**Resumen de la primera reunión 28-08**

Estimados, este resumen de la reunión es mucho más útil si antes leen los documentos que envió Camilo. Ya están hechos todos los requerimientos, los módulo y esas cosas asique leanlos por favor.  **Clientes**

Son Benedetto y Rizzo, alumnos del profesor Neyem. Tenemos a los clientes en el DCC y a nuestro tutor Hans Findel a 4 metros de distancia asi que aprovechemos.

**Definición resumen del proyecto**

Task Management (tipo Kanbanery), es un administrador de proyectos (e-tracking). El usuario debe ser capaz de crear proyectos en la aplicación, crear tareas que sean asignables a los usuarios y tengan referencia con git. Los administradores debe poner las tareas y despues se asignan a los integrantes, deben haber comentarios, pedir ayuda, tener estados (en desarrollo, testing, todo dinámico,etc). Todo debe ser configurable por el administrador.

**Situación Actual**

**WEB**

Nicolas Rizzo ya hizo la parte web, está hecho el Kanban, con comentarios, con Github, con Dropbox y muchas otras pero hay que hacer Pollishing. Debemos hacer que muchas funcionalidades se hagan con AJAX para evitar que se recarguen todas las páginas, entre otras cosas. Esta todo lo de Casos de Uso que permite crear requisitos y a medida que se van realizando las tareas se va completando. SCRUM y Extreme Programming, se crean templates que se completan con la información de la literatura y los desarrolladores completarán estos templates para crear tareas. Un caso de uso puede tener un número de tareas asociadas de kanban y va mostrando el pocentaje de avance.

Hans (nuestro tutor) nos dijo que vamos a tener que tener todo programado usando los “module” de Rails asique atentos a eso cuando estemos aprendiendo.

En general nos pidió 5 cosas (ordenadas por importancia)

1) Módulo de Evaluación (el más importante)

- Módulo de Evaluación: 360 reviewing. - Todos pueden evaluar el trabajo de los miembros del equipo. Usando Template de Evaluación. A idea es que se puedan poner deadlines. Podrían ingresarse puntajes, comentarios (cuantitativo y cualitativos). Criterios específicos las puede ingresar cada empresa. Analizar Excels que envía Neyem, el template del curso es lo mínimo.

2) Modularización (debemos ser conscientes mientras programamos)

- Todas las funcionalidades específicas a un tipo de proyecto deben ser prorgamadas en módulos. Estos podrán ser activables o desactivables a la hora de crear un nuevo proyecto. La idea es que eventualmente se podrían programar módulos para proyecto mineros, de construcción, etc.

3) Calendarización de Tareas

- Se deben definir Milestone. Si yo soy administrador debo poder ver el hito y ver el porcentaje de avance. Como un release, la idea es que no sea una pequeña tarea. Los programadores deben ir registrando su horas hombres y sus avances, milestones = puntos de control del proyecto. Las tareas pueden estar hechas o no hechas. La decisión de cómo se marcan las tareas hechas o no hechas queda para nosotros

- Eventos: reunión

4) Polishing

- Se debe poder agregar comentarios, chat, bloqueos, todo lo de kanbaney

- Se debe poder asociar código a cada tarea.

- Nueva interfaz para módulo de Casos de Uso

- Está sólo la base de estadísticas de commits se deben agregar más funcionalidades

- Agregar niveles de jerarquías (Organizaciones antes de crear proyectos)

- Se deben crear prerequisitos

- Extender la integración de Dropbox con SkyDrive y OneDrive

5) Testing

- A través del proyecto ojalá creemos buenos test de todo lo que hagamos. Es importante mantener bloques bien testeados y cohesionados.

**Android**

- Objetivo: mejorar reuniones de clientes

- Local Axis Note 3 S-Pen

- Hay algo hecho pero se permite empezar desde 0.

- Mala tecnología usada en Wi-fi

- Las personas deben poder enlazar sus dispositivos en las reuniones para que puedan hacer diagramas para organizarse.

- Video: Sansung Galaxy Note PRO E-Meeting

- Hay módulo de documentos donde deberían subirse esos documentos (primera meta es el bitmap, read-only)

- REALIABILITY. Que pasa si mi conexión se pierde (para el sprint 1)

- El usuario de la parte mobile: los clientes

- Se exige Android Nativo – API Level 17 → 19

- Secondary Display: agregar conecxión a pantallas más grandes (ChromeCast)

- No queremos Mirroring, queremos algo más enrriquecido.

- Diagramas como las paletas de Creatly y de Gliffy. Partir con Casos de Uso, agregar fotos del telefono

- Objetivo: es levantar requisitos con los clientes y entre desarrolladores

- Librería NIO

- Tal vez hay que botarlo todo pero se puede usar la base