Este documento define como el proyecto se va a ejecutar, monitorear y controlar, y dar por finalizado así como también provee contexto sobre la visión del mismo, los usuarios e interesados, el alcance, el equipo de trabajo y el plan de trabajo.

Plan de gestión de Proyecto

PhoneTicket

Contacto: taller-de-desarrollo-ii-phoneticket-2do-2013@googlegroups.com

|  |  |
| --- | --- |
| **Versión** | **Cambios** |
| 1.0.0 | Versión Inicial previa a la primer reunión con el cliente |

Índice

[Metodologías tomadas como base 2](#_Toc365759402)

[Adaptación de Scrum 2](#_Toc365759403)

[Roles 2](#_Toc365759404)

[Gestión del alcance y planificación 3](#_Toc365759405)

[Gestión de tiempos 3](#_Toc365759406)

[Gestión de Riesgos 3](#_Toc365759407)

[Escala cuantitativa de impacto 3](#_Toc365759408)

[Definición Probabilidades 3](#_Toc365759409)

[Matriz de Probabilidad e Impacto 4](#_Toc365759410)

[Gestión de Cambios 4](#_Toc365759411)

[Alcance 4](#_Toc365759412)

[Tiempo 4](#_Toc365759413)

[Otros Cambios 5](#_Toc365759414)

[Descubrimiento de requerimientos y especificación funcional 5](#_Toc365759415)

[Definición de pruebas, criterios de aceptación, registro de resultados y tratamiento de defectos 5](#_Toc365759416)

[Definición de pruebas funcionales 5](#_Toc365759417)

[Regresión de pruebas funcionales 6](#_Toc365759418)

[Registro de resultados 6](#_Toc365759419)

[Tratamiento de defectos 6](#_Toc365759420)

[Severidad 6](#_Toc365759421)

[Ciclo de Vida 7](#_Toc365759422)

[Mantenimiento de trazabilidad del proyecto 7](#_Toc365759423)

[Elementos utilizados para comunicar 7](#_Toc365759424)

[E-mail 7](#_Toc365759425)

[Reportes de progreso 7](#_Toc365759426)

[YouTrack 8](#_Toc365759427)

[Reuniones 8](#_Toc365759428)

[Pruebas unitarias y elementos para registrar diseños 8](#_Toc365759429)

[Seguimiento y Control del Proyecto 8](#_Toc365759430)

[Especificación de atributos de calidad 8](#_Toc365759431)

# Metodologías tomadas como base

Utilizaremos las siguientes metodologías durante el transcurso del proyecto. Adaptaciones a las versiones originales de las mismas pueden verse en la sección de adaptación de cada una de ellas.

* Scrum ([adaptación](#_Adaptación_de_Scrum)). **Documentación metodología**: <http://static.squarespace.com/static/51e3f87ce4b0031a73dac256/t/51f6a2fde4b0eca7a9be140f/1375118077451/Scrum_Guide.pdf>

# Adaptación de Scrum

Las siguientes son las modificaciones que haremos a la metodología Scrum para este proyecto.

* La reunión deScrum no se hará de forma diaria. Se hará una reuniónScrumobligatoria los días Miércoles, Sábado y Domingo siendo opcional el resto de los días.
* La reunión de retrospectiva se hará de forma opcional. Queda a criterio del equipo de trabajo si la reunión debe hacerse o no.
* La duración por defecto de un Sprint será de 2 semanas. Esto puede adaptarse con anticipación según las necesidades de los involucrados.
* Cada 2 semanas habrá una **reunión con el cliente**. En la misma se realizaran el Sprint Review de la iteración anterior y el Sprint Planning de la iteración futura.
  + El cliente aprobará los entregables de la iteración pasada.
  + El ProductOwner explicará el trabajo a realizar en la siguiente iteración. (<http://www.scrumalliance.org/community/articles/2012/november/your-client-isn-t-your-product-owner>)
  + El cliente puede dar feedback respecto al backlog, en su rol de interesado.
* En vez de utilizar el concepto de ProductBacklog (que abarca solo el alcance de producto) se utilizará el concepto de Project Backlog. El mismo incluirá todas las tareas que serían parte del ProductBacklog, así como también tareas relacionadas con la gestión del proyecto. Por ejemplo un BacklogItem podría ser: *"Identificar los riesgos relacionados de laintegración con redes sociales"*.
* El ProductOwner y el Scrum Master son parte del DevelopmentTeam. En base a lo hablado con el cliente, estos roles rotarán en cada iteración. Es posible que en una iteración la misma persona oficie de Scrum Master y ProductOwner.

