetivos:

- Exponer el incremento producido por la iteración al Cliente. Una buena forma de realizar esto es a través del entorno de Desarrollo/Testing, mediante un recorrido del flujo de las nuevas funcionalidades, con ejemplos concretos.
- Recibir feedback sobre el incremento, incluyendo la aprobación (o rechazo) parcial o total del mismo.

La demostración puede ser realizada por cualquier miembro del Equipo, pero la responsabilidad inicialmente es del EP. Sin embargo, es necesario que exista rotación de tareas de demostración, esto mejorará las habilidades de comunicación; la persona tomará una visión organizacional de la solución, ya que verá que realmente es usada y le sirve a alguien; y también tendrá la oportunidad de ser (aún más) reconocido por el incremento que ayudó a desarrollar.

Si la reunión resulta breve, y si la situación de los participantes lo permite, se puede realizar la reunión de Definición de Necesidades a continuación, así aprovechando que nos encontramos reunidos y disminuyendo el tiempo de planificación de una reunión adicional.

Retrospectiva

Al finalizar la iteración tenemos que analizar cómo se realizaron las cosas. La reunión de retrospectiva mira atrás en el tiempo y nos permitirá saber qué cosas se hicieron bien, qué cosas se podrían haber hecho mejor, qué cosas deberíamos dejar de hacer, qué cosas deberíamos implementar, qué cosas nos resultaron frustrantes, etc. Es aquí donde se produce la realimentación que convierte a este Marco en un sistema de lazo cerrado.

Esta reunión es presencial, no debería durar más de una hora, y tiene a todos los miembros del Equipo como participantes. Si bien se reconoce que identificar cosas positivas es de valor para saber qué hay que continuar haciendo, sacar a la luz elementos a mejorar es tan o más importante, **siempre** habrá algo que mejorar, hay que apuntar a encontrar defectos para poder mejorar el proceso. Identificar elementos “negativos” puede ser difícil en un ambiente donde no haya confianza, una forma de mejorar esta situación es reconocer que cada miembro del Equipo realizó el mejor trabajo posible.

El Equipo decidirá sobre qué elementos de la retrospectiva elaborar un plan de mejora, este plan será implementado en la próxima iteración.

Hay numerosas técnicas de retrospectiva, muchas toman forma de juegos y son fáciles de implementar. Las recomendaciones que da este Marco es variar, implementando aquellas que permitan responder los temas especificados en el segundo párrafo.

La siguiente página cuenta con un buen catálogo de técnicas de retrospectiva: <http://www.funretrospectives.com/category/retrospective/>

Análisis estadísticos

De forma conjunta a esta reunión se hace una revisión de las estadísticas de desempeño del Equipo, estas sirven de complemento a la hora de elaborar el plan de mejoras y en la toma de decisiones. Tenemos 3 opciones para la demostración de esta información:

1. Demostración grupal: Se realizará la revisión/análisis en presencia de todos los miembros del Equipo, esto tiene el beneficio de que todos están en la “misma página” sobre las fortalezas y dificultades de los demás, contando con información precisa para poder ayudar; sin embargo, esto es solamente posible cuando hay un ambiente de total confianza en el Equipo, un desempeño pobre puede causar incomodidad al ser planteado públicamente.
2. Demostración personal: El LP realizará la revisión con alguna otra persona de la organización, la misma puede ser otro miembro del Equipo o bien una persona de RR.

HH. Este tipo de revisión demanda más tiempo, y es el recomendable para Equipos que tienen cierta confianza, pero la misma no es plena.

3. No realizar demostraciones: El LP refina el plan de mejoras y actúa en base a la información recolectada pero no realiza una demostración de la misma a los miembros del Equipo, porque no hay confianza suficiente.

En los dos primeros casos la recomendación es que el análisis esté disponible a todos los miembros del Equipo.

La recomendación que dimos en la definición de Artefactos es la utilización de un Sistema de gestión de proyectos, todos proveerán de estadísticos mínimos, como tareas y puntos completados y si usamos uno enfocado a metodologías ágiles un gráfico llamado Burndown Chart, que nos indicará el desempeño del Equipo en base a las tareas y el tiempo.

Un estadístico necesario es la Velocidad [VD/Día], la unidad de medida nos indica qué es lo que mide, se calcula dividiendo la cantidad de VD completados en la iteración anterior por la cantidad de días hábiles trabajados en la misma. Es tal vez el estadístico más importante porque nos limitará la cantidad de trabajo a afrontar en la próxima iteración.

Otros estadísticos importantes son:

- Relación tiempo desarrollo/bugs. Nos indica la cantidad de tiempo utilizado para los ítems planteados inicialmente y el invertido en solucionar bugs del sistema implementado en Producción. Idealmente esta relación es infinita, lo que nos indica que no ocurrieron bugs, cuando empiece a tomar valores bajos es un indicador de testing deficiente.
- Varianza de estimación de tiempo/VD estimado e invertido. Nos indica la varianza de estimaciones de tiempo y puntos de dificultad. Se usa la fórmula de la distribución Gamma por su función de densidad de probabilidad y uso para eventos sin memoria, estas razones la hacen ideal para su uso en este caso.

$$Var = \frac{a}{b^2}$$

Dónde:

a= Tiempo o VD estimado.

b=Tiempo o VD invertido o “real”.

Idealmente la varianza será 1/a, las desviaciones en estimación se verán acentuadas y así será más fácil identificar aquellos casos en los que sea necesario tomar medidas de corrección.

- Relación tiempo desarrollo/soporte. Actúa de forma complementaria al indicador de tiempo desarrollo/bugs, nos indicará la relación entre cantidad de tiempo dedicado al soporte y al desarrollo, si bien se reconoce que el soporte es necesario y siempre estará presente, un exceso del mismo indicará fallas en la capacitación, UX, GUI, documentación o funcionalidad.

Cuando una persona nueva entra al Equipo no se conoce su forma de trabajar, por lo tanto, las estimaciones que se puedan llevar para determinar el trabajo próximo no son de gran exactitud. Por más idónea que sea, el desempeño del Equipo sufrirá un deterioro inicial causado por el periodo de adaptación, para navegar correctamente por el mismo tendremos que ayudarnos por los números, es por esto que se ideó el siguiente método de estimación:

Ajuste comparativo y ponderado: Se realiza mediante una clasificación de desarrolladores en niveles: Senior, Semi-Senior, Junior y Principiante, cada uno de estos niveles tiene asociado el coeficiente 1, 1.3, 1.5 y 1.8 respectivamente (los mismos se toman de la serie de Fibonacci). El LP, con la ayuda de personal de RR. HH. determinará a qué clasificación corresponde el nuevo integrante. De forma posterior a la asignación de responsabilidades se realiza el ajuste de horas estimadas por el coeficiente correspondiente. Este valor es solamente visible por el LP y nos indicará la eficiencia del proceso de adaptación y el nivel progreso del individuo.

Semana de innovación

La Semana de Innovación es una semana que se toma el Equipo para trabajar en proyectos fuera de los relacionados a la organización. Este evento ocurre como mínimo una vez al año y tiene una duración de 7 días.

El día hábil previo al inicio de la Semana de Innovación se realiza una exposición donde se muestran los temas elegidos y grupos formados, los cuales pueden estar formados por 1 o más personas, los temas pueden ser de cualquier tipo, aunque se reconoce que por lo general estarán relacionados a la tecnología. La única regla del evento es que al finalizar se tiene que mostrar un prototipo funcional.

Mientras dure el evento los equipos pueden trabajar en sus proyectos en el lugar que quieran, las oficinas permanecerán abiertas por si algún grupo las quiera utilizar. La competitividad “saludable” es beneficiosa, en las primeras iteraciones el LP decidirá si es aceptable premiar al mejor proyecto, pero a medida que avance la sinergia del Equipo, se debería pensar en adoptar esta práctica.

Las ventajas que presenta un evento de estas características son muchas para nombrar, pero entre las principales podemos identificar:

- Aumento de confianza en el Equipo.
- Incremento de nivel de pertenencia con la Organización.
- Mejora en las habilidades de comunicación.
- Tomar conocimiento de diversas áreas de conocimiento.
- Entrenamiento de pensamiento lateral.
- Permite de conocer y plantear intereses.
- Incremento en capacidad de organización.

Verificación experimental

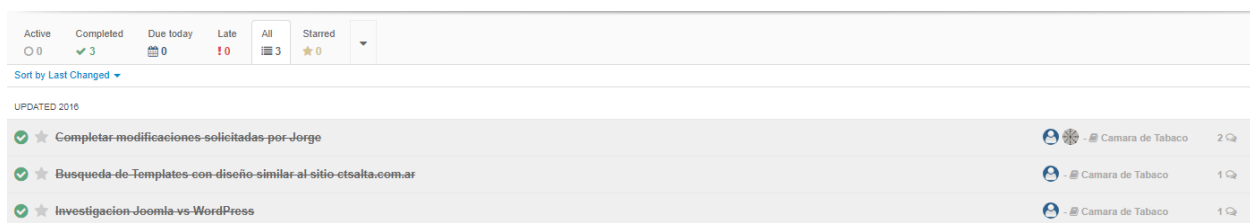
La verificación experimental de este Marco de trabajo tiene lugar en la empresa PucaráTech (anteriormente MKT Salta), esta empresa cuenta con 4 miembros de Equipo permanentes, que trabajan a jornada completa. Pucará es una empresa de desarrollo de software que tiene como objetivo desarrollar software de calidad y proponer soluciones innovadoras utilizando las últimas tecnologías disponibles en el mercado. Pucará tiene la particularidad de tener un producto propio PucaráFactura, que es un servicio de factura electrónica, pero esto no la limita a la hora de enfrentar desarrollos de software a medida.

En esta organización se empezó a trabajar con metodologías ágiles desde el mes de marzo de 2016, para el mes de agosto del mismo año se adoptó este Marco de Trabajo. La empresa trabaja en simultáneo con hasta 4 proyectos, es por esto que describiremos el funcionamiento de Pucará con el Marco de Trabajo centrándonos en un proyecto, incluyendo referencias a PucaráFactura y otros proyectos cuando sea necesario aclarar en detalle algún punto.

El propósito de esta tesis no es hacer realizar una demostración extensiva y detallada del paso a paso de cómo se realizó la implementación, ni revivir el día a día del trabajo de la empresa. Las hipótesis que plantea este trabajo serán probadas si se cumplen con los requerimientos de cada proyecto; como así también mediante el uso de informes estadísticos sobre el desempeño del Equipo de Pucará.

Situación anterior

De forma previa a la implementación de este marco de desarrollo, en PucaráTech había un concepto muy abarcador sobre actividades consideradas como contribuyentes al desarrollo, esto se ve reflejado en que el sistema de gestión de proyectos (Producteev¹³) funcionaba como una agenda de actividades generales, donde se registraban todas las tareas que podía realizar una persona en un proyecto, independientemente si estaban relacionadas directamente con el desarrollo o no.



The screenshot shows the Producteev interface with a task list. At the top, there are filters for 'Active' (0), 'Completed' (3), 'Due today' (0), 'Late' (1), 'All' (3), and 'Starred' (0). Below the filters, it says 'Sort by Last Changed'. The main section is titled 'UPDATED 2016' and contains three tasks, each with a star icon, a checkmark, and a link to a document in the 'Camara de Tabaco' folder.

Task	Folder	Count
Completar modificaciones solicitadas por Jorge	Camara de Tabaco	2
Busqueda de Templates con diseño similar al sitio ctsalta.com.ar	Camara de Tabaco	1
Investigacion Joomla vs WordPress	Camara de Tabaco	1

Ilustración 20 - Lista de tareas - CTSalta

En ese momento trabajaban 4 personas a jornada completa, Alberto, Sebastián, José y Pablo, Alberto y Sebastián eran los encargados de todas las actividades administrativas, colaborando

¹³<https://www.producteev.com> Vigente al 11/09/2017

con el desarrollo en las horas restantes, mientras que José y Pablo estaban dedicados exclusivamente al desarrollo.

Para cada proyecto los requerimientos caían en una lista. Los requisitos planteados por los clientes se tomaban de dos maneras: en reuniones referidas a tratar módulos medianos-grandes de software, y mediante planteos por email. En las reuniones se establecía una fecha estimativa de entrega para el módulo, esto determinaba la “iteración” con la que trabajaría el Equipo, para decidir la fecha de entrega, la estimación sobre las actividades era responsabilidad exclusiva de Alberto y Sebastián, esto no estaría del todo mal si se tuvieran sólidas estadísticas sobre el desempeño del Equipo, pero las mismas no estaban presentes, y la naturaleza del método de trabajo, al acarrear iteraciones de duración variable haría muy difícil medir el desempeño tanto individual como grupal.

+ Type here to create a task...

Active

Completed

Due today

Late

All

Starred

0

✓ 101

0

10

107

0

Sort by Last Changed ▾

DESC ▾

UPDATED 2016

☒	★ INCIDENTE: Cambio de estado de ficha	 DENTIS	4 
☒	★ INCIDENTE: Chequear edición de fichas para el AUDITOR	 DENTIS	3 
☐	★ INCIDENTE: Cantidad de prestaciones al momento de duplicar	 DENTIS	1 
☐	★ INCIDENTE: Descuento con presupuestos	 DENTIS	1 
☒	★ Mejorar la estética del reporte rpt_presupuestos de manera tal de que no se solapen los campos	 DENTIS	1 
☐	★ Modificación de pantalla de presupuesto	 DENTIS	3 
☒	★ Modificaciones a modulo de Presupuestos	 DENTIS	3 
☒	★ Crear reporte en excel: Listado de Fichas y Presupuestos	 DENTIS	5 
☐	★ Investigación de cambio de color de background color a checkbox	 DENTIS	
☒	★ Pago de Presupuestos – Ocultar Campos tarjeta	 DENTIS	2 
☒	★ Modificar detalle de reporte rpt_presupuesto	 DENTIS	2 
☒	★ Hacer un volcado de PROD a DESA	 DENTIS	2 
☒	★ Modificaciones a Modulo de Fichas	 DENTIS	2 
☐	★ Ejecutar proceso de facturación DENTIS	 DENTIS	
☒	★ Modificaciones a Modulo de Cta Cte	 DENTIS	1 
☒	★ Revision de codigo de insercion de Pagos	 DENTIS	2 

Ilustración 21 - Lista de tareas - Dentis

Como se puede ver en la ilustración anterior, las actividades no se separaban en grupos, las tareas pertenecientes a una “iteración” estaban establecidas en el documento resultante de la reunión de toma de requisitos. Ante un nuevo requisito, Alberto y Sebastián se encargaban de asignar responsabilidades a las mismas. Esta coordinación se realizaba mediante la única reunión pre-establecida entre miembros del Equipo.

En el día a día, cada miembro trabajaba en base su lista de tareas asignadas, en un entorno de Desarrollo, cuando se completaba una tarea se avisaba a Sebastián, quien hacía un testing final para posteriormente implementar en Producción, una vez hecha la implementación, se notificaba al cliente la implementación exitosa, detallando las pantallas y cambios involucrados.

Roles, personas y organización

Desde sus comienzos Pucará entendió que la única manera de ser competitivos es centrándose en las personas, el conocimiento y el orden.

Pucará sigue la estructura de 4+1 Roles propuesta por este Marco (pág. 28) con pequeñas modificaciones al rol de LP y EP, la toma/asignación de roles se dio de forma natural, la misma se vio influenciada por la experiencia, el conocimiento y la personalidad de cada persona. La definición de responsabilidades de los Roles es la siguiente:

Desarrollador

El desarrollador es quien realiza las tareas “duras” necesarias para completar un requerimiento, las responsabilidades y características que se desprenden de este objetivo son las siguientes:

- Realizar estimaciones.
- Identificar tareas.
- Participar de forma proactiva en reuniones.
- Colaborar con el Arquitecto en la definición y refinamiento de Arquitectura.
- Estar dispuesto a realizar tareas fuera de su especialidad.
- Mantener un ambiente de comunicación, respeto y colaboración en el Equipo.
- Buena adaptabilidad y predisposición al cambio.
- Compromiso a dar un esfuerzo perfecto.
- Capacidad de producir buen código y software de calidad.

