

Cátedra: Ingeniería de Software

Curso: 4k1

TRABAJO PRÁCTICO N°8

“SCRUM – Release and Sprint Planning – Planificación de
Release y de Sprint”

GRUPO 9

Integrantes:

Battig Chavez, Joaquin Antonio - 76056

Cosci, Franco - 78644

Folli, Nicolás – 78531

Pages Juan Ignacio, 78314

Reartes Alvaro Pablo, 75481

Slavik, Lucas - 78341

Docentes:

Meles, Judith

Boiero, Gerardo

Crespo, Mickaela

Fecha de entrega:

23/09/2021



ÍNDICE

| | |
|---------------------|-----------|
| ENUNCIADO | 3 |
| DESARROLLO | 4 |
| BIBLIOGRAFÍA | 10 |



ENUNCIADO

Objetivo:

Que el estudiante sea capaz de simular una de las ceremonias de SCRUM, Sprint Planning, cuyo propósito es la definición del Sprint Backlog.

Que comprenda la importancia de la planificación en el contexto de la gestión ágil de proyectos.

Consigna:

Tomando como base la definición de producto realizada para Taxi Mobile: Sistema web mobile para seguimiento de taxis y teniendo en cuenta el MVP definido, realizará con su equipo SCRUM la primera reunión de planificación de Sprint (Sprint Planning).

Instrucciones:

- En grupos trabajarán la consigna.
- Tomarán el MVP definido para el producto de Taxi Mobile: Sistema web mobile para seguimiento de taxis, ya estimado.
- Definirán las condiciones de contexto necesarias para la planificación del release.
- Definirán el Plan de Release indicando cuantos sprints serán necesarios y que user stories entregarán en cada uso y por consiguiente la duración del Plan de Release para la entrega de la versión del producto.
- Definirá la minuta para el Sprint 1 y el Sprint Backlog.



DESARROLLO

Planificación del Release Planning

Contexto para la planificación del release:

- Un Scrum Master
- Un Administrador de Base de Datos
- Dos Desarrolladores
- Dos Testers

La velocidad aproximada será de 8SP por Sprint.

Definición del Release - consideraciones del contexto

En el primer release se desarrollará el MVP. hemos decidido realizar 2 Sprints que tendrán una duración de 10 días hábiles cada uno con el objetivo de entregar valor al cliente en cada iteración al finalizar el sprint y para luego poder obtener una retroalimentación y poder aprender y validar que lo que venimos haciendo, es lo que el cliente espera; y si algo no es lo esperado, nos servirá para poder adaptarnos a esos cambios que pudieran surgir lo más rápido posible y evitar pérdidas de tiempo y dinero.

Al finalizar el sprint 1 ya tendríamos desarrollado el producto de software con lo mínimo necesario para poder validar la idea de negocio del mercado.

Luego, vendrá el segundo release el cual irá agregando nuevas funcionalidades que no fueron consideradas en el MVP (salvo Notificar a taxista solicitud de taxi y Buscar taxi cercano, las cuales todavía no fueron implementadas y forman parte del MVP).

La duración total del plan de release será de 20 días hábiles.

A continuación especificamos el Release Backlog:



| ID | Historia de Usuario | Rol | Story Points | Prioridad | Riesgo | Sprint | |
|----|---------------------------------------|-----------------------------|--------------|-----------|--------|----------|-----|
| 1 | Pedir taxi | Pasajero | 5 | 1 | Bajo | Sprint 1 | MVP |
| 2 | Loguear taxista | Taxista | 2 | 1 | Alto | Sprint 1 | |
| 3 | Notificar a taxista solicitud de taxi | Pasajero | 3 | 3 | Medio | Sprint 2 | |
| 4 | Liberar taxi | Taxista | 2 | 2 | Bajo | Sprint 1 | |
| 5 | Ocupar taxi | Pasajero | 2 | 1 | Bajo | Sprint 1 | |
| 6 | Buscar taxi cercano | Pasajero | 3 | 3 | Alto | Sprint 2 | |
| 7 | Ver ubicacion pasajero | Taxista | 5 | 1 | Alto | Sprint 1 | |
| 8 | Loguear pasajero | Pasajero | 2 | 3 | Medio | Sprint 2 | |
| 9 | Registrar central de taxi | Administrador de la central | 1 | 3 | Bajo | Sprint 2 | |
| 10 | Marcar taxi como fuera de servicio | Administrador de la central | 2 | 3 | Bajo | Sprint 2 | |
| 11 | Ver mapa de taxis | Administrador de la central | 5 | 2 | Medio | Sprint 2 | |