# Roles

Los roles del equipo de producto son los definidos por Scrum. Estos son:

1. ProductOwner
2. Developer
3. Scrum Master. El Scrum Master en cada iteración tendrá la responsabilidad de asegurarse que los procesos planteados en este documento sean cumplidos.

# Gestión del alcance y planificación

Se creará un mapa de entregables, el cuál por cada iteración provee una lista de las cosas que el equipo estima que se podrán entregar. El mismo será actualizado de forma constante a lo largo del desarrollo del proyecto para reflejar cambios.

Un ejemplo del mismo podría ser:

|  |  |  |
| --- | --- | --- |
| Iteración 1 | Iteración 2 | Iteración 3 |
| 1 jun - 15 jun | 15 jun - 30 jun | 1 jul - 15 jul |
| Alta Usuarios | Baja Usuario | Modificación Usuario |
| Alta Producto | Baja Producto | Modificación Producto |

Se utilizará la plataforma YouTrack (<http://tallerproyectos2.myjetbrains.com/youtrack/>) para llevar cuenta del Project Backlog y el Sprint Backlog.

Dentro de las 24 hs. posteriores a la reunión de Sprint Planning, el equipo de trabajo enviará un e-mail al cliente con el Sprint Backlog definido en la reunión y la versión actualizada del mapa de entregables.

# Gestión de tiempos

Para la estimación de las tareas se utilizarán StoryPoints (<http://martinfowler.com/bliki/StoryPoint.html>) con la escala 1, 2, 4, 6, 8. Todos los miembros del equipo participarán de la estimación y cada BacklogItem (sea de proceso o producto y sea un feature o un bug) tendrá su estimación.

Está técnica nos permitirá estimar la velocidad del equipo en StoryPoints, la cuál será tenida en cuenta al momento de planificar cada iteración.

# Gestión de Riesgos

## Escala cuantitativa de impacto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Escala**  **Objetivo** | **0,05** | **0,1** | **0,2** | **0,4** | **0,8** |
| **Tiempo** | No significativo | <= 5% | 5% < X <= 10 % | 10% < X <= 20 % | > 20% |
| **Alcance** | No detectable | Pocas áreas afectadas | Gran cantidad de áreas afectadas | Inaceptable para el sponsor (negativo) | Proyecto inútil (negativo) |
| **Calidad** | No detectable | Funcionalidad específica | Requiere aprobación del sponsor (negativo) | Inaceptable para el sponsor (negativo) | Proyecto inútil (negativo) |

## Definición Probabilidades

|  |  |
| --- | --- |
| **Probabilidad** | **Descripción** |
| 0,1 | Difícilmente ocurra |
| 0,3 | Poco probable |
| 0,5 | Probable |
| 0,7 | Muy Probable |
| 0,9 | Casi seguro ocurre |

## Matriz de Probabilidad e Impacto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Probabilidad** |  |  |  |  |  |
| **0,9** | 0,045 | 0,09 | 0,18 | 0,36 | 0,72 |
| **0,7** | 0,035 | 0,07 | 0,14 | 0,28 | 0,5 |
| **0,5** | 0,025 | 0,05 | 0,1 | 0,2 | 0,4 |
| **0,3** | 0,015 | 0,03 | 0,06 | 0,12 | 0,24 |
| **0,1** | 0,005 | 0,01 | 0,02 | 0,04 | 0,08 |
|  | **0,05** | **0,1** | **0,2** | **0,4** | **0,8** |
|  | **Impacto** | | | | |

Los riesgos con exposición mayor a 0,25 tendrán un plan de contingencia. Los demás riesgos serán aceptados.