Arquitecto

El Arquitecto es el encargado de la definición la estructura tecnológica necesaria para la solución, la definición de esta estructura establecerá los lineamientos sobre los cuales la solución será desarrollada. Para poder cumplir con esta meta el rol de Arquitecto tiene las siguientes características y responsabilidades:

- Lidera la definición de la Arquitectura requerida por la solución. La definición de la Arquitectura puede ser realizada de forma exclusiva por el Arquitecto, pero de ser necesario, Desarrolladores participarán, aquí el Arquitecto guiará al equipo y tendrá la última palabra.
- Identificación de necesidades especiales de documentación y modelado en Ítems.
- Elaboración del modelaje específico para ítems y Arquitectura.
- Estar al tanto de las mejores prácticas y tecnologías, y asegurarse de que el Equipo las emplee cuando sea conveniente.
- Conocer de forma extensiva el dominio del Proyecto
- Determinar la tecnología a usar en el Proyecto.
- Crear el plan de contingencia técnico.

- Participar en la priorización de la Lista del Proyecto y determinación de Lista de Ítems, avocando por la inclusión de elementos críticos de la Arquitectura.
- Identificación de requerimientos no funcionales requeridos por la solución.
- Educar al Equipo sobre la Arquitectura.

En la situación actual del Equipo, si una sola persona toma el rol de Arquitecto tendrá tiempo libre, independientemente de la complejidad del Proyecto, por esta razón, quien asuma el rol de Arquitecto, tomará como rol secundario el de Desarrollador. La situación planteada naturalmente cambiará cuando el Equipo cuente con mayor personal y pueda afrontar Proyectos mucho más grandes; en dicho caso el rol de Arquitecto será exclusivo.

Líder del Proyecto

El Líder del Proyecto es el líder servicial del Equipo, esto implica que el no gobierna al Equipo, sino que las necesidades del grupo guiarán sus acciones. El LP ayuda a crear y mantener un ambiente organizacional que permita al Equipo realizar sus tareas de la mejor manera posible. Las características y requerimientos del rol, para que pueda cumplir su meta son:

- Incentivar un ambiente de comunicación, cooperación, aprendizaje y creatividad en todo Pucará.
- Asistir y facilitar la resolución de todo tipo de problemas y consultas que tenga el Equipo.
- Aislar al Equipo de interrupciones externas.
- Mantener al Equipo enfocado en las metas.
- Facilitar la comunicación entre Desarrolladores, EP y Cliente.
- Facilitar las reuniones.
- Ayudar y asistir al Encargado de Producto, Arquitecto y Cliente en la definición de la Visión del proyecto.
- Determinar las directivas de UX en el proyecto.
- Elaborar planes de contingencia para los distintos niveles involucrados en el Proyecto.
- Contactar a staff especializado.
- Asegurar la correcta implementación y uso de metodologías y prácticas de desarrollo en el Equipo.
- Elaborar informes estadísticos estadísticas y, en conjunto con el LP crear el plan de mejoras para el Equipo.
- Asegurarse del correcto funcionamiento de los elementos de comunicación y colaboración.
- Guiar al Equipo en la utilización de elementos de organización y comunicación.

Encargado del Producto

El rol de Encargado del Producto tiene el objetivo de representar las necesidades del Cliente dentro de la Organización, esto implica:

- Establecer y mantener canales de comunicación con el Cliente.
- Representar las necesidades del Cliente en el Equipo, esto requerirá una comunicación constante y de óptima calidad.
- Representar las necesidades del Equipo al Cliente.
- Encontrar rápidamente respuestas tiempo para consultas tanto del Cliente como del Equipo.
- Participar en la priorización de la Lista del Proyecto y determinación de Lista de Ítems, avocando por la maximización del valor para el Cliente.
- Participar activamente en Testings de Aceptación (validaciones finales).
- Explicar el dominio del Cliente al Equipo.
- Negociar el presupuesto, alcance y tiempo del Proyecto con el Cliente.

Cliente

El rol Cliente representa a los miembros de la organización destinataria de la solución, los Clientes pueden ser Usuarios finales, Administradores, Supervisores, etc. La disposición para la comunicación y cooperación es una característica necesaria y requerida en un Cliente.

Personas

José Dousdebés

El perfil profesional resulta en una preferencia para trabajos de investigación, y optimización de procesos. Estas características y preferencias fueron reconocidas rápidamente y resultaron cruciales para la definición del rol y responsabilidades a ocupar. José ocupa el rol de Desarrollador y además asume las siguientes responsabilidades, que típicamente estarían a cargo del LP:

- Asegurar la correcta implementación y uso de metodologías y prácticas de desarrollo en el Equipo.
- Elaborar informes estadísticos estadísticas y, en conjunto con el LP crear el plan de mejoras para el Equipo.
- Asegurarse del correcto funcionamiento de los elementos de comunicación y colaboración.
- Guiar al Equipo en la utilización de elementos de organización y comunicación.

Alberto Dousdebés

Líder de Proyecto, Desarrollador y Arquitecto en Pucará. Al contar con amplia experiencia en el desarrollo de software y manejo de personas, el rol de LP le resulta nativo; esto sumado a su especialización en Ingeniería del Software, hacen que las responsabilidades del Arquitecto sean ideales. Por las características de los Proyectos sobre los cuales trabaja Pucará, asume el rol de Desarrollador cuando termina de atender sus responsabilidades de LP y Arquitecto.

Juan Sebastián Ibáñez

Encargado del Producto y Desarrollador. Es la persona encargada de representar a los Clientes, el desempeño en este rol EP se ve facilitado y complementado por su MBA. Al contar con una especialización en Bases de Datos, actúa como Desarrollador, estando su tarea enfocada en la definición y mantenimiento de estructuras de datos, participación en la definición de Arquitectura y, es el encargado de dar el último vistazo a el DDL y DML aplicado en la BD.

Pablo Alarcón: Desarrollador. Es una persona con gran entusiasmo y muchas ganas de aprender, cuenta con experiencia en el desarrollo de sistemas de gestión de stock y páginas web personales, es por esto que ocupa el rol de Desarrollador.

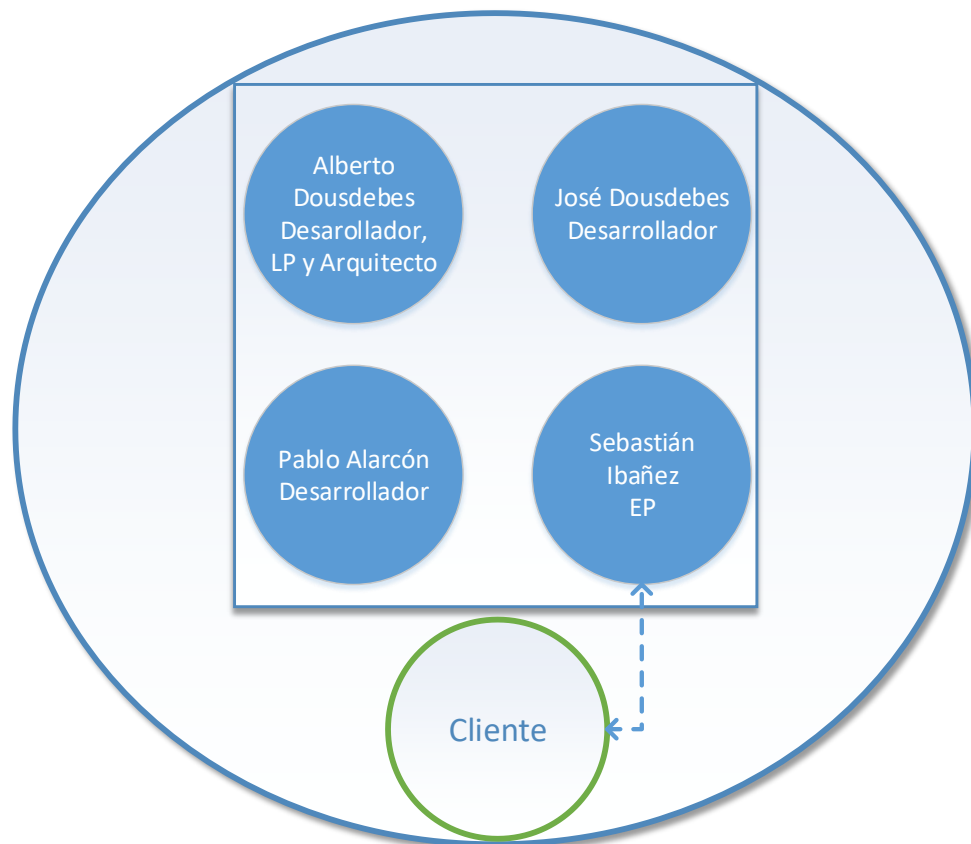


Ilustración 22 - Equipo PucaráTech



Ilustración 23 - Equipo Pucará. De izquierda a derecha, Alberto, José, Pablo y Sebastián

El Equipo está localizado bajo un mismo techo, lo cual contribuye a una mejor comunicación y calidad de información. El grupo tiene experiencia en el desarrollo de aplicaciones web empresariales, las herramientas de mayor uso son Java EE y Oracle.

Pucará tiene como objetivo la creación de soluciones innovadoras mediante software de calidad. Para lograr este objetivo, todos los miembros concuerdan en que las personas representan el comienzo y fin de cualquier proyecto, es por esta razón que se hace énfasis en lograr un ambiente agradable, que motive la comunicación, el aprendizaje y la escritura de código de calidad, todos los miembros contribuyen diariamente a hacer esto una realidad.

Artefactos y Elementos

Los artefactos usados son los que recomienda el Marco propuesto, la implementación de los mismos se realiza principalmente en el Sistema de Gestión de Proyectos llamado *Taiga*¹⁴, se lo eligió por su:

- Impecable UX y UI
- Capacidad y facilidad de customización, elemento crítico ya que el Marco de Trabajo no se podrá adaptar completamente a un Sistema Gestor exclusivo de Scrum o Kanban
- Open Source, lo que significa poder montarlo en servidores propios.
- Exportación e importación de proyectos, que resulta útil si en algún momento hay que migrar de servidores.

¹⁴ www.taiga.io

- Exportación de datos. La posibilidad de exportación automática de datos resulta extremadamente ventajosa al momento de elaborar informes estadísticos, ninguna otra herramienta del mercado brinda esta posibilidad.

Todos los artefactos se cargan en Taiga, sin embargo, el Equipo mantiene en un tablero físico (de corcho) la Lista de ítems de todos los proyectos, esto brinda visibilidad instantánea sobre el estado actual y nivel de progreso.

A continuación, se muestra el Panel de Control de Taiga para el proyecto Factura Electrónica, se recuadra en Azul la Lista del Proyecto y en verde la Lista de ítems.

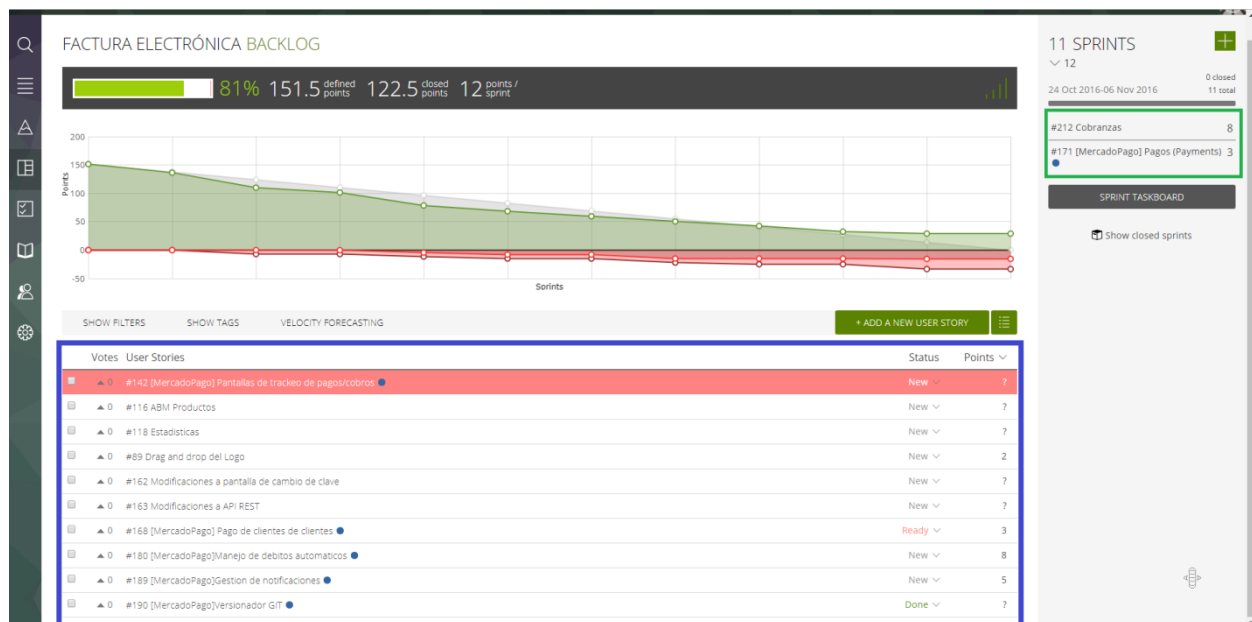


Ilustración 24 - Panel de control: Lista del Proyecto

Ítems y tareas

Las Listas de Proyecto y de iteración (Lista de ítems) están compuestas por ítems, los mismos tendrán las siguientes características o campos:

- Valor de Dificultad, que en la mayoría de los casos estará relacionado con el tiempo requerido.
- Tiempo estimado para que el ítem se considere Hecho. Solamente se indica esto cuando se carga en Taiga.
- Tiempo tomado o invertido para completar el ítem. Solamente se indica esto cuando se carga en Taiga.
- Título o nombre.
- Descripción de lo requerido.

- Criterios de aceptación, en caso de ser necesario.
- Estado. Representa el Estado actual del ítem.
- Tareas relacionadas, aquellos ítems complejos llevan adjunta tareas técnicas.
- Documentos adjuntos. Solamente será posible en el caso de cargar el ítem en Taiga.