Estimaciones

- Estimación de horas por personas: 4hs
- Estimación de horas disponibles por iteración :
240 horas -> 4 hs * 6 integrantes * 10 días por iteración
- Historias de Usuario a entrar en la primer iteración:
 - Loguear Taxista (25hs)
 - Codificar Interfaz de usuario loguearse taxi: 4hs
 - Desarrollar Frontend Javascript: 3hs
 - Desarrollar Backend para logueo taxista: 5hs
 - Realizar pruebas de logueo: 2hs
 - Refactorizar código: 2hs
 - Realizar pruebas de usuario: 3hs
 - Realizar pruebas de regresión: 2hs
 - Realizar "push" de código de repositorio: 2hs
 - Generar documentación de la US: 2hs
 - Pedir taxi (54hs)
 - Acordar el estilo de interfaz con el PO: 5hs
 - Desarrollar módulo de comunicación de backend con sistema de geoposicionamiento: 8 hs
 - Solicitar credenciales y desarrollar el módulo que hará la integración con Google maps: 4hs
 - Diseñar interfaz: 5hs
 - Desarrollar código frontend: 4hs
 - Desarrollar código backend: 4hs



- Realizar pruebas de regresión: 4hs
- Realizar pruebas unitarias: 2hs
- Refactorizar código: 5hs
- Realizar pruebas de usuario: 3hs
- Diseñar y ejecutar casos de prueba: 5hs
- Realizar push de código al repositorio: 2hs
- Generar la documentación de la US: 3hs
- Liberar taxi (22hs)
 - Diseñar interfaz gráfica: 3hs
 - Desarrollar código para capa cliente: 4hs
 - Desarrollar código de lógica para cambio de estado de taxi: 4hs
 - Revision de codigo: 3 hs
 - Realizar pruebas unitarias: 2hs
 - Realizar pruebas de usuario: 2hs
 - Realizar push de código en repositorio: 1hs
 - Generar documentación: 3hs
- Ver ubicación del pasajero (38 hs)
 - Diseñar interfaz gráfica: 6 hs
 - Desarrollar código para capa lado cliente: 5hs
 - Desarrollar backend de comunicación y sincronización de ubicación: 5hs
 - Refactorizar código: 3hs
 - Realizar pruebas unitarias: 2hs
 - Realizar pruebas de regresión: 3hs
 - Realizar pruebas de usuario: 4hs
 - Generar documentación: 4hs
 - Revision de codigo: 5hs
 - Realizar push de código al repositorio: 1hs
- Ocupar taxi (23hs)
 - Diseñar interfaz gráfica: 3hs
 - Desarrollar código para capa lado cliente: 4hs
 - Desarrollar código de lógica para cambio de estado de taxi: 4hs



- Revisión de código: 3 hs
- Realizar pruebas unitarias: 3hs
- Realizar pruebas de usuario: 2hs
- Realizar push de código en repositorio: 1hs
- Generar documentación: 3hs
- Horas totales: 162 hs

Definición de Hecho (DoD)

- Todos los criterios de aceptación implementados y probados
- Tareas de desarrollo completas
 - Código inspeccionado (code review)
 - Maquetado que cumple con lineamientos
 - Funcionalidad validada
 - Integración con servicio correcto
 - Desarrollo en repositorio
 - Código desplegado en ambiente
- Tareas de testing completas:
 - Casos de prueba diseñados y revisados
 - Ciclos de prueba ejecutados y superados completos
 - Defectos validados
 - Casos de prueba automatizados
- Etapas de testing superadas:
 - Prueba de Integración realizada y superada
 - Prueba de Regresión realizada y superada
 - Sanity test pasado
- Bugs gestionados:
 - Documentados y priorizados correctamente
 - Resueltos y validados por el equipo
- Documentación actualizada:
 - Análisis funcional completo y validado por el PO
 - Historias de usuario completas y validadas por el PO
- Aprobación por parte del PO.