# Gestión de Cambios

## Alcance

Para introducir un cambio en el producto se debe tener en cuenta las siguientes consideraciones:

* La persona interesada en introducir un cambio puede
  1. Enviar un e-mail que incluya a la lista de distribución del proyectoy al cliente explicando el cambio que se desea introducir.
  2. Explicar el cambio en la reunión de Sprint Planning.
* Pueden proponerse cambios en el medio de una iteración. En este caso, el ProductOwner (teniendo en cuenta los criterios del equipo de desarrollo) tendrá que tomar la determinación de incluir al mismo en la iteración actual o agregar el ítem al Project Backlog para priorizarlo a futuro.
  1. Si se agrega en la iteración actual, se enviará un e-mail al cliente comunicando que ítems del Sprint Backlog quedaron fuera del alcance de la iteración. El e-mail contendrá además una imagen actualizada del Sprint Backlog.

## Tiempo

Es posible que durante el transcurso de una iteración, el equipo de trabajo se percate de que un ítem del Sprint Backlog tomará un tiempo diferente al estimado o que se necesitan tareas adicionales para completar una UserStory. En este caso:

1. El ProductOwner actualizará el Sprint Backlog agregando o removiendo tareas según sea el caso.
2. Se enviará un e-mail cliente con copia a la lista del proyecto explicando la razón del cambio.

En caso que el cambio afecte al mapa de entregables, el mismo también se actualizará y enviará al cliente.

## Otros Cambios

En caso de haber un cambio relacionado con alguna otra área, se enviará un e-mail al cliente con la acción propuesta por defecto pidiendo que responda de forma negativa en caso que la misma no sea de su satisfacción.

# Descubrimiento de requerimientos y especificación funcional

Los requerimientos se descubrirán en las reuniones con el cliente cada 2 semanas. En las mismas, el cliente podrá especificar nuevos requerimientos (o [cambios](#_Alcance)) y asignarles una prioridad.

El formato a utilizar para llevar cuenta de los requerimientos será el de UserStory. Cada UserStory debe tener:

* Título. No necesariamente debe tener el formato *"As a (role) I want (something) so that (benefit)"*. Esto queda a criterio del ProductOwner para cada UserStory.
* Prioridad. Definida por el cliente. La prioridad no necesita ser absoluta, puede ser relativa a las demás UserStories.

Cada UserStory puede tener:

* Criterios de aceptación. Solamente las UserStories con las que se va a trabajar en el Sprint deben tenerlo. No es necesario que todas las UserStories del Project Backlog tengan criterios de aceptación.

# Definición de pruebas, criterios de aceptación, registro de resultados y tratamiento de defectos

En cada iteración se crearán backlog ítems para las siguientes actividades:

1. [Definición de pruebas funcionales](#_Definición_de_pruebas)
2. [Regresión de pruebas funcionales](#_Regresión_de_pruebas)

El tiempo que consumen las mismas es similar al tiempo que consume cualquier otro ítem del backlog. Es decir, que estas tareas podrán ser estimadas por el equipo de trabajo y priorizadas por el cliente.

## Definición de pruebas funcionales

Por cada UserStory se determinarán un conjunto de pruebas las cuáles validan que la funcionalidad buscada por la misma (criterios de aceptación) se cumpla. Cada prueba funcional debe especificar:

1. Pasos a seguir.
2. Criterio de aceptación.

Las pruebas funcionales quedan a criterio del equipo de desarrollo.

## Regresión de pruebas funcionales

Implica hacer una pasada completa por todas las pruebas creadas a lo largo del proyecto. Sirve para verificar que todo lo que anteriormente estaba funcionando continua haciéndolo.