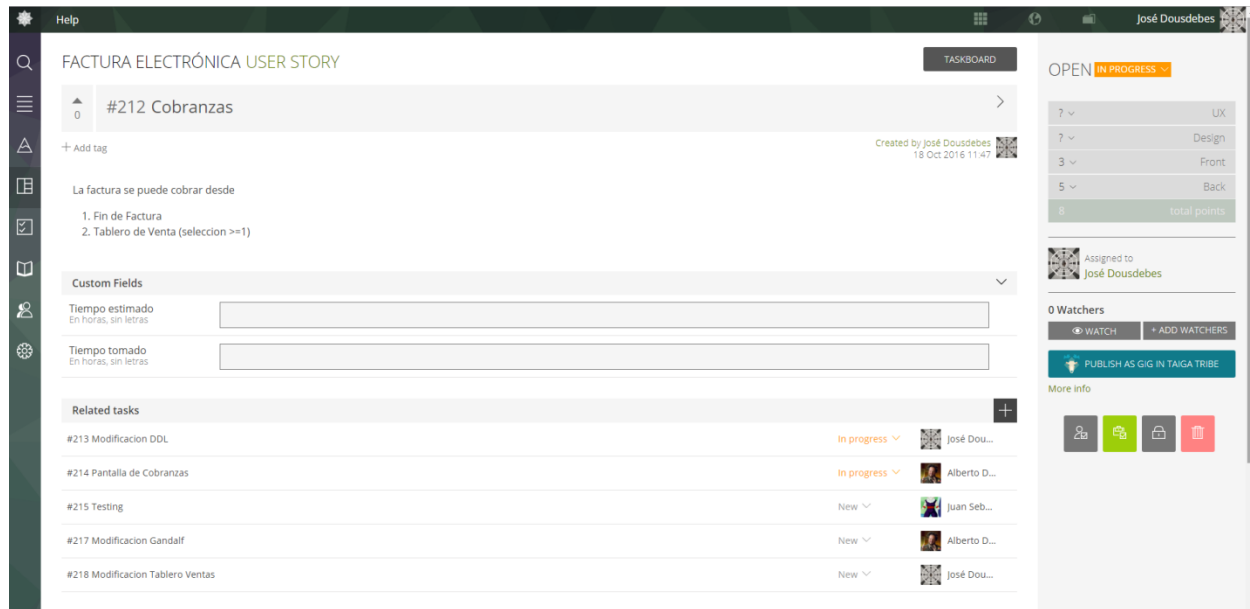


Ilustración 25 - Ítem en Taiga

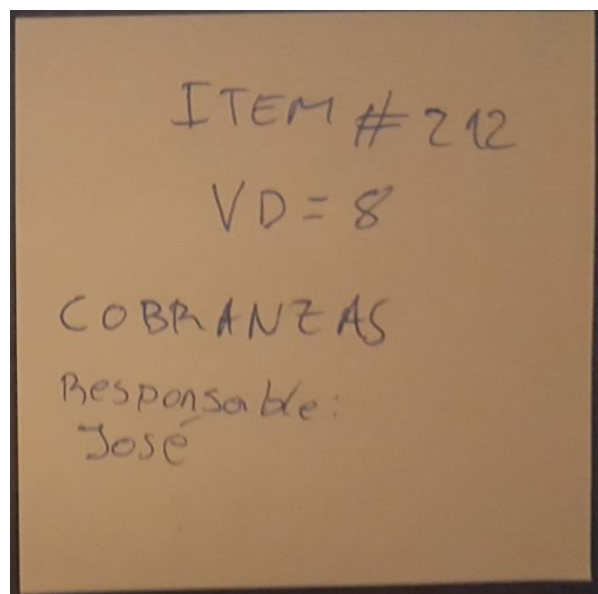


Ilustración 26 - Ítem en papel

El Equipo determinó que aquellos ítems que resulten complejos contarán con tareas relacionadas, el Tiempo Estimado e Invertido para estos tipos de ítems es la sumatoria de los Tiempos de las tareas que lo componen. En la mayoría de los casos las tareas serán definidas en la reunión de

planificación, sin embargo, esto no inhibe al desarrollador de incluir tareas una vez que se comience a trabajar sobre el mismo.

Las tareas que se desprenden de un ítem tienen un título o nombre, estado y en el caso de que se carguen en Taiga, tiempo Estimado e Invertido.

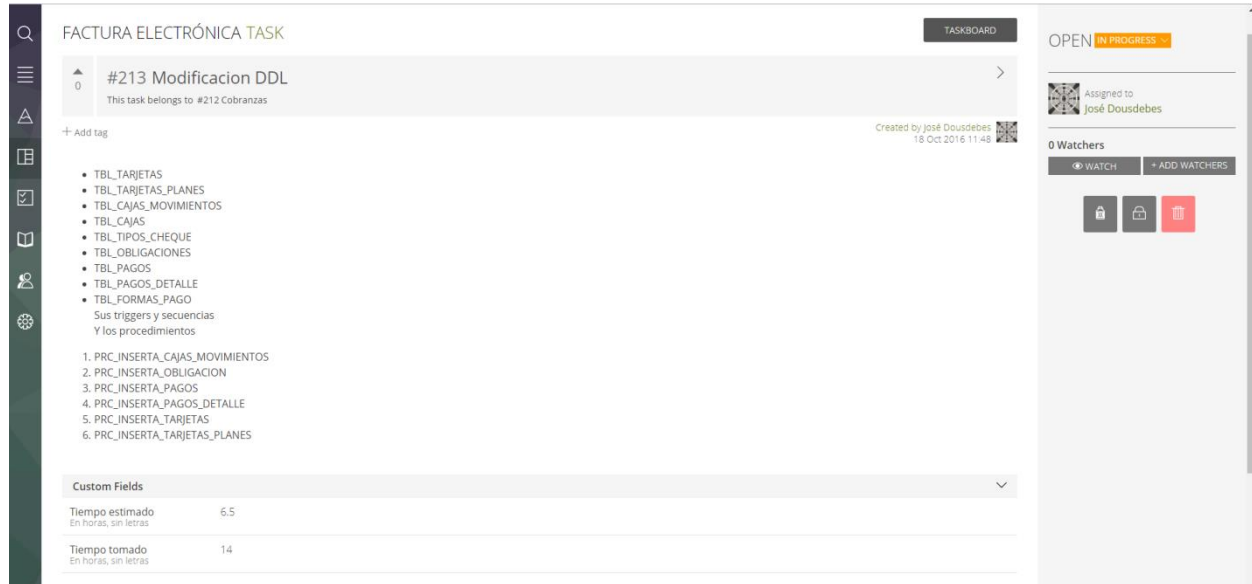


Ilustración 27 - Tarea en Taiga

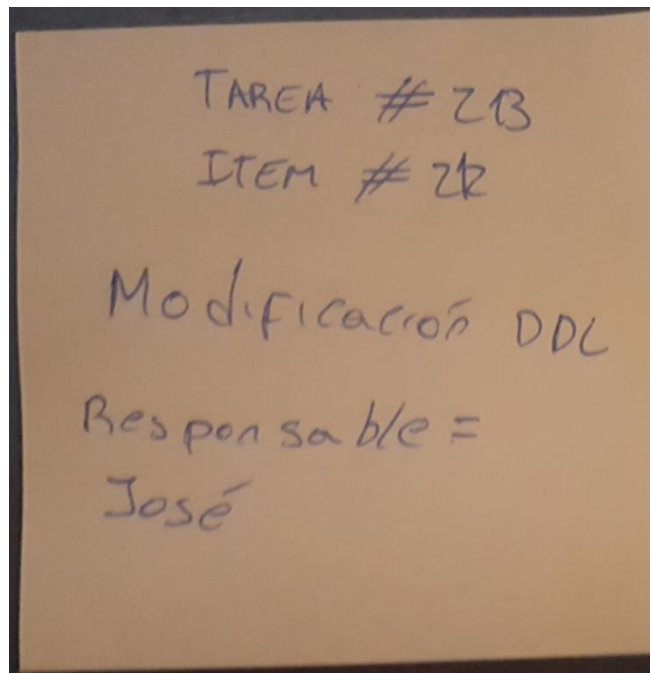


Ilustración 28 - Tarea en papel

Cuando se emplea el tablero de corcho se utilizan papeles de colores para representar pertenencia de ítems a un Proyecto, esto se debe a que se trabaja en más de un Proyecto a la vez, en la

ilustración siguiente el color celeste es para Dentis, Amarillo para el Colegio de Bioquímicos y Naranja para PucaráFactura. Se reserva el color Rojo para errores, independientemente del Proyecto que provengan.

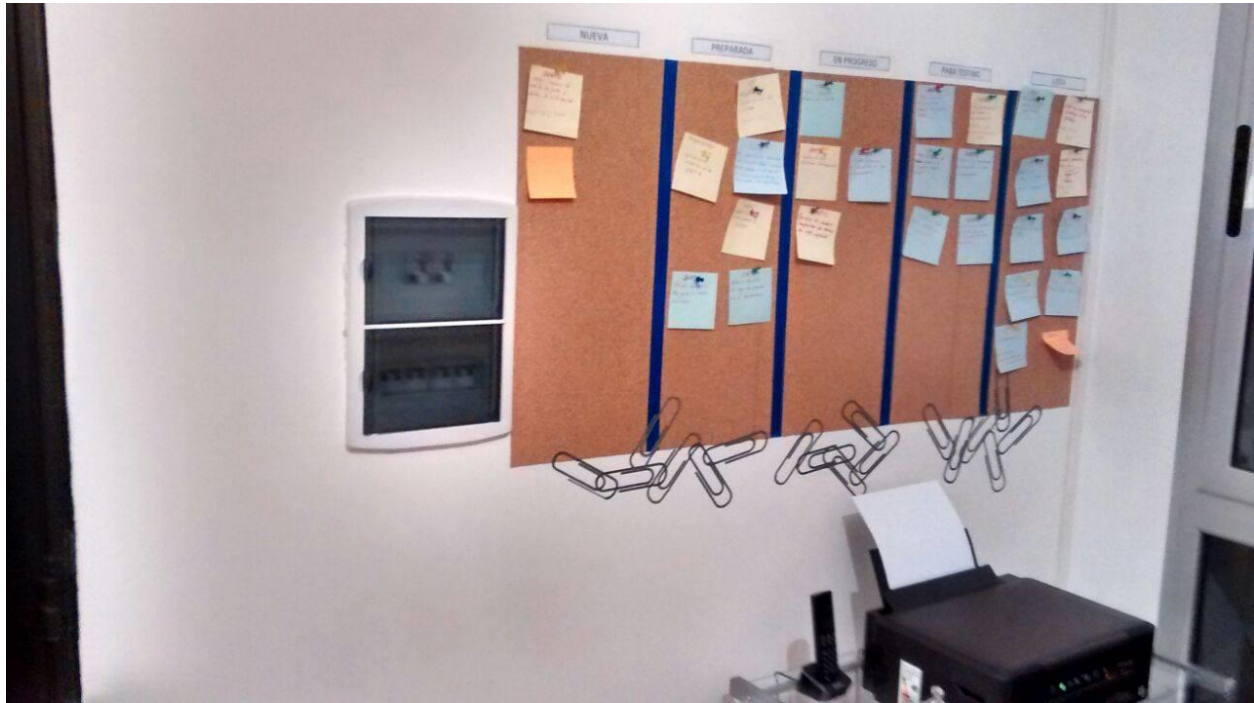


Ilustración 29 - Tablero de corcho



Ilustración 30 - Tablero en Taiga

Lista del proyecto

La Lista del Proyecto está compuesta por requisitos planteados por el Cliente y necesidades de la Solución, estos requisitos y necesidades se representan mediante ítems priorizados en esta lista, de la cual se desprenderá el trabajo iterativo e incremental del proyecto.

El caso de Factura Electrónica representa una situación particular debido a que el Mercado es el Cliente del proyecto y no una sola persona; además del feedback que proporcionan los usuarios, Sebastián y Alberto actúan como EP al plantear las necesidades y rumbo del producto.

#142 [MercadoPago] Pantallas de trackeo de pagos/cobros	New	?
#116 ABM Productos	New	?
#118 Estadísticas	New	?

Ilustración 31 - Extracto de la Lista del Proyecto

La ilustración 31 muestra un extracto de la Lista del proyecto de Factura Electrónica, el ítem #118 y #142 tienen lugar en base a la Visión establecida, mientras que la existencia del ABM de Productos se ve influenciada por necesidades planteadas por varios Clientes.

Lista de ítems

Los ítems que se trabajaran en una iteración dada están tomados de la Lista del proyecto, al mantener una lista priorizada, la elección no debería presentar mayores inconvenientes.

10	
26 Sep 2016-10 Oct 2016	
10 closed	
10 total	
#191 Modificar Template FE	1
#121 Tutorial interactivo	5
#166 Nuevo Indicador en tablero de ventas	3
#161 Paginas de error amigables	1

Ilustración 32 - Iteración 10 de Factura Electrónica

La ilustración anterior muestra los ítems que el Equipo se comprometió a entregar en esa iteración, para la misma el objetivo principal era terminar de modernizar y hacer más amigable la UI y UX, los ítems #161, 121 y 191 contribuyen a cumplir este objetivo, mientras que el ítem #166 involucra la inclusión de un nuevo indicador en el tablero de ventas, que brindará información totalizada sobre la forma de pago de los comprobantes.

Errores

Los errores representan un tipo especial de ítem. Tiene las mismas características de un ítem común, pero debido a su naturaleza requiere un trato prioritario, por esto se le agrega la característica de Prioridad, que puede variar entre Normal, Importante y Crítica, en Taiga esto se representa mediante un campo adicional, y en el Tablero físico se manejará mediante su posición vertical.



Ilustración 33 - Representación de un Error en Taiga

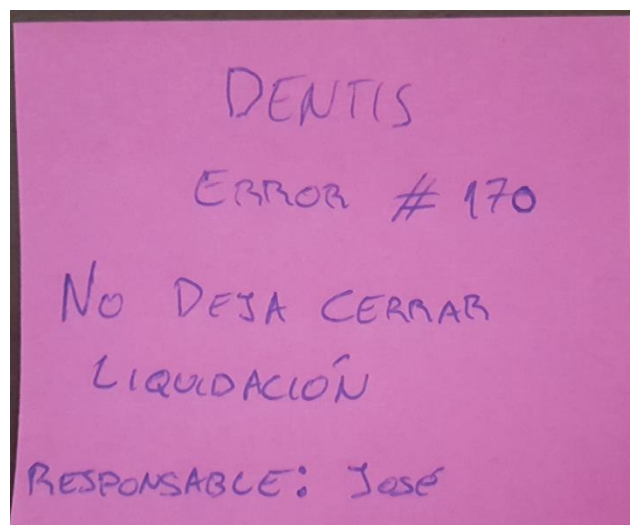


Ilustración 34 - Error en papel

Estados

Los estados por los que puede pasar un ítem son:

- **Nuevo:** Un ítem en estado Nuevo significa que ha sido sacado de la Lista del Proyecto, pero no contiene la suficiente información para poder ser desarrollado o bien no tiene asignado un Desarrollador. Cuando un ítem se saca de la Lista del Proyecto y se coloca en la Lista de ítems tiene esta clasificación (a menos que no requiera información adicional). Todos los ítems que compongan la Lista del Proyecto tendrán el estado Nuevo.
- **Preparado:** Cuando un ítem tiene información suficiente para ser desarrollado y tiene asignado un Desarrollador se considera como Preparado.
- **En progreso:** Significa que el ítem está siendo desarrollado, en este estado se desarrollan las tareas de la fase Desarrollo del Marco.

- Listo para testing: El testing de la funcionalidad se realizó en el estado anterior, este representa un testing de validación por una tercera persona.
- Hecho: Los requerimientos del ítem se han completado.
- Implementado: El ítem ha sido implementado en Producción y aceptado por el Cliente.

Los estados por los que puede pasar una Tarea son:

- Nueva: La tarea ha sido definida pero todavía no se comenzó a trabajar.
- En progreso: La tarea se encuentra bajo desarrollo.
- Lista: La tarea se ha completado y se encuentra sin errores.

Iteraciones

Para poder demostrar el funcionamiento del Marco en Pucará se describirá el trabajo de iteraciones. El Proyecto en cual nos centraremos será el Sistema de Gestión para el Colegio de Bioquímicos de Salta; cuando sea necesario aclarar o visitar algún punto del Marco que no haya sido necesario por el primer proyecto usaremos PucaráFactura, Cámara del Tabaco de Salta, Combustibles del Norte y Dentis.

Definición

Identificación de las necesidades del Cliente

PucaráFactura

Las necesidades de PucaráFactura están gobernadas por requerimientos del Mercado, el Proyecto ya cuenta con un producto estable, con muchos requerimientos ya implementados. Las necesidades para esta iteración siguen el orden de prioridad de la Lista del Proyecto. Las mismas representan mejoras en la UX, UI, referidas al registro de usuarios; trackeo de publicidades y funcionalidades de Facturas.

Como dijimos, la dirección que toma el proyecto está determinada por el comportamiento y necesidades del Mercado. Para la facturación electrónica se distinguen dos clientes, aquel que emite una gran cantidad de facturas, y el pequeño monotributista que emite una pequeña cantidad. Los factores por los cuáles el sistema será juzgado de calidad por el Cliente depende de cuánto les facilite su tarea. En el primer caso el Cliente valorará la posibilidad de cargar y procesar facturas en baches, contar con un panel de resumen de información, entre otros; el segundo Cliente tendrá en buena estima un sistema intuitivo y sencillo.

Bioquímicos

El proyecto comienza cuando Sebastián se contacta con Marcia, la encargada de manejar el sistema informático actual del Colegio, para plantear la necesidad de migración del sistema de escritorio a un sistema web, alojado en la nube.