Minuta del sprint Planning

Sprint Nro. 1

Duración del Sprint en días: de 1 a 14 días (10 días hábiles)

Objetivo del Sprint

- ❖ Realizar la implementación para el Product Owner de las funcionalidades de la aplicación relacionadas con la solicitud de un taxi a través de un mapa interactivo, permitirle al taxista ver la ubicación de su pasajero y establecer el estado actual de su vehículo en OCUPADO o LIBRE.

Equipo Scrum:

- Battig Chavez, Joaquin Antonio
- Cosci, Franco
- Folli, Nicolás
- Pages Juan Ignacio
- Reartes Alvaro Pablo
- Slavik, Lucas

Capacidad del Equipo en Horas Ideales:

$$6 \text{ integ} * 10 \text{ días hábiles} * 4 \text{ horas} = 240 \text{ horas}$$

Contexto

Particularmente en este sprint, un integrante del equipo solicitó vacaciones, por lo que la capacidad se ve reducida. A continuación, un detalle de la misma:

| Nombre | Perfil | Horas diarias | Días disponibles | Horas en eventos scrum | Horas en Sprint | Horas efectivas |
|----------------|------------------|---------------|------------------|------------------------|-----------------|-----------------|
| Alvaro Reartes | Scrum Master | 4 | 9 | 10 | 36 | 26 |
| Lucas Slavik | Desarrollador | 4 | 10 | 10 | 40 | 30 |
| Joaquin Battig | Administrador BD | 3 | 9 | 10 | 27 | 17 |
| Juan Pages | Tester | 5 | 9 | 10 | 45 | 35 |
| Nicolás Folli | Tester | 4 | 9 | 10 | 36 | 26 |
| Franco Cosci | Desarrollador | 4 | 8 | 10 | 32 | 22 |



Alcance del Sprint

| Historia de Usuario | Tarea | Tipo | Estimación horas | Responsable |
|---------------------|-----------------------------------|------|------------------|----------------|
| Loguear taxista | Maquetado | DEV | 1 | Lucas Slavik |
| Loguear taxista | Implementación de Funcionalidad | DEV | 2 | Lucas Slavik |
| Loguear taxista | Integración con Servicios | DEV | 1 | Franco Cosci |
| Loguear taxista | Desarrollo SP | DBA | 4 | Joaquin Battig |
| Loguear taxista | Diseño de Caso de Prueba | QA | 2 | Juan Pages |
| Loguear taxista | Revisión de Caso de Prueba | QA | 1 | Nicolás Folli |
| Loguear taxista | Revisión estructurada | DEV | 1 | Franco Cosci |
| Loguear taxista | Ejecución de Caso de Prueba | QA | 2 | Nicolás Folli |
| Loguear taxista | Corrección de Defectos | DEV | 1 | Franco Cosci |
| Loguear taxista | Validación de Defectos | QA | 1 | Nicolás Folli |
| Loguear taxista | Automatización de Casos de Prueba | QA | 1 | Juan Pages |

| Historia de Usuario | Tarea | Tipo | Estimación horas | Responsable |
|---------------------|-----------------------------------|------|------------------|---------------|
| Pedir taxi | Maquetado | DEV | 4 | Lucas Slavik |
| Pedir taxi | Implementación de Funcionalidad | DEV | 6 | Lucas Slavik |
| Pedir taxi | Integración con Servicios | DEV | 6 | Franco Cosci |
| Pedir taxi | Diseño de Caso de Prueba | QA | 3 | Juan Pages |
| Pedir taxi | Revisión de Caso de Prueba | QA | 2 | Nicolás Folli |
| Pedir taxi | Revisión estructurada | DEV | 1 | Franco Cosci |
| Pedir taxi | Ejecución de Caso de Prueba | QA | 5 | Nicolás Folli |
| Pedir taxi | Corrección de Defectos | DEV | 2 | Franco Cosci |
| Pedir taxi | Validación de Defectos | QA | 2 | Nicolás Folli |
| Pedir taxi | Automatización de Casos de Prueba | QA | 8 | Juan Pages |



BIBLIOGRAFÍA

- Apunte de cátedra.