

FEEDBACK GRUPOS

- Contenido

- La idea del producto debe estar incluida, pero más breve que anteriormente.
- Mostrar aplicación funcionando al final del sprint con lo que habíamos planificado, además de **desplegarlo y asignarle una URL**.
- Mostrar **quién** ha realizado la tarea y **qué** ha hecho. Con ello, ¿está completado o no?. También se debe añadir la dedicación de cada miembro en la tarea (horas).
- Indicar quiénes son los clientes y cómo nos hemos acercado a ellos. **Plan piloto**.
- Incluir un enlace ya funcional con lo que se dice que va a estar listo para el sprint.
- Indicar qué porcentaje de avance llevan las tareas.
- Indicar **qué** problemas hemos encontrado y **cómo** lo hemos solucionado. Apoyarlo con métricas.
- Incluir el **rendimiento de los miembros** del equipo. Reflejar quién ha tenido un mejor rendimiento, peor, etc. Quién ha cumplido y quién no.
- No poner qué feedback hemos hecho en la presentación.
- Casos de uso y su avance respecto al sprint y el tiempo.
- Realizar gráfica burndown.
- El **análisis de riesgos de las tecnologías** varía de una persona a otra. Quiénes lo tienen que aprender, de dónde parten, a donde queremos llegar, cuánto van a tardar. Hacerlo por cada miembro del grupo o por cada tecnología.
- No poner mockups, poner mejor lo que ya está hecho.
- Las horas deben ser las justas, y adaptar la planificación a esas horas.
- Estimar las horas y decir que ha pasado si no se han llegado a las horas o se han superado. Indicar problemas y soluciones.

- Presentación

- Especificar riesgos de aprendizaje.
- No tardar demasiado en presentar la idea.
- No contar lo que ha hecho cada miembro del equipo en la propia presentación de los miembros.
- Tienen que quedar claros los bloques de la presentación: **qué hemos hecho, qué tenemos, qué está por realizarse**.
- No romper el inicio efectivo, y menos con el feedback.
- Hay que transmitir seguridad a la hora de mostrar el trabajo que se lleva realizado, esté o no implementado.
- Si se es ingeniero software, no puede ser de marketing ni secretario.
- Incluir las actividades de todos los miembros del grupo en una sección aparte.

- **Estilo**

- Las fotografías del equipo deben ser simétricas.
- El blanco sobre el verde no se ve nada.
- Las letras de los mockups y las capturas de pantalla se deben ver claramente.
- Añadir quién está logado cuando sea necesario en los casos de uso.
- Usar o inglés o español pero no los dos. Preferiblemente inglés.
- Las letras deben ser grandes, que se vean. Transparencia en la que no se vea algo **no apto**.
- Cuidado con las pedofotos. ¿?

- **Negocio**

- El cliente puede pedir cambiar funcionalidades al final del sprint, hay que prevenirlo.

- **Miscelánea**

- Lo que se dice que se va a hacer en el sprint tiene que estar hecho **si o si**.
- Las tecnologías tienen que estar aprendidas **desde ya**.
- Auth0 para el login.

FEEDBACK CLIMBCODE

- Contenido

- En los riesgos de las tecnologías/aprendizaje por personas, añadir más detalles
- Las estimaciones mejor y peor están realizadas en base a mucha incertidumbre, no se puede expresar en referencia a sprints, no puede existir tanta duda en ellos. Además, no tienen sentido en este entregable.
- Las estimaciones de tiempos son muy divergentes. En el caso de que existan divergencias, definir el problema y su solución.

- Presentación

- Las dos frases iniciales no tienen continuidad una con otra, enlazarlas mejor.
Pero esa es la idea, son cosas distintas que unimos.

- Miscelánea

- Tener en cuenta el CrossSite Scripting (XSS) .
- Especificar roles como si fueran responsabilidades (Backend/frontend por ejemplo).
- “Contestar bien” a los compañeros: no decir que algo está presentado así porque lo dijo el profesor X. Se acepte o no el feedback de un profesor o de un compañero de clase,justificarlo.
- Sugerencia: Tabla visual de elementos terminados/WIP/no empezados.

FEEDBACK GENERAL (CONCLUSIONES)

- Moderación/varios

- Realizar la siguiente encuesta si se desea: goo.gl/bcwiBG
- Tener en cuenta los feedback de organización de la clase que los profesores proponen.
- Dejar más tiempo a las conclusiones finales, 20 minutos no es suficiente.