En ese momento sistema estaba desactualizado tanto en requerimientos funcionales como no funcionales, su mantenimiento resultaba complejo y costoso al no estar alojado en la nube y haber sido desarrollado con tecnologías que el equipo ya no manejaba.

En base a esto se toma la decisión de migrar el sistema de escritorio a un sistema basado en la nube.

Cuando ambas partes concuerdan sobre esta decisión inicial, Sebastián coordina una reunión de Definición para recolectar los requisitos.

En esta reunión Marcia indica que el sistema actual cumple con la mayoría de sus requisitos, sin embargo, le gustaría que:

- Tenga mejor UX en la sección Laboratorios (ABM), específicamente en la parte del paso a paso.
- Poder realizar reportes personalizables sobre bajas y altas de Laboratorios.
- Contar con usuarios para cada persona que utiliza el sistema.

Para poder definir de la mejor forma posible las características a cumplir de cada requerimiento propuesto se toma la decisión de documentar ítems en base al sistema actual, basándonos en las pantallas y su comportamiento. De esta forma se elabora una lista de requerimientos avanzada – contiene más información que una lista tradicional.

A continuación, se muestra un extracto de la Lista con la que trabajó el Equipo.

LISTADO DE LABORATORIOS

Alta, baja y modificación de Laboratorios :: Colegio de Bioquímicos

Nuevo Modificar Eliminar Grabar Cancelar Imprimir Salir

LISTADO DE LABORATORIOS LABORATORIOS

Filtros de Búsquedas:

Departamento: Localidad: Especialidad: Tipo de Análisis:

Domicilio del Laboratorio:

NÚMERO	NOMBRE	DIRECCIÓN	TELÉFONO	FAX
1228	ABRAHAM GRACIELA ESTER	LOS FRESNOS 413 B° TRES CERRITOS	4333057	
1374	ABUD LILIANA MARCELA	BUENOS AIRES 179	4210158	
1419	ACOSTA ALICIA ESTER	AVDA. INDEPENDENCIA 1151	4230910	
1380	ADEN ANA CAROLINA	JUAN XXIII Y L. ALEM	03873-15519777	
1159	AGUIRRE FERNANDO RAMON	AVDA. SARMIENTO S/N	0387-154154758	
1130	AGUIRRE ROGELIO	CORONEL VICENTE TORINO 161	4939829	
1382	AGUIRRE SILVIA LIA	24 DE SETIEMBRE 464	03878-471477	03878-471477
1184	ALMENDRO MARIA ELENA	GENERAL GUEMES 898	4317535	
1362	ALONSO ANTONIO GUMERSINDO	AVDA. 9 DE JULIO 291	4904508	
1227	ALVAREZ ELSA VICTORIA	ZUVIRIA 898	4315696	4315696
1341	AMADOR SILVIA GRACIELA	COLON 535	03878-423849	
1364	ANGEL ROBERTO CESAR	URQUIZA 150 1° PISO	4220053	
1339	ARAMAYO ALICIA GABRIELA	20 DE FEBRERO ESQUINA BUENOS AIRES	03878-491927	
1449	ARAMAYO CARMEN VIVIANA	SAN MARTIN 1331	4314550	
1229	ARAMAYO MIRIAN GRACIELA	JULIY 140	4318222 - 4315094	
1261	ARGOITIA MARIA JOSE	BELGRANO 1078	4220074	
1170	ARIAS MARIA CECILIA	DEAN FUNES 785	4228865	
1255	ARRECHE SONIA ROSANA	PARQUE BELGRANO 1° ETAPA MANZANA H C	154063069	
1325	ATENDOR PATRICIA MONICA	DEAN FUNES 620	6081272	
1294	AYALA LUISA TERESA	HIPOLITO IRIGOYEN 341	03878-422681	
1185	AZAR STELLA MARIS	SARMIENTO 771	4212997	4212997

Este es el formulario más complejo del sistema, está desarrollado como un wizzard. En cada una de las solapas el sistema va haciendo las validaciones necesarias.

Para dar de alta un Laboratorio el usuario tiene que ir a la solapa Laboratorios y completar los datos necesarios. Si el usuario quiere dar de baja o modificar un laboratorio, hace click secundario en la fila que corresponde y se le despliegan dos opciones, borrar o modificar, si selecciona modificar los datos actuales se cargaran en el wizzard.

Primera solapa:

Consulta de Laboratorios :: Colegio de Bioquímicos

Salir

LISTADO DE LABORATORIOS

LABORATORIOS

Número de Laboratorio: 1228

Reactivos

Equipamiento

Datos Finales

Datos Generales

Nombre del Laboratorio (*): ABRAHAM GRACIELA ESTER

Domicilio del Laboratorio (*): LOS FRESNOS 413 B* TRES CER

Departamento: CAPITAL

Superficie total del Laboratorio (*): 16
Min. 6 m2, con un ancho mínimo de 2 mt.

Superficie en mesada (*): 3.6
Min. 3 m2, distribuidos en 1 o mas mesadas, 5m x 0,60 cm

Material Impermeable: PORCELANATO

Mesadas:

☒ Bajo Mesadas ☒ Estantes sobremesadas

☒ Área administrativa independiente de zona de trabajo

Teléfono: 4393057

Fax:

Localidad: SALTA

C.P.: 4400

Carteles:

☐ Placa Identificatoria ☒ Cartel ☐ Placa Mural

Ubicación:

Ubicación: POLICONSULTORIO

Sala de Extracción Independiente:

Superficie: 3.4

☒ Mesa carro con material adecuado para extracciones

☒ Camilla

☒ Apoya Brazo

☒ Material descartable y de vidrio

Boxes: 1

Informes Modalidad:

☒ Soporte Informático

☒ Computadora

☒ Impresora

☒ Archivo de Resultados

☒ Instrucciones escritas para los pacientes

☒ Sanitarios Independientes

☐ Paredes revocadas y pintadas hasta un alto no menor a 1,80 mt.

☐ Zona de lavado con pileta y con conexión de agua corriente y desagüe

☐ Hace domicilio

Validaciones: El nombre, el domicilio, la superficie total, la superficie de mesada son datos obligatorios.

Definición de la Visión y objetivos

PucaráFactura

PucaráFactura tiene como Visión minimizar el tiempo administrativo invertido y proveer a los Clientes de un Sistema que les permita administrar sus activos de forma eficiente.

Para satisfacer esta Visión, PucaráFactura tiene como objetivos la:

- Disminución de tiempo administrativo.
- Simplificación del proceso fiscal y de emisión de comprobantes.
- Personalización de comprobantes.
- Generación automática de elementos Contables.
- Proveer un Sistema que facilite la gestión de Compras y Ventas.

Bioquímicos

El Objetivo del sistema de gestión que tiene el Colegio de Bioquímicos de Salta es el de posibilitar la administración del registro de Laboratorios en la provincia de Salta, como así también controlar la asignación de Recursos y Personal a los mismos.

La Visión que se tiene para el Proyecto es la migración del Sistema existente a una plataforma web, con mejoras a la UX y UI, y permitir realizar reportes sobre las altas y bajas de Laboratorios.

Determinación de la Arquitectura

Bioquímicos

La Arquitectura del Sistema estará basada en el patrón de diseño Modelo-Vista-Controlador (MVC). El sistema estará alojado en la nube, esto quiere decir que tanto el Servidor de Aplicaciones como el Servidor de Base de Datos estarán online. Usuarios del Colegio de Bioquímicos podrán acceder al sistema mediante sus usuarios desde cualquier PC con conexión a Internet.

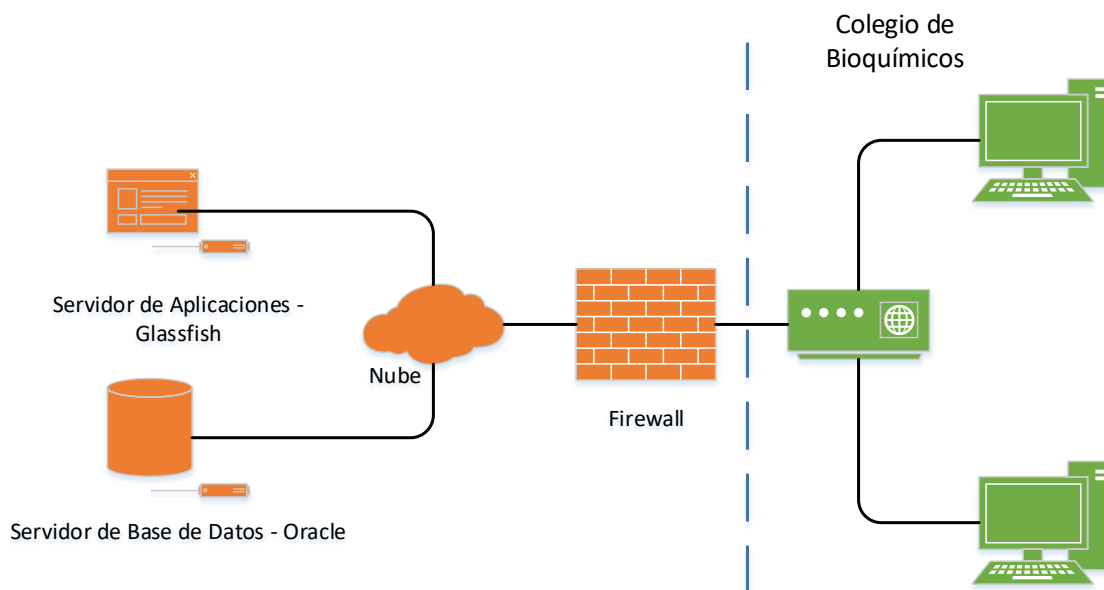


Ilustración 35 - Diagrama de red y comunicación. Gentileza de Pucará

La Arquitectura de datos, en su mayor parte representará la actual, sin embargo, será necesario hacer una modificación para permitir el acceso identificador en base a usuarios.

Para todos los proyectos se determinó la utilización de un DDL estándar para la Base de Datos, desde el nombramiento de Tablas y Atributos hasta la definición de Triggers, Procedimientos y secuencias. A continuación, un extracto para Tablas y Atributos.

- Las tablas comenzarán con TBL_ y su nombre será lo más descriptivo posible, sin pasarse de 3 palabras. Por ejemplo: TBL_CRITERIOS_ACEPTACION; TBL_ITEMS.
- Los atributos de cada tabla tendrán prefijo 3 letras, en el caso de que sea una tabla de 1 palabra, las 3 primeras; si es de dos, la primera letra de la primera palabra y las dos primeras de la segunda; si es de tres, la primera letra de cada palabra. Por ejemplo, para TBL_CRITERIOS_ACEPTACION: CAC_ID; CAC_NOMBRE.

Análisis de riesgos

Bioquímicos

Debido a que el Proyecto se centra en la Migración, más que en el desarrollo de nuevas funcionalidades, los riesgos no están gobernados por situaciones cambiantes, sino por la adaptabilidad de las funcionalidades del Sistema anterior a la nueva tecnología. De aquí se desprende la posibilidad de que un requerimiento no pueda ser representado de forma idéntica y se necesite realizar una modificación; el usuario final está acostumbrado a usar el sistema y cualquier cambio de UX en lugares no especificados le significaría una inversión de tiempo para aprender a usar. Por esto, se buscará representar de la forma más cercana posible la UX anterior.

Plan de entregas

Bioquímicos

Las características del Proyecto permiten realizar estimaciones y predicciones naturalmente más acertadas que en aquellos de naturaleza cambiante. A partir de la lista de requerimientos o necesidades elaborada por Sebastián (EP), se realiza la Lista del Proyecto

Votes	User Stories	Status	Points ▾
☐	▲ 0 #38 [Bioquímicos] Super-Item Laboratorio 1	New ▾	15
☐	▲ 0 #54 [Bioquímicos] Super-Item Laboratorio 2	New ▾	8
☐	▲ 0 #56 [Bioquímicos] Actualizacion de datos	New ▾	3
☐	▲ 0 #61 [ColegioBioquímicos] Como usuario quiero crear un nuevo laboratorio	New ▾	4
☐	▲ 0 #63 [ColBioSa] Traer todos todos los laboratorios	New ▾	2
☐	▲ 0 #60 [ColegioBioquímicos] Datatable y Datepicker en JQuery Steps	New ▾	3
☐	▲ 0 #112 [ColBioSa] Modificaciones varias	New ▾	3
☐	▲ 0 #117 [ColBioSa] Implementacion en produccion	New ▾	2
☐	▲ 0 #65 [ColBioSa] Modificaciones varias de UX	New ▾	3
☐	▲ 0 #114 [ColBioSa] Validaciones en Laboratorio	New ▾	2
☐	▲ 0 #113 [ColBioSa] Login	New ▾	5
☐	▲ 0 #127 [ColBioSa] Reporte de bajas de laboratorios en un periodo	New ▾	4
☐	▲ 0 #120 [ColBioSa] Como usuario quiero realizar reportes de listas de labs	New ▾	5

Ilustración 36 - Lista del Proyecto

Se acuerda terminar el Proyecto en un periodo de tres meses con iteraciones de 2 semanas. Las implementaciones del entregable se realizarán al final de la iteración, después de las 15 hs, independientemente del día.

El primer entregable representará el ítem más complejo del sistema, el Wizzard de 6 pasos del ABM de Laboratorios, este ítem será entregado en la segunda iteración, una entrega anterior resulta imposible por la atomicidad del mismo. Asimismo, el Equipo tendrá que realizar la tarea técnica de migración de BD, pasando de SQL Server offline a Oracle XE online.

La tercera iteración tratará el desarrollo de la sección de ABM de personal y directores de laboratorios, también se prevé perfeccionar el módulo anterior en base al feedback recibido por el Cliente.

La cuarta iteración se enfocará en el desarrollo del módulo ABM de Elementos y Reactivos, los cuales representan el stock de productos asignados a los Laboratorios.

En la quinta se atenderá el tema de Reportes “impresos” sobre Laboratorios.

Por último, en la sexta y última iteración se trabajará sobre el login y usuarios adicionales. Se deja para el último este tema ya que en la actualidad Marcia es la única persona que trabaja sobre el sistema, sin embargo, se presenta la necesidad de escalabilidad a más usuarios.

Estimaciones

Bioquímicos

Al contar con todos los requisitos inicialmente, y ante la poca probabilidad de modificación o cambio de los mismos, se toma como referencia la Lista del Proyecto para realizar la estimación grupal de Tiempo y Esfuerzo para cada ítem de la Lista. A esta altura las estimaciones no tienen en cuenta la subdivisión de algunos Ítems en Tareas, que tendrá lugar posteriormente en la reunión de Planificación.

Para todos los ítems, se realizó la estimación de tiempo y VD en la reunión de definición de Entregas, en ella sobre cada ítem el EP explicaba los requerimientos y los demás miembros expresaban su estimación sobre los valores del ítem.

En un caso, debido a la falta de experiencia del Equipo sobre el desarrollo de un Wizzard con tanta variedad y complejidad de elementos. Este fue la única vez en donde se empleó la estimación de tiempo por distribución Beta.

Siguiendo la formula

$$TE = \frac{(4TM + TP + TE)}{6}$$

Cada miembro del Equipo enunció los valores que consideraba correcto para el Tiempo esperado (TE), Tiempo pesimista (TP) y Tiempo más probable (TM), los valores del mismo tipo fueron promediados para poder calcular el Tiempo Esperado (TE) mediante la fórmula propuesta.

Determinación de necesidades de capacitación

PucaráFactura

Al tratar con AFIP, el usuario final del producto de factura electrónica tiene que estar completamente familiarizado con el funcionamiento del sistema antes de realizar cualquier transacción.

Bioquímicos

La familiaridad del Cliente con el funcionamiento del Sistema anterior implica una capacitación reducida, es decir sin documentación sin manuales de usuario, diccionario de datos, etc. Esto será verdad siempre y cuando no se altere drásticamente la UX.