## Registro de resultados

Para el registro de los resultados se utilizará una planilla similar la siguiente (no necesariamente igual):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | 20-jun | Bug | 21-jun | Bug | 22-jun | Bug | 23-jun | Bug |
| 1 | Fallo | #127 | Paso |  | Paso |  | Fallo | #128 |
| 2 | Fallo | #220 | Fallo | #220 | Fallo | #220 | Paso |  |
| 3 | Paso |  | Paso |  | Fallo | #221 | Paso |  |

## Tratamiento de defectos

En caso que una prueba de regresión falle y no haya un bug cargado por la causa del fallo se deberá cargar un nuevo bug en el sistema YouTrack especificando:

1. Identificador del caso de prueba.
2. Pasos a seguir para reproducir el problema.
3. Severidad.
4. Estado: Open

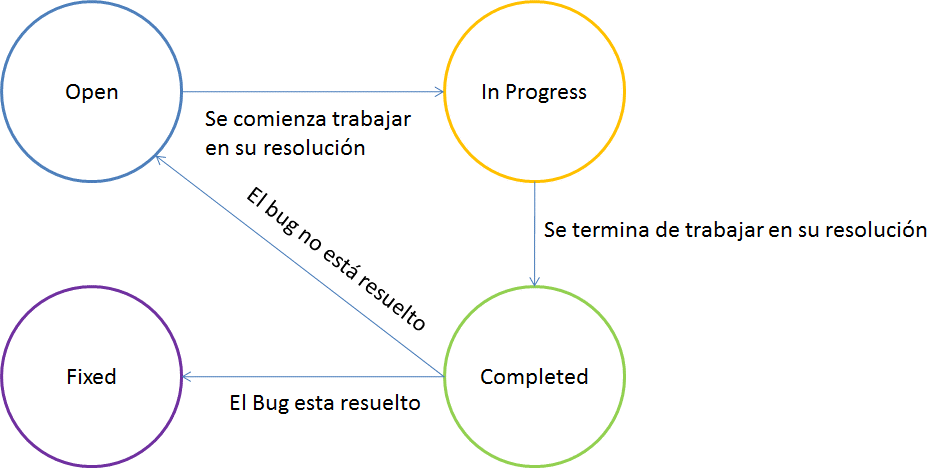
El ProductOwner puede decidir, según la funcionalidad que afecte el defecto y su severidad, si el mismo debe ser arreglado durante la iteración actual o se agregará al ProductBacklog. En este caso, debe comunicarse el cambio de plan siguiendo el procedimiento especificado en [Gestión de Cambios (tiempo)](#_Tiempo).

Es posible realizar pruebas exploratorias (exploratorytesting) para buscar bugs. En caso que un miembro del equipo de trabajo detecte un bug, puede agregarlo al Project Backlog con consentimiento del ProductOwner. De la misma forma si el cliente reporta un bug, el ProductOwner aprobará su adición al Project Backlog.

### Severidad

|  |  |
| --- | --- |
| **Severidad** | **Criterio** |
| **SEV 1 (Bloqueante)** | Afecta datos o funcionalidad crítica en un feature importante, afectando severamente a los usuarios, sin un workaround posible. |
| **SEV 2 (Severo)** | Afecta datos o funcionalidad crítica en un feature importante, y tiene un workaround complicado. En caso de pérdida o corrupción de datos el problema es difícil de detectar por el usuario. |
| **SEV 3 (Periférico)** | Afecta datos o funcionalidad no crítica en un feature importante, y tiene un workaround fácil. |
| **SEV 4 (Cosmético)** | Afecta la estética, o que la aplicación tenga una vista profesional. |

### Ciclo de Vida



# Mantenimiento de trazabilidad del proyecto

Se utilizará el sistema de versionado SVN. Cada uno de los mensajes de commit introducidos al sistema por los miembros del equipo debe incluir el número de backlogitem sistema YouTrack.

# Elementos utilizados para comunicar

A continuación se detallan los elementos utilizados para la comunicación entre el equipo de trabajo y el cliente.

## E-mail

El proyecto tendrá una lista de distribución en la cual estarán todos los miembros del equipo de trabajo.