- Contenido

- No mezclar inglés con español. Todo en un idioma, preferiblemente en inglés, incluido documentación en el entregable así como la presentación. **Todo.**
- Se tiene que ver claramente la lista de personas en el entregable. Identificar mejor al grupo.
- Dar comienzo efectivo e idea (tiene que durar muy poco). Asignar tiempo para ello.
- Las presentaciones tienen que estar enfocadas en el ámbito del proyecto. No hay que parar de trabajar en las cosas que no se vayan a presentar.
- Cuidado con los costes, no un estudio detallado de los costes pero sí que se sepan identificar los costes directos, indirectos, etc.
- El feedback no tiene por qué salir en la presentación. Puede considerarse para un futuro.
- Debe de haber una **demo** de lo más importante (despliegue en remoto), hacerla de lo que nos dé el hecho diferencial. Además, **URL con el proyecto desplegado.**
- Por cada problema (**deben existir problemas**) dar una solución, mejora, métrica y resultado de la mejora. Ver si la solución ha surgido efecto.
- Hay que decir por cada miembro del grupo el trabajo que se realiza, así como las horas que ha invertido y si ha cumplido o no.
- Ser honestos con los tiempos del trabajo de los miembros del grupo.
- Centrarse más en las issues que en las horas (poner las dos cosas)
¿Cuántas tareas se le han asignado y ha completado el miembro del grupo?
Crear gráficas burndown para realizar el seguimiento de las mismas.
- Sugerencia: gráfica de las tareas que se están llevando con el coste que está teniendo.
- Las tarjetas amarillas para grupos en concreto, son tarjetas para todos los grupos, ya que son cosas que se suelen repetir varias veces, hay que tener en cuenta todos los feedbacks.
- Crear un plan de pilotaje. El primer día del sprint 1 tiene que existir un plan para poner la aplicación en un usuario real (intentar que no sean familiares). Pilotar lo antes posible. (Plan para poner en uso con clientes reales)
- Centrarnos en caso de uso estrella y pilotaje.
- Hablar sobre el despliegue, tener claro donde y como se despliega. Incluirlo en las tecnologías.
- **Centrarnos en Sprint**, no en el MVP.
- Hay que transformar y quitar peso a la parte del modelo del negocio y centrarse en los sprint.

- Mencionar brevemente la arquitectura tecnológica.
- Justificar trabajo de cada persona.
- Especificar más que roles del equipo, responsabilidades.
- Presentación: Idea con inicio efectivo muy breve y el peso principal es sprints, ver que está haciendo cada uno, porcentaje de desviación, demostración de lo que se tiene, y brevemente lo que se va a hacer en el sprint 2 y 3.
- Justificar la divergencia de horas en caso de que exista. Si la hay se va a preguntar y hay que explicarla.
- Optimistas y pesimistas no tiene sentido en este sprint.
- Internamente debemos mirar el riesgo de los aprendizajes.
- Usar checklist para las tareas que se van a realizar y comprobarlas al final de la semana.
- Team building. Planificar actividad para hacer en grupo. Hemos quedado un día para ir a tomar una cerveza, para ir senderismo, etc. Decir cómo estamos fomentando el team building en las horas imaginarias.

- **Presentación**

- Echarle un vistazo a Zendesk para ver ejemplos de videos.
- Para conocer algunos consejos para presentar en público: Julian Treasure, How to talk so that people want to listen.
- No pueden existir transparencias que no se vean (letras grandes, fondos claros, etc.), si ocurre, no apto.
- Todo el grupo debe estar en clase (a excepción de justificación) ya que a veces se les pide justificar algo a algún miembro del grupo y no están en clase. Si no está la persona cuando se le pregunte, es no apto para todo el grupo.
- Unir presentación del equipo con lo que ha hecho el equipo sólo si van una cosa detrás de otra.
- Especificar más que roles del equipo, responsabilidades.
- Pasado, presente y futuro en la estructura de la presentación. No hablar de algo futuro cuando en realidad es pasado.
- Verificar que la presentación contiene todo el feedback.

- **Tareas a realizar**

- Desarrollo elementos
- Gráficas y métricas
- Pilotaje -> Plan para llevarlo a usuarios
- Despliegue, Travis, Heroku
- Definir lógica