Reunión de Planificación

La reunión de planificación para la primera iteración tiene lugar de forma posterior a la finalización de la fase de Definición, el Equipo acuerda trabajar sobre los ítems estipulados anteriormente (Wizzard). Para estos ítems el Equipo concuerda en que no será necesaria una definición inicial de Tareas. Al momento de empezar a trabajar con Bioquímicos el Equipo ya se encontraba trabajando en 2 proyectos adicionales, por ello, los desarrolladores se asignan la mínima cantidad de ítems, la suficiente para comenzar a trabajar, esto resultó beneficioso porque a medida que un Desarrollador terminaba un ítem podía elegir otro con libertad, eliminando así cuellos de botella.

Es importante notar que, al trabajar con varios Proyectos a la vez, se puede tornar un poco compleja la selección de prioridades por los desarrolladores a la hora de tomar nuevas tareas una vez comenzada la iteración. Esto fue solucionado a través de una priorización de ítems entre diferentes proyectos, por ej.: “Tenemos que tratar de terminar hasta el martes con el ítem X de PucaráFactura”; “En esta iteración tenemos que priorizar los ítems de Bioquímicos sobre los de Cámara del Tabaco”. Siempre se realiza la planificación pensando en cumplir con todas las responsabilidades, pero es prudente establecer prioridades sobre las mismas.

La iteración tiene la misma duración, fecha de inicio y fin para todos los proyectos, esto resulta en una enorme ventaja organizacional, al disminuir la cantidad de reuniones necesarias. Para el comienzo de esta iteración, se toma el valor de Velocidad de la iteración anterior de 4.55 para determinar la cantidad máxima de ítems a trabajar.

Diseño

Determinación de ítems a documentar

La necesidad de documentar un ítem, en todos los proyectos en que trabaja Pucará está determinada por el nivel de complejidad del requerimiento, o por la persona que asume responsabilidad sobre el ítem.

En el primer caso, queda a criterio del Equipo qué ítems documentar y qué tipo de documentación acarreará, sin embargo, hay excepciones, como por ejemplo: Al trabajar con la API para consumir MercadoPago, el Arquitecto (Alberto) determinó que deberíamos realizar una documentación del código en inglés, siguiendo patrones de diseño y nombramiento estándares para API's de pago.

La segunda condición tiene origen en la meta de lograr una organización orientada al aprendizaje y en donde cada miembro del Equipo (especialmente los desarrolladores) cuenten con el cúmulo de habilidades común al grupo. Es por esto que se documenta un ítem cuando va a ser desarrollado por una persona que no tiene mucha experiencia sobre el tema que trata.

Elaboración de modelos y Arquitectura

PucaráFactura

En la iteración comprendida entre XXX y YYY, el Equipo tenía que desarrollar un ítem que requería un Dashboard para el usuario final, ante la especificidad de los requerimientos del tablero se creó un Modelo de pantalla para acompañar a la descripción de requerimientos y funcionalidad del ítem.

La afirmación de que una imagen vale más que mil palabras es especialmente válida al tratar con elementos de la UI.

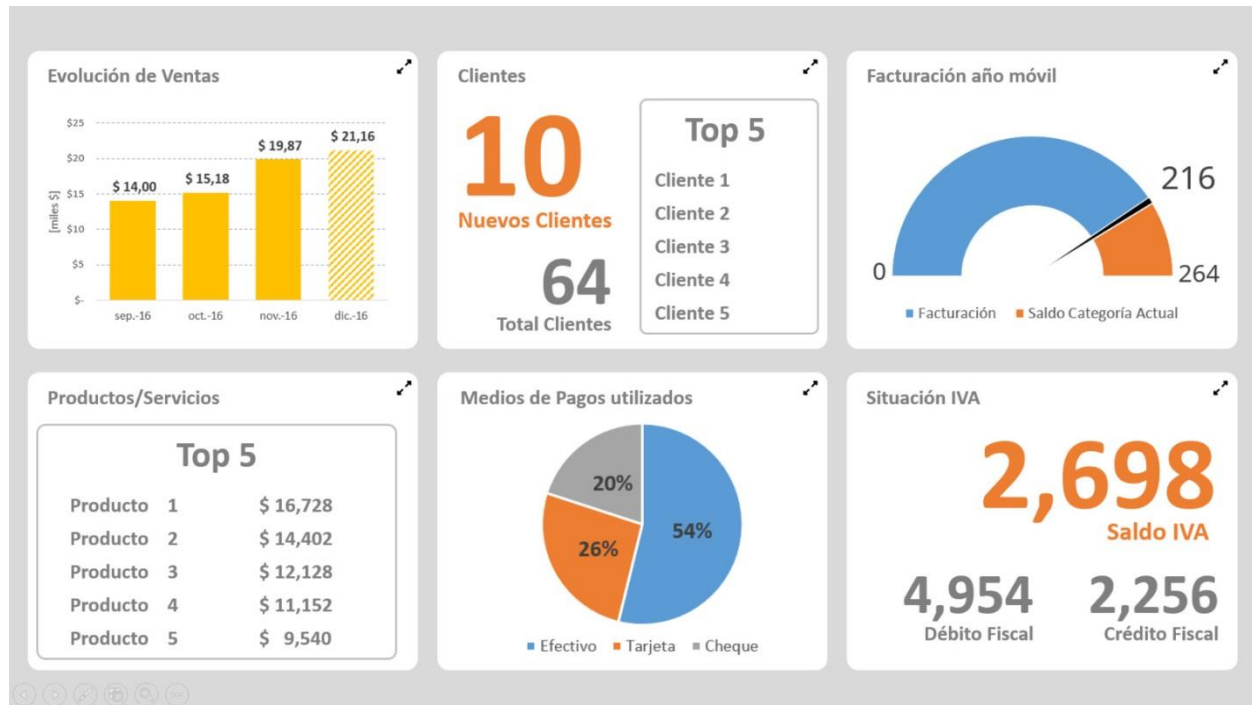


Ilustración 37 - Modelo para el Dashboard

Bioquímicos

Cuando fue necesario desarrollar el login para el sistema el Equipo tuvo como meta la escalabilidad y robustez. Para lograr esta meta, el equipo considera:

- Más de un tipo de usuario.
- Auditoría sobre fechas acceso.
- Recuperación segura de credenciales.

Todos estos elementos representan cambios en la Arquitectura Tecnológica, específicamente la Arquitectura de Datos. La ilustración 38 muestra el DER para diseñado cumplir con los parámetros anteriores.

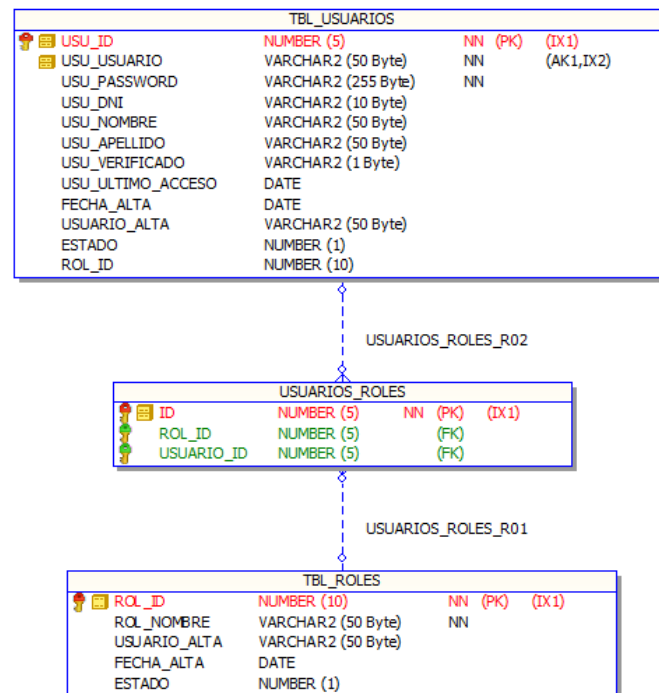


Ilustración 38 - DER para el módulo de Login

Análisis de la cultura informática

PucaráFactura

Si bien el producto está apuntado a los clientes con facturaciones medianas o grandes, ya que se les permite ahorrar una cantidad considerable de Tiempo Administrativo, estas personas u organizaciones cuentan con conocimientos informáticos, fiscales y contables más que suficientes para hacer uso del sistema.

Sin embargo, hay otro tipo de sector, que se encuentra en una situación diferente; la facturación electrónica es posible por la página de AFIP para todas las personas, sin embargo, la UX que proporciona es muy mala y no es acorde a las necesidades y capacidades del sector. Este target no cuenta con un sistema de gestión de ventas/compras, en muchos casos ni siquiera cuenta con una computadora, el dispositivo tecnológico reinante es el celular, la necesidad de crear un sistema intuitivo, sencillo y rápido deriva de esto.

Bioquímicos

El análisis de la cultura informática de la organización del Cliente tuvo sustento en la reunión de recolección de Requisitos y en la experiencia anterior de trabajo de Sebastián con Marcia. Se concluyó que debido a la familiaridad del Cliente con el Sistema no sería necesaria la elaboración de manuales siempre y cuando se respete el flujo del sistema durante la migración.

Definición de planes y directivas de capacitación

PucaráFactura

La empresa no se puede dar el lujo de visitar personalmente a cada usuario del Sistema para explicarle el funcionamiento, ya que la tarea sería demasiado costosa. La mejor forma de lograr que el Cliente se familiarice con las funcionalidades es mediante una capacitación cuyo tiempo de dictado dependa de ellos. Esto se puede lograr de 3 formas:

- Video-Tutorial
- Tour del Sistema en tiempo real
- Manual de usuario

Con la ayuda de opiniones de expertos determinó que la mejor forma de capacitar al usuario es mediante video-tutoriales, los mismos consisten en una demostración con ejemplos reales sobre las funcionalidades más relevantes del sistema. Este tipo de capacitación también actúa como marketing.

En las iteraciones del mes de noviembre del año 2016, para agregar valor a la solución, se implementó la segunda forma de capacitación, el tour (los desarrolladores lo conocerán como bootstrapour) consiste en un componente de UI que realiza una guía paso a paso indicando la función de cada componente de la interfaz.

Toda esta capacitación se realizará en un entorno de prueba llamado homologación, de funcionamiento idéntico al de producción, pero en donde las acciones efectuadas no tienen ningún efecto “permanente”.

Bioquímicos

Cuando se desarrolle una nueva funcionalidad, cuando se trate de un ítem complejo, o cuando la UX difiera mucho con respecto al sistema original será necesaria una demostración presencial del entregable, en todos los otros casos la reunión de demostración será reemplazada por un aviso de que la funcionalidad está implementada.

Para los demás proyectos, en general la capacitación y demostración tiene lugar de forma posterior a la implementación, el trabajo cercano con el EP permite que su validación sobre las funcionalidades sea suficiente y hasta el momento siempre acertada.

Desarrollo

Codificación y Testing

Una vez que se ha terminado la fase de diseño para un ítem se puede comenzar con el desarrollo de la misma. El desarrollador se guía por la descripción, criterios de aceptación y modelos de cada ítem para desarrollar las tareas técnicas necesarias para completar la funcionalidad. El

desarrollador crea “modelos” temporales que actúan como apoyo en esta labor de desarrollo, la entrega de cuadernos y lapiceras al Equipo prueba ser extremadamente útil.

En la API que corresponde a la integración de PucaráFactura con MercadoPago se implementa Acceptance TDD, los criterios de aceptación vienen dados tanto por requerimientos propios como por necesidad de la API de MercadoPago.

Para el resto de los ítems, cuando el Desarrollador haya terminado el desarrollo (codificación y testing manual) de los requerimientos, se implementa el ítem en el entorno de desarrollo y se avisa a Sebastián (el EP) para que valide los resultados. Si se tiene el visto bueno se realiza el commit al repositorio y el ítem pasa al estado Hecho, quedando disponible para implementación. En caso de que se detecten errores, Sebastián comunica los mismos al Desarrollador para su corrección, esta forma de trabajo se justifica por dos razones:

- Es posible por el ambiente de comunicación en la organización.
- Es “necesario” por la tecnología que se usa y el estado del Equipo. Antes de la implementación de API's REST (como la mencionada en el segundo párrafo) el Equipo trabajaba con tecnologías que hacían muy difícil el uso de tests unitarios o de regresión.

Cuando el EP no estaba disponible, se usaba una plantilla para poder llevar a cabo el testing. A continuación, se muestran los resultados de un testing efectuado sobre un ítem de PucaráFactura.

			
Estado Rechazada		Título Implementación de limitación de comprobantes	
		Fecha Asignada: 07/06/2016	
Nº de Revisión: 1			
Información de la Prueba			
Tipo de Prueba:	Funcional-Caja negra	Objetivo de la Prueba: Verificar que impida cargar facturas nuevas cuando esta desactivada la carga para ese usuario. Aclaración: Este valor de activado/desactivado se cambia de forma manual en la BD; solamente se verificará el funcionamiento de la lógica de negocios desde el lado de la aplicación, no desde la BD.	
Tester:	Seba		
Sistema:	Factura electrónica		
Nº Versión de Repositorio	594		
Base de Datos:	Producción		
Nombres de Usuarios:	demo	Proceso de Prueba	
Observaciones:		1. Login con el usuario DEMO 2. Carga Simplificada 3. Selección de archivo 1. Pre visualizar 2. Importar	
Fecha de Prueba:	07/06/2016	Prioridad:	-
Resultados			
Resultados Esperados: Comprobantes cargados al Sistema.		Resultados Obtenidos: Error en el paso 3 al intentar cargar el archivo. “No es posible generar un nuevo comprobante. Por favor póngase en contacto con nosotros a contacto@pucaratech.com ”	

Errores

Los errores son tratados como un ítem de alta prioridad, para resolver el error, es requerido que además de la descripción del problema en cuestión, se describan los pasos requeridos para

reproducir el error. El proceso de resolución varía de acuerdo al error tratado, pero para entender el concepto, describiremos el ciclo de vida del siguiente error.

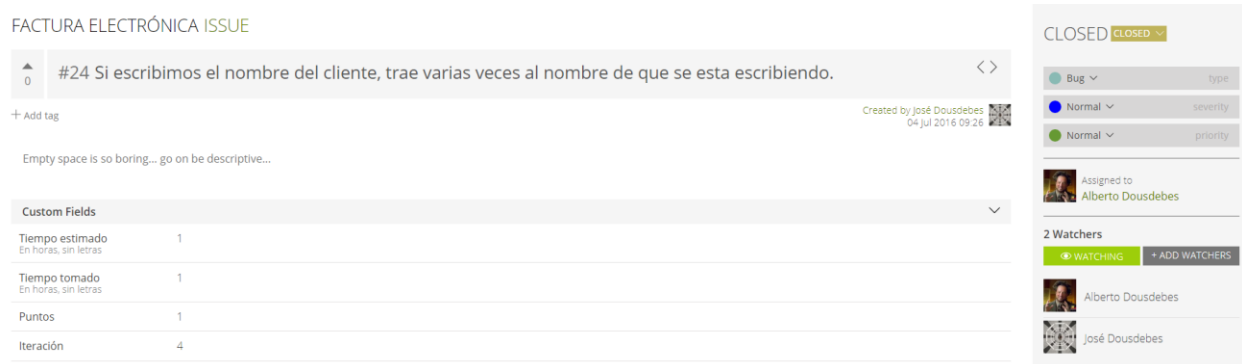


Ilustración 39 - Error de PucaráFactura

Primero es necesario entender el contexto: Los usuarios de PucaráFactura pueden tener clientes registrados; para hacer más sencilla la tarea de emisión de facturas, se implementó un autocomplete en todos los lugares donde se requiera ingresar el cliente. El problema consiste en que cuando hacía uso de esta funcionalidad, le traía repetidos algunos clientes.

Siguiendo las recomendaciones del marco para la resolución de errores, primero se observó la arquitectura, la función que trae estos resultados es común a todo el proyecto, así que ya tenemos el lugar a revisar. En este caso el error radicaba en la consulta SQL que se ejecutaba cuando se hacía uso de la funcionalidad.