## Reportes de progreso

Los días Miércoles y Sábado se enviará al cliente (con copia a la lista del proyecto) un e-mail que incluirá:

* El Sprint Backlog en su estado actual.
* El burndown chart de la iteración. En caso de que el equipo de trabajo lo considere necesario, puede ir acompañado de un análisis.
* Un resumen de los items del Sprint Backlog hasta el momento (ej: descripción, imágenes).
* Cuestiones relacionadas con cambios o decisiones que puedan involucrar al cliente.

En caso que uno de esos días el equipo no fuera a enviar un e-mail, deberá comunicarlo al cliente con anticipación.

Si los miembros del equipo de trabajo lo consideran útil, podrán también enviar este e-mail cualquier otro día.

## YouTrack

En cualquier momento el cliente tendrá la posibilidad de acceder a <http://tallerproyectos2.myjetbrains.com/youtrack/> para verificar el estado actual del backlog.

## Reuniones

Como se explicó en [Adaptación de Scrum](#_Adaptación_de_metodologías), el equipo de trabajo se reunirá con el cliente una vez cada dos semanas.

# Pruebas unitarias y elementos para registrar diseños

La lógica de negocio de la aplicación será desarrollada utilizando TDD. El código generado automáticamente por herramientas de desarrollo (como parte de templates) no necesariamente tendrá pruebas unitarias relacionadas.

|  |
| --- |
| **Importante:** Se aclara lógica de negocio por la imposibilidad de realizar test *unitarios* a componentes visuales, de integración con base de datos, etc. |

En lo referido al diseño, TDD es una técnica que permite diseñar las interfaces de los objetos y la interacción entre los mismos.

Adicionalmente, el equipo de desarrollo tiene la libertad de crear los elementos de diseño que considere útiles. Los mismos deben ser almacenados en el repositorio dentro de la carpeta **docs**(no necesariamente en el primer nivel). Él puede requerir elementos de diseño, determinando una prioridad para los mismos con respecto a los demás backlogitems. De esta forma, se trata al elemento de diseño como un item más del backlog.

# Seguimiento y Control del Proyecto

Se utilizará un burndown chart (provisto por la herramienta YouTrack) para llevar cuenta del progreso de la iteración en curso.

Debido a que el equipo trabajará principalmente los fines de semana y no con una cantidad fija de horas diarias, el ProductOwner deberá analizar al burndown chart en base a la proporción del tiempo total planeado que se ha dedicado a las tareas del Sprint Backlog.

# Especificación de atributos de calidad

Serán definidos a lo largo del proyecto con el cliente. El sitio <http://msdn.microsoft.com/en-us/library/ee658094.aspx> provee una lista de atributos comúnmente usados en proyectos de software.

En principio, se consideraran los siguientes atributos:

* **Mantenibilidad**
  + *Consideraciones*
    - Implementación de las dependencias entre capas mediante el uso de abstracciones.
    - Diseño de componentes coherentes y de bajo acoplamiento.
    - Automatización de pruebas.
  + *Medición*
* **Disponibilidad**
  + *Consideraciones*
    - Diseño de soporte de failover para tiers del sistema
    - Manejo adecuado de excepciones para reducir fallas de la aplicación
  + *Medición*
    - Porcentaje del tiempo de inactividad total del sistema durante un periodo predefinido.
* **Rendimiento**
  + *Consideraciones*
    - Uso de comunicación asíncrona
    - Elección de transacciones adecuada.
  + *Medición*
    - En términos de latencia (Tiempo necesario para responder a cualquier evento) o rendimiento (Numero de eventos que tienen lugar en una determinada cantidad de tiempo)
* **Escalabilidad**
  + *Consideraciones*
    - Manejo de picos de trafico y carga
    - Diseño de capas y niveles de escalabilidad
  + *Medición*
* **Testeabilidad**
  + *Consideraciones*
    - Uso de TDD
    - Sistema orientado de manera modular para soportar pruebas
  + *Medición*
    - Porcentaje de Cobertura de Código.