Alberto se encargó de resolver este Error, una vez resuelto, se implementó la solución en el entorno de desarrollo y José realizó el testing, como no se encontraron errores, la solución al error quedó como Hecha, lista para implementación.

Documentación

De la Arquitectura

Para los proyectos de PucaráFactura, Dentis, Combustibles del Norte y Bioquímicos se realizó un modelaje de la Arquitectura, a través de un Diagrama de Implementación simplificado. En el caso de Dentis, debido a la necesidad de contar con un sistema offline en caso de que internet se corte se realizó un Modelo de Red, en forma conjunta con el Diagrama de Implementación para que los datos mantengan la consistencia antes, durante y después del corte.

Del usuario

Para la Cámara del Tabaco de Salta (Boletín Wordpress), se determinó como necesario un Manual de Usuario, el desarrollo de este tipo de ítem, tiene que estar en el lenguaje del usuario, apoyándonos en los resultados obtenidos mediante el análisis de la cultura informática. En este caso el usuario final es una persona mayor, con conocimientos informáticos apenas suficientes

para el manejo de la herramienta. Esto nos indica que el manual debe tener un buen nivel de granularidad, detalle y una demostración paso a paso en los casos siempre que sea posible.

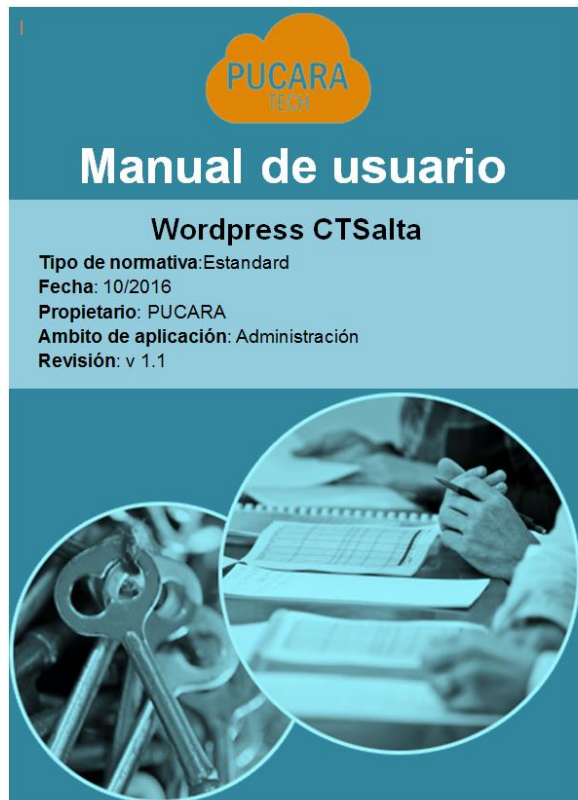


Tabla de contenidos	
Entradas.....	4
Todas las Entradas.....	4
Acciones en Lote.....	4
Editar.....	4
Mover a la papelera.....	4
Opciones sobre Entradas.....	4
Edición rápida.....	4
Editar.....	4
Papelera.....	4
Agregar nueva.....	5
Categorías.....	5
Visualizar todas las categorías.....	6
Agregar una categoría nueva.....	6
Medios.....	6
Add from Server.....	6
Media categorías.....	6
MailPost.....	6
Usuarios.....	7
Apariencia.....	7
Personalizar.....	7
Typography.....	7
Dazzling options.....	8
Slider.....	8
Layout options.....	8
Header.....	8
Footer.....	8
Social.....	8
Identidad del sitio.....	8
Colores.....	8
Color de fondo.....	8
Color de texto del encabezado.....	9
Imagen de cabecera.....	9
Imagen de fondo.....	9
Menú.....	9
Menu1.....	9
Reordenar.....	9
Quitar.....	9
Agregar.....	9
Social menu.....	9
Agregar.....	9
Widgets.....	10
Sidebar.....	10
Footer Widget.....	10
Agregar un banner.....	10
Modificar un banner.....	10

Ilustración 40 - Índice del manual de usuario CT Salta

La ilustración 40 representa el manual en su última versión, sin embargo, la construcción del mismo se dio en iteraciones.

En PucaráFactura hay algunas acciones que son de naturaleza repetitiva ante el alta de un nuevo cliente. Para estos casos se creó un manual de usuario que contiene un procedimiento paso a paso. Estos casos son:

- Aceptación de delegación del servicio.
- Alta de punto de venta del Cliente.
- Pasos delegación del servicio por el Cliente.

Técnica

El Equipo, exceptuando aquellos acuerdos de la fase de diseño, deja a consideración de cada miembro determinar qué parte del código requiere documentación; teniendo en cuenta siempre la máxima de que los comentarios contribuyan a la legibilidad del código y no produzcan el efecto contrario en una persona ajena.

Modificaciones

Un Cliente, en cualquiera de los casos puede requerir un cambio en cualquier momento. Los cambios son manejados como ítems. Por ejemplo:

En el proyecto de Bioquímicos, en la iteración que se desarrolló la funcionalidad asignación de RR. HH. a los Laboratorios se consideraban zonas de tiempo, Mañana y Tarde. Marcia (Cliente) se comunicó con el EP para indicarle que prefería tomar lapsos de dos horas para la definición de tiempos asignados, en vez de tener zonas de tiempos. Al momento que se requirió el cambio, el ítem todavía no se encontraba bajo desarrollo, pero la nueva estimación del equipo indicaba un incremento en la cantidad de tiempo y VD del ítem, por esta razón se tuvo que renegociar el entregable de la iteración.

Implementación

Capacitación

La capacitación de Combustibles del Norte, Dentis, Cámara del Tabaco y Bioquímicos tiene lugar en la Reunión de Demostración de entregables, a menos que el Plan de Capacitación de la iteración indique lo contrario. En estos casos la misma es de carácter presencial y demostrativa, donde el expositor expone el flujo de funcionamiento de los ítems del entregable a través de ejemplos cotidianos.

La capacitación en PucaráFactura sigue una modalidad diferente, en el aviso de implementación se indican las nuevas funcionalidades y de ser necesario (por complejidad de uso de alguna) una explicación de uso, si es una funcionalidad de mucho valor para el Cliente, la explicación se realiza con un vídeo paso a paso.

En la Cámara del tabaco, en base al análisis de la cultura informática del cliente, fue necesario realizar una capacitación en dos frentes, el primero fue el descrito en el primer párrafo, y se vio complementado por un Manual de Usuario, que se usaba para guiar la demostración.

Planificación e implementación

Toda implementación, en cualquier proyecto se realiza previo aviso por mail al Cliente, indicándoles el tiempo que se estima estará Offline el Sistema y los cambios a implementar. Esto es muy beneficioso, tanto desde un punto de vista administrativo como de marketing, porque el Cliente puede ver que periódicamente se agregan o mejoran funcionalidades; el punto de marketing es especialmente verdadero en el caso de PucaráFactura ya que la Reunión de Definición no se da con los usuarios.

Si no hay una funcionalidad crítica, de valor para el Cliente elevado o el Plan de Entregas indique lo contrario, las implementaciones se realizan después de finalizada la iteración, los lunes las 21Hs, así, si ocurre algún error se lo podrá tratar al día siguiente.

Para realizar implementaciones, el Equipo elaboró una secuencia de pasos, la mismo es válida para todos los proyectos de misma tecnología y arquitectura similar (Dentis, ColBioSa, Combustibles del Norte), y en términos genéricos es el siguiente.

1. Bloquear el Sistema al público
2. Completar transacciones pendientes
3. Realizar backup de la Aplicación y la BD
4. Desplegar cambios a la BD, verificando que no haya errores de compilación de scripts.
5. Desplegar cambios a la Aplicación WEB
6. Verificar que los cambios de la aplicación funcionen correctamente
7. Liberar el Sistema al público

En los primeros pasos del Equipo con una aplicación de Android para un pequeño grupo de usuarios, los incrementos (empaquetado .apk) se entregaban como descargables en una carpeta compartida, esto resultaba muy tedioso tanto para el Equipo como para el Cliente, el inconveniente fue notado en la reunión de retrospectiva de la misma iteración y se implementó un despliegue automático por la PlayStore para el siguiente entregable.

Reuniones

Definición

En la reunión de Definición se recolectan las necesidades del Cliente a través de una entrevista. En el caso de PucaráFactura, la reunión se da entre algunos miembros del Equipo, para determinar cuál es el estado actual del Mercado y así plasmar sus necesidades en requerimientos. Con esta información, el Equipo trabaja sobre los elementos de la fase de Definición.

Desde iteración posterior a la que da inicio al sistema, si se realiza una reunión de Demostración, la reunión de Definición puede ocurrir inmediatamente después, si la demo dura menos de 1:30 hs y si no hay impedimentos de parte de los miembros, es por esto que la planificación temporal resulta esencial.

Planificación

Una vez que se tiene el feedback del Cliente y se cuenta con la Lista del Proyecto Priorizada, el Equipo se reúne durante no más de 3 horas con el objetivo de dejar listo el trabajo para la iteración, estableciendo la Lista de Ítems, realizando estimaciones, determinando Tareas, asumir responsabilidades sobre Ítems y empezar a implementar el Plan de Mejoras. Salvo situaciones especiales, las responsabilidades sobre ítems se planifican de tal forma que todos los miembros del Equipo tengan inicialmente algo que hacer.

Diaria

La reunión diaria se da al inicio de la jornada laboral y es una reunión breve, de no más de 15 minutos. Los miembros del Equipo llegan a las 8 AM a las oficinas, después de que se

acomodan, se da unos minutos para que planifiquen lo que expondrán al Equipo en la reunión diaria. Esto típicamente resulta en anotar en los cuadernos las cosas importantes a hablar, así no se pasa por alto ningún tema importante, después que pasan 3-5 minutos, el LP llama al Equipo para que se reúna y se da inicio a la Reunión diaria donde los miembros mencionan en qué trabajaron, si tuvieron algún inconveniente y qué tareas afrontaran en el día.

Demostración

La reunión de Demostración es una reunión que no toma más de 1 hora y media, en ella una o más personas del Equipo expone al Cliente el entregable de la iteración. Si se trata con un entregable pequeño, en donde la demostración no tome demasiado tiempo, se puede optar por tener la reunión de Definición inmediatamente después, para lograr esto hay que avisarle de antemano al Cliente para que esté preparado.

Retrospectiva

Finalizada la iteración, el Equipo se reúne durante 1 hora para hablar sobre la forma en que se desarrolló el trabajo día a día de la iteración. Siempre se apunta a encontrar falencias más que elementos positivos, aunque reconocer esto último es importante. Si encontramos elementos a mejorar podremos elaborar un Plan de mejoras, en caso contrario el mismo estará vacío o dirá “Seguir trabajando de la misma manera”.

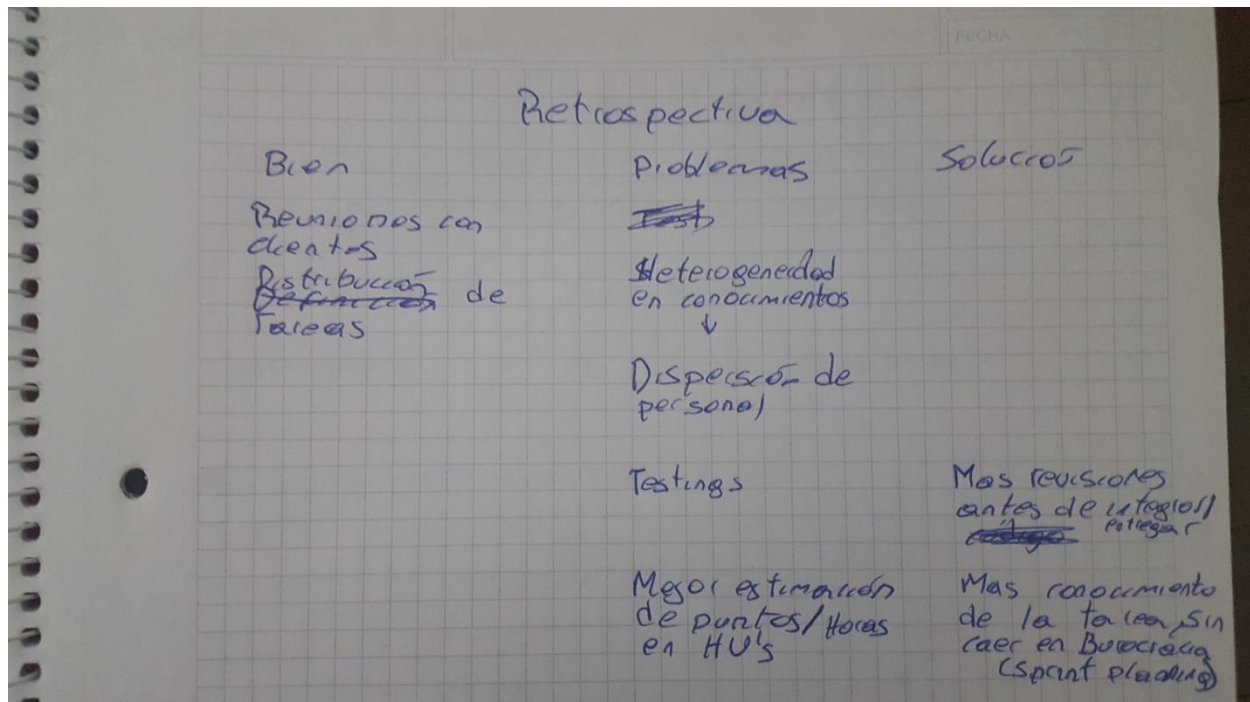


Ilustración 41 - Retrospectiva del día 15/07/2016

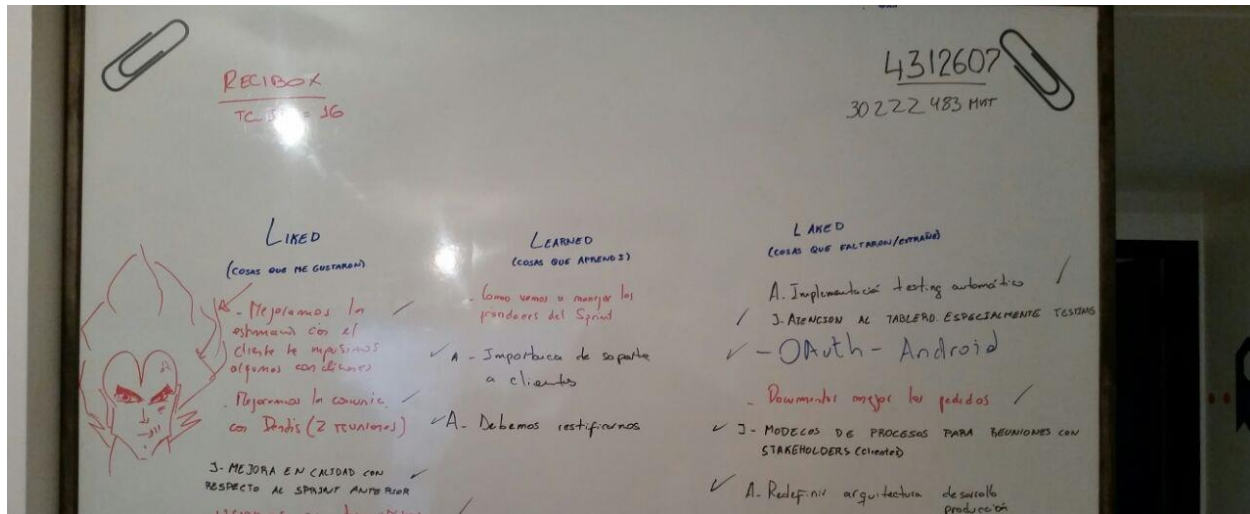


Ilustración 42 - Resumen de Retrospectiva del 01/07/2016

Análisis estadísticos

Una vez finalizada la reunión de retrospectiva, José, junto con Sebastián y Alberto se juntan para analizar las estadísticas que produjo el Equipo en la iteración.

El plan de mejoras se elabora principalmente en base al feedback obtenido en la retrospectiva y se apoya en estos números.

Análisis de rendimiento del Marco de trabajo

El transcurso de los hechos descritos en la implementación comprende un periodo de 3 meses, donde tuvieron lugar 6 iteraciones para desarrollar actividades de hasta 5 proyectos. La revisión de resultados se realizará como un análisis grupal de los proyectos trabajados en cada una de las 6 iteraciones.

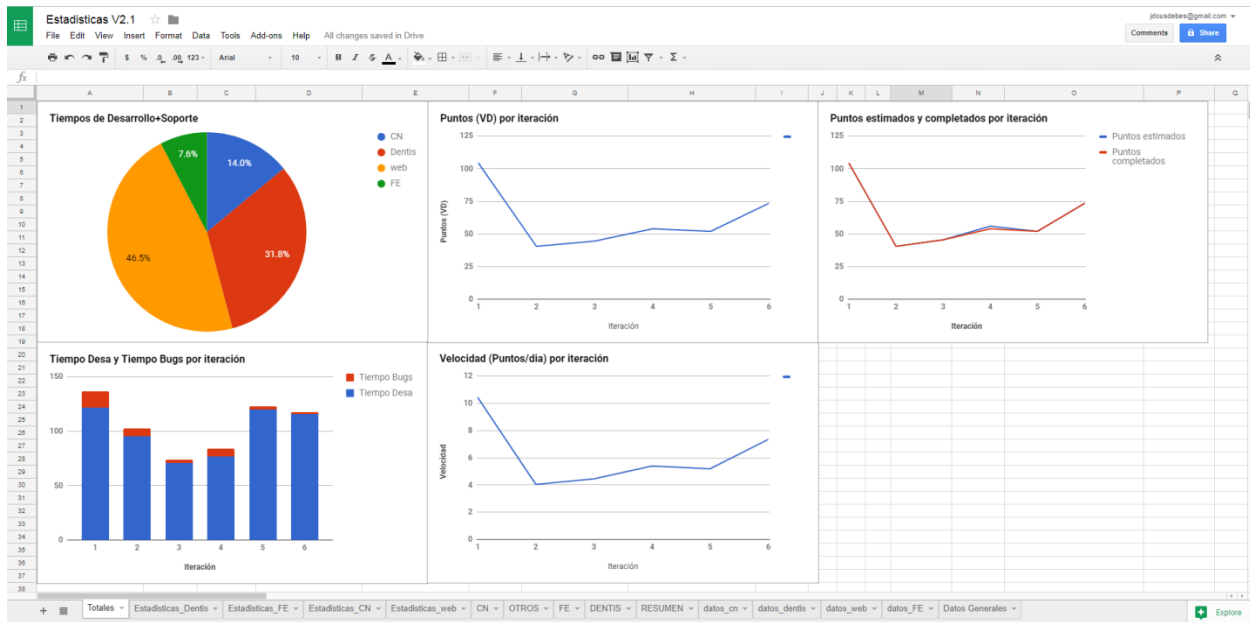


Ilustración 43 - Resumen de estadísticos

La ilustración32 comprende los principales estadísticos usados para la revisión del trabajo. Procedemos a explicar los resultados de cada uno. Los estadísticos comprenden 5 proyectos, pero figuran 4 porque para fines prácticos, la Cámara del Tabaco de Salta y Colegio de Bioquímicos de Salta se organizan bajo un mismo Proyecto.

Tiempo de desarrollo + Soporte

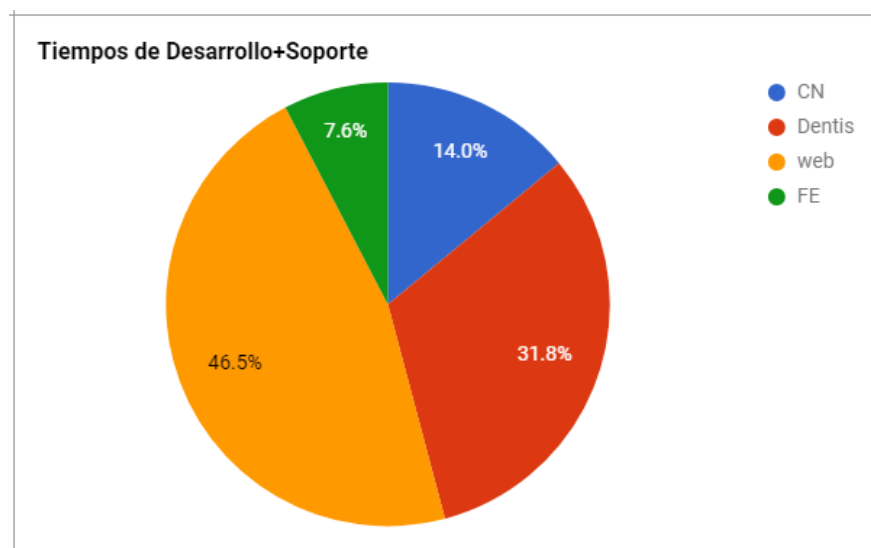


Ilustración 44 - Tiempo de Desarrollo más soporte de la iteración 6

Este estadístico nos indica qué porcentaje de tiempo demandó cada proyecto en la última iteración para desarrollar el entregable planificado y dar soporte al Cliente. Proyectos Web tiene un alto porcentaje porque se transitaba la última iteración de Bioquímicos.

Tiempos de desarrollo y bugs

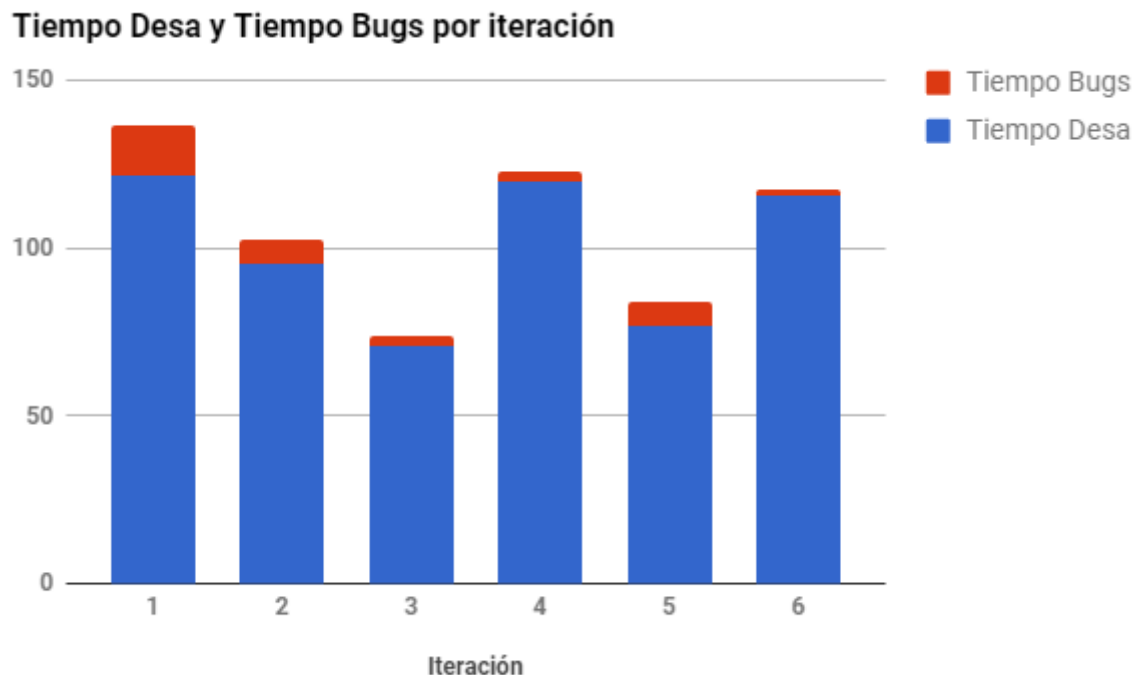


Ilustración 45 - Tiempo de Desarrollo/Bugs

El análisis de la relación tiempo desarrollo/bugs es un indicador de la calidad del software. Cuanto menos tiempo pase el Equipo corrigiendo bugs incrementará la cantidad de nuevas funcionalidades desarrolladas y el Cliente se sentirá satisfecho con el producto.

En la primera iteración, el Equipo pasó una gran cantidad de tiempo corrigiendo errores, esto también fue notado en la reunión de retrospectiva y el plan de mejoras, para mejorar esta situación, mencionaba la necesidad de mayores testeos por el Desarrollador antes de entregar la funcionalidad para que sea validada por el EP.

A medida que el Marco de Trabajo se fue integrando y perfeccionando en el Equipo se puede ver una clara disminución de la cantidad de tiempo destinado a corregir errores, produciendo un software de mucha mejor calidad que al principio.

En la tercera iteración, que tuvo lugar en la segunda mitad del mes de Julio del 2016, se ve una disminución del tiempo total del Equipo, esto ocurrió porque José pidió 1 semana libre para estudiar y rendir Dirección de Proyectos.

Puntos (Valor de dificultad) por iteración

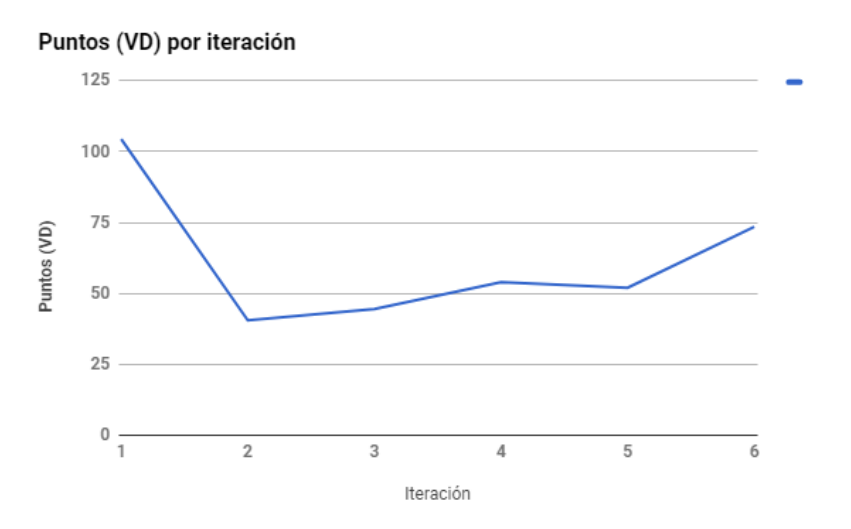


Ilustración 46 - Puntos por iteración

Los puntos estimados en cada iteración representan la cantidad de trabajo correspondiente al entregable completo, es decir los ítems + errores que surjan durante la iteración. Inicialmente parecería que le Equipo sufrió un declive importante de productividad, sin embargo, este no es el caso, en la primera iteración se migró de tema (UI) de la web de PucaráFactura, en la reunión de Planificación el Equipo consideró que era una tarea muy compleja por las particularidades de algunas pantallas, sin embargo, este no resultó ser el caso y en la reunión de Retrospectiva fue mencionado. Ya que si bien era una tarea que demandaba cierto tiempo, la misma no era de la dificultad estimada inicialmente.

En iteraciones posteriores vemos que, al contar con elementos contra los cuales comparar el esfuerzo requerido, y al tener experiencia de coocurrido en la primera iteración, se suavizó la línea, notando un incremento gradual la eficiencia del Equipo al desarrollar entregables cada vez más grandes. Para poder lograr esto, el aumento de las habilidades del Equipo tiene que estar acompañado de tomar el riesgo de empujar el límite de lo que puede desarrollar el Equipo.

Puntos estimados y completados por iteración

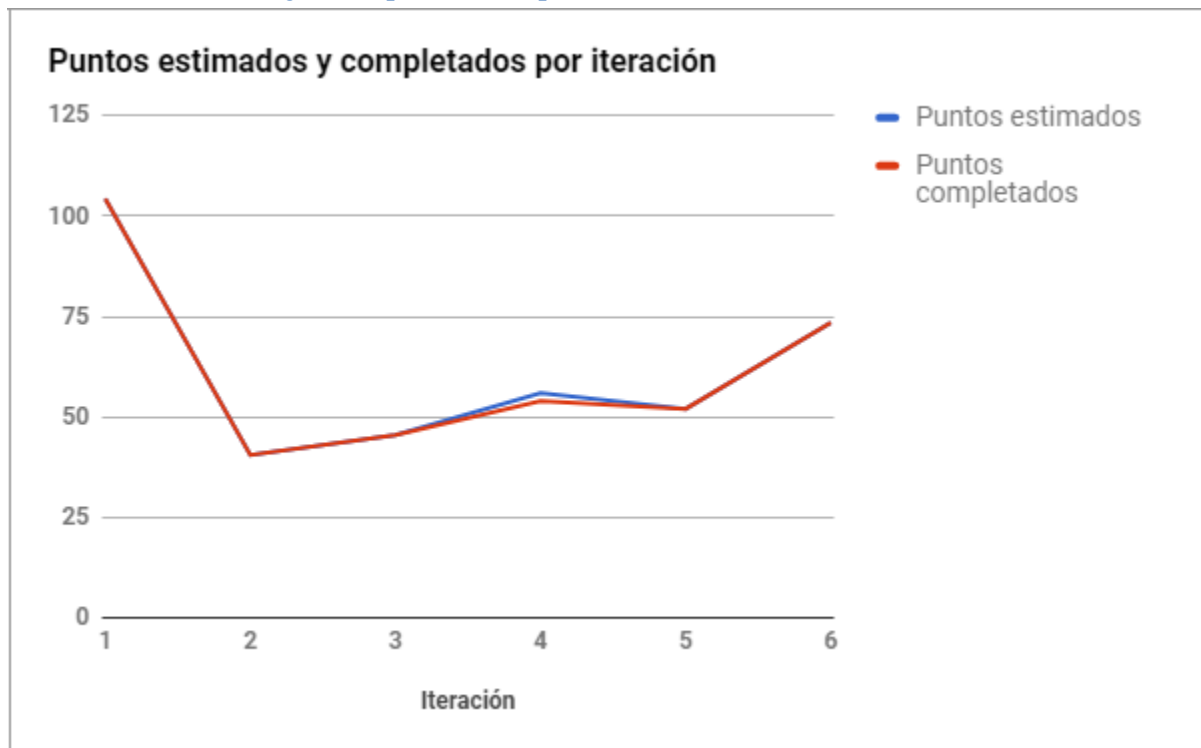


Ilustración 47 - Puntos estimados/completados por iteración

De relación muy estrecha con el estadístico anterior. Idealmente los puntos completados comprenden la totalidad de los estimados, esto indica que se desarrolló por completo la Lista de Ítems y de forma más amplia que la Planificación fue correcta.

En la cuarta iteración vemos que esta situación lamentablemente no se dio, se habían planificado entregar 56 puntos y se terminaron entregando 54, por esto, en la siguiente iteración, se disminuyó de 56 a 52 la cantidad de VD a afrontar, con resultados positivos.

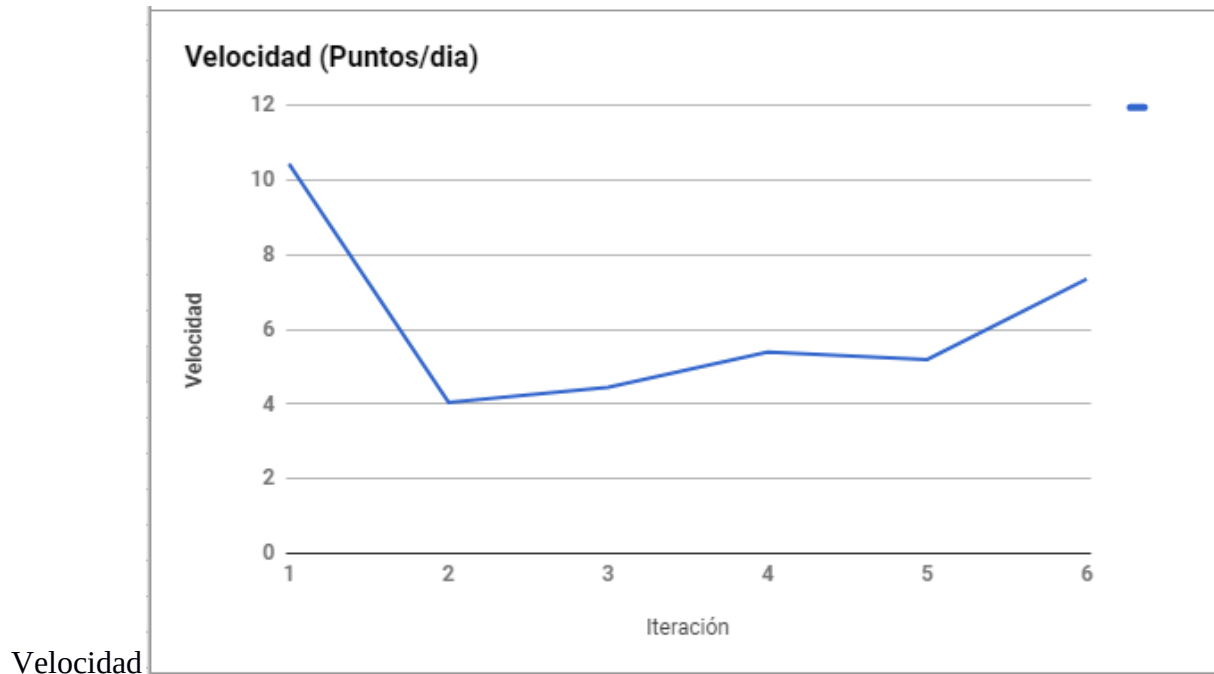


Ilustración 48 - Velocidad por iteración

La velocidad del Equipo es el medidor de eficiencia por excelencia. Sacando de lado el outlier de la primera iteración, la eficiencia del Equipo aumento de 4.05 a 7.25 puntos por día en 4 iteraciones o dos meses, esto significa una mejora del 55.8%, la cual no es nada despreciable.

Estadísticos adicionales

Jose-Bugs-Suporte		1	1	José-Items	11	Jose-Todo	13	CN	Dentis	web	FE	Varianza con respecto a estimación Overtime (horas)	0.019469983
Tiempo Estimado (horas)		1	1	Tiempo Estimado (horas)		41	43	0	11	32.00	0	6.00	
Tiempo Invertido (horas)		1	1	Tiempo Invertido (horas)		47	49	0	12.5	36.50	0		
Puntos		0.5	0.5	Puntos		26.5	27.5	0	10.5	17.00	0		
Puntos Terminados		0.5	0.5	Puntos Terminados		26.5	27.5	0	10.5	17.00	0		

Ilustración 49 - Estadísticos varios de José para la cuarta iteración

La ilustración 49 muestra un conjunto de estadísticos individuales y por proyectos, los mismos tienen como objetivo llevar un seguimiento detallado para establecer planes de acción. Por ejemplo, en esta iteración José se dedicó exclusivamente a ítems de Dentis y Web (Bioquímicos y Cámara del Tabaco), con un 74.4% del tiempo destinado a estos últimos. Casi todo el tiempo estuvo dedicado a nuevas funcionalidades.

El estadístico “Varianza con respecto a la estimación” indica el nivel de desvío de la estimación de tiempos, en este caso la variación no es mucha, un 1%, si este valor fuera más elevado, el plan de mejoras incluiría una revisión de las estimaciones.

El Overhead es un complemento del estadístico anterior, actuando como un indicador de la diferencia entre el tiempo estimado e invertido.

Análisis

Análisis económico

El análisis económico apunta a representar la inversión necesaria para el desarrollo completo de este Trabajo, reflejando tanto la fase teórica como la verificación experimental.

Recursos humanos

Para el desarrollo de este trabajo se requieren 5 personas. Por un lado, tenemos lo necesario para el desarrollo teórico, con el Investigador y el Consultor como únicos actores; en la verificación experimental se tiene en cuenta al personal de la Empresa donde se realizó la misma, con 4 Miembros de Equipo ocupados a jornada completa durante un periodo de 3 meses.

Decimos que la Tesis requiere de 5 personas porque el Investigador también es Miembro del Equipo.

Nombre	Cantidad	Cantidad horas	Precio/hora	Subtotal
Investigador	1	350	\$150	\$52.500
Consultor	1	20	\$250	\$5.000
Miembros del Equipo	4	240*	\$250	\$240.000

Tabla 1 - Recursos Humanos

*La implementación del Marco se realiza en un periodo de 3 meses en PucaráTech.

Recursos de Hardware e Insumos

El hardware e insumos necesarios para el desarrollo de este trabajo son:

Nombre	Cantidad	Precio	Subtotal
Notebook HP Envy 15-ay197cl	4	\$22.973	\$91.892
Impresora láser HP M12w	1	\$2.900	\$2.900
Resma Ledesma A4 500 hojas	2	\$100	\$200
TV Philips 55PUG6700/77	1	\$25.000	\$25.000
Pizarra Blanca	1	\$1.500	\$1.500
Panasonic Kx-tgc222	1	\$1.900	\$1.900
Útiles varios*	1	\$1.000	\$1.000

Tabla 2 - Recursos de Hardware e Insumos

*Papeles de colores, pizarra de corcho, lapiceras, fibrones, borrador, cinta adhesiva y cuadernos.

Recursos de Software

La correcta elección de los recursos de Software es crítica para el desarrollo de cualquier trabajo, siempre que es posible se opta por aquellos con licencia GPL por las ventajas de seguridad, robustez y costo.

Nombre	Cantidad	Precio	Subtotal
Sistema de Gestión - Taiga.io	1	\$0 ¹⁵	\$0
Gitlab (10 usuarios)	1	\$0	\$0
Netbeans	4	\$0	\$0
Linux Debian 8.2	4	\$0 ¹⁶	\$0
Oracle XE 11	1	\$0	\$0
Servidor 60GB (3 direcciones IP)	1	\$500*	\$500

Tabla 3 - Recursos de Software

*Precio del servicio por un periodo de 3 meses

Recursos de Infraestructura

La Infraestructura representa aquellos recursos necesarios para mantener funcionando la oficina donde trabajará el Equipo durante la fase de verificación experimental, el inmueble tiene que tener los servicios básicos, y una buena conexión a internet resulta indispensable.

Nombre	Cantidad	Meses	Precio	Subtotal
Alquiler oficinas	1	3	\$6.000	\$18.000
Agua, Luz, Cloacas	1	3	\$700	\$2.100
Internet	1	3	\$500	\$1.500

Tabla 4 - Recursos de Infraestructura

Resumen

La Tabla 5 refleja el resumen de gastos necesarios para el desarrollo completo de este Trabajo.

Nombre	Subtotal
Recursos Humanos	\$297.500
Recursos de Hardware e Insumos	\$124.392
Recursos de Software	\$500
Recursos de Infraestructura	\$21.600
TOTAL	\$443.992

Tabla 5 - Resumen de Recursos

Como podemos ver, el desarrollo de software no es una actividad barata. Predeciblemente, por la naturaleza de la actividad, la mayor parte de la inversión está destinada a los Recursos Humanos; los recursos de Hardware e Infraestructura toman el segundo puesto como el tipo de recurso que

¹⁵ El código fuente de Taiga está bajo licencia GNU.

¹⁶ La distribución de Linux Debian se encuentra bajo licencia GNU.

más inversión requiere, las notebooks asumen una posición mayoritaria al requerir el 73.8% de ese total; la infraestructura tiene un impacto del 4.8% sobre el total, con una cifra que si bien no es menor, palidece en comparación con el total de casi medio millón de pesos; el costo de software es insignificante, esto se debe a una correcta elección de ese tipo de recursos, si el Equipo hubiera decidido ir hacia herramientas más tradicionales y con otro tipo de licencias la historia sería otra.

Análisis de riesgo

Los elementos del análisis de riesgo están principalmente enfocados a la verificación experimental del Marco, ya que ahí es donde se prueban las hipótesis planteadas en la solución propuesta. Sin embargo, se incluyen elementos cuya definición de riesgo está vinculada de forma indirecta con planteamientos del Marco de desarrollo.

Identificador	Riesgo	Probabilidad de ocurrencia	Importancia	Peso
F1	Se rescinde el contrato con el/los cliente/s	6%	10	60
F2	Crisis económica	13%	5	65
F3	Aumento en precios de hardware	22%	4	88
D1	Fallas de hardware	1%	7	7
D2	Corte de electricidad	10%	8	80
F4	Aumento de honorarios en trabajadores	10%	7	70
D3	Conflictos en el equipo de trabajo	5%	8	40
D4	Fallos en el servidor virtual	3%	7.5	22.5
D5	Renuncia/Abandono de algún miembro del equipo	20%	9	180
D6	Fallas en el servicio de internet	15%	5.25	78.75
D7	Incomprensión de la forma de trabajo	2.5%	7	17.5
D8	Fallos en la integridad de la base de datos	1%	9	9
D9	Fallo del Backup	0.01%	9.5	0.095
D10	Pérdida de validez del Marco de trabajo	0.001%	9	0.009

Tabla 6 - Tabla de riesgos

Plan de contingencia

Identificador	Riesgo	Probabilidad de ocurrencia	Importancia	Peso
F4	Aumento en precios de hardware	22%	4	88
D2	Corte de electricidad	10%	8	80
D5	Renuncia/Abandono de algún miembro del equipo	20%	9	180

Tabla 7 - Plan de contingencia

F4, Aumento en los precios de hardware: Si el aumento de precios supera el margen estipulado, hay varias opciones.

1. Buscar presupuestos nuevamente, incluyendo aquellos lugares que no fueron considerados al comienzo.
2. Buscar presupuestos por hardware similar, pero de menores prestaciones, de ser posible, dadas las condiciones técnicas y contractuales.
3. De ser posible, afrontar el costo excedente con fondos de la organización.
4. Renegociar con el Cliente.

D2, Corte de electricidad: Si se da un corte de electricidad en las oficinas, y se espera que el mismo no vuelva por la duración de la jornada laboral, habrá que despachar a los empleados a sus casas, donde trabajarán hasta completar la jornada, considerando el tiempo de traslado. En caso de que el corte se dé el día de una reunión, los participantes de la misma decidirán si se la puede realizar de forma remota o si hay que asentarla en otro día.

D5, Renuncia/Abandono de algún miembro del Equipo: Ante la salida de un miembro del Equipo hay que denegarle el acceso a la BD, Versionador, entornos de desarrollo, Sistema y Taiga (Sistema de gestión de proyectos). Desde el aspecto de administración de equipo hay que tener en cuenta la disminución en productividad que implicará contar con un miembro menos, el LP se basará en las estadísticas recolectadas para definir la velocidad de la iteración siguiente.

En caso de que haya alguna actividad en progreso por esta persona, la tarea deberá ser tomada por otro miembro del Equipo, donde estará a su criterio seguir o no la misma línea de desarrollo.

Conclusiones

La definición de un Marco de Trabajo implica muchísimo tiempo, al momento de escribir esto, el tiempo de edición es de 18.572 minutos, lo que equivale a casi 310 horas o 13 días continuos de trabajo, esto sin contar el tiempo invertido en la lectura de literatura.

Cuando se trabaja con este tipo de investigaciones es fácil navegar en literatura inesperada con el fin de proponer reglas o recomendaciones fundamentadas, esto lo experimenté al desarrollar los temas referidos a personas, los textos orientados a gestión de proyectos no tratan las características básicas de comportamiento humano, los libros de psicología sí.

Al trabajar con cualquier tema que trate sobre personas, el escepticismo y los conocimientos de ciencias duras resultan clave para que el trabajo no sea pura charlatanería; así, el método científico juega un papel importantísimo, ya que nos dice que una vez planteado hay que buscar formas de demostrar que no funciona, esto resulta muy difícil de mentalizar.

En definitiva, todo lo aprendido en la universidad tiene un punto de confluencia aquí, donde se aplican los conocimientos adquiridos tanto de los primeros años, como Análisis Matemático y, Probabilidad y Estadística, hasta los de los últimos como Modelos y Simulación y Dirección de Proyectos.

Futuras líneas de investigación

El avance del conocimiento tiene lugar en micro incrementos, hay excepciones de personas con un gran intelecto que provocan saltos agigantados, pero no es la norma. Es por esto que espero que mi micro-incremento sirva como un escalón más para aquellos determinados a responder una pregunta o a saciar su curiosidad.

Este trabajo deriva de forma directa las siguientes líneas de investigación:

- Técnicas de motivación generales, referida a personas dedicadas a la informática.
- Elaboración estadísticos que discriminando desempeño por tipo de tecnología.
- Desarrollar un método para determinar la velocidad de un Equipo que recién se inicia.
- Abordar implicancias legales de un Contrato Iterativo.
- Determinar técnicas de contratación en base a requerimientos y necesidades de cada rol.

Anexo

Glosario

- LP: Líder del Proyecto.
- EP: Encargado del Producto.
- Miembro: Toda persona perteneciente al Equipo, es decir, cualquier Desarrollador, EP, LP o Arquitecto.
- VD: Valor de dificultad.

Licencia

Este trabajo se encuentra bajo licencia Creative Commons: Attribution + ShareAlike3.0 (BY-SA). Donde cualquier persona es libre de copiar y distribuir este trabajo por cualquier medio o formato. Teniendo el permiso de modificar, transformar o construir en base a este material, bajo cualquier propósito, incluso uno comercial.

Se tiene que dar crédito al autor original (José Dousdebes), proveer un link a la licencia e indicar si se realizaron cambios. Todos los cambios o modificaciones que reciba este trabajo deben estar bajo la misma licencia, estando prohibida la inclusión de cláusulas legales o tecnológicas que impidan a terceros sobre prácticas permitidas por la licencia.¹⁷

¹⁷ Licencia CC BY-SA 3.0, <https://creativecommons.org/licenses/by-sa/3.0/deed.en>. Vigente al 16/08/2017

Bibliografía

Los siguientes son algunos de los textos que se esperan utilizar en el desarrollo de la tesis.

- Martín Alaimo, 2013. Proyectos ágiles con Scrum. 1^{ra} Edición. ISBN 978-987-45158-1-0
- Henrik Kniberg, 2015, Scrum and XP from the trenches. 2^{da} Edición. ISBN: 978-1-4303-2264-1
- Jurgen Appelo, 2011, Management 3.0. 1^{ra} Edición. ISBN: 978-0-321-71247-9
- Apuntes de la cátedra Sistemas 3.
- Apuntes de la cátedra Ingeniería del Software.
- Apuntes de la cátedra Modelos y simulación.
- Apuntes de Metodologías ágiles para gestión de proyectos de desarrollo de software, 2015. UTN.
- Apuntes del Curso de gestión ágil de proyectos, Scrum+Kanban. 2014, Escuela de Administración Pública.
- Object Management Group, 2012, Unified Modeling Language (OMG UML)- Superstructure. Versión V2.1.2.
- PMI, 2013, Guía de los fundamentos para la dirección de proyectos (PMBOK). 5^{ta} Edición. ISBN 978-1-62825-009-1
- David Heinemeier H. & Jason Fried & Matthew Linderman, 2006, Getting Real: The Smarter, Faster, Easier Way to Build a Web Application. 1^{ra} Edición. ISBN 9780578012810.
- Poelstra John, 2011, Are Your Meetings Toxic?. <http://johnpoelstra.com/are-your-meetings-toxic/>. Página vigente al 14/08/2017
- Peterson Jordan B., 2002, Maps of Meaning: The Architecture of Belief. ISBN 9780415922210.
- Roger S. Pressman, 2010. Ingeniería del Software, Un enfoque práctico. 7^{ma} Edición. ISBN: 978-607-32-0603-7.
- Ian Sommerville, 2005. Ingeniería del Software. 9^{na} Edición. ISBN: 978-607-15-0314-5