Climbcode

- En la demo debería verse SIEMPRE el usuario logado aunque se diga debe aparecer. Hacer la demo algo menos rápida, pero no mucho porque se aburre.
- Controlar los tiempos en la demo. Álvaro habla demasiado rápido (porque la demo lo es).
- EFECTIVO, pero hay otro mejores(muller jaja)
- Contar las ideas que están en el aire, como los cursos de formación. Apuntar en el feedback del cliente que no interesa. Si nos lo demandan, aparenta que no están muy asentadas. Debe ser algo claro del pilotaje: lo demandan o no. Si lo demandan, ¿se va a hacer?
- El amarillo de las tecnologías molesta,
- Riesgo de Heroku hay que cuantificar el riesgo.
- Le ha gustado como hemos planteado las soluciones, métricas y resultados. Falta indicar si se considera suficientemente buena (es aceptable la mejora o se necesita más, indicar cuál es el rango de que se considera aceptable) la solución.
- A la hora de estimar: analizar la mala estimación y buscar soluciones. No le ha gustado que estimemos entre todos.(obtener métrica sobre la estimación)
- Añadir en la 43 la estimación ideal del coste. El coste en vez de bajar, sube.
- Última diapositiva cambiar “éxito” por felicitación al propio equipo (“vamos bien”).
- Plan de contingencia/castigo para los problemas ocasionados por los miembros del grupo.
- Último sprint de desarrollo: hay que priorizar lo que se va a completar y lo que pidan los clientes (priorizar tareas) e indicarlo por si se queda algo sin entregar, que sea lo menos prioritario (y enfade menos al cliente).

Feedback Grupos

Presentación

- Idea para la gráfica del % de error de cada miembro:
  - Leyendas a los lados (%)
  - Cada barra un miembro del equipo.
  - Sobre cada barra una foto en miniatura del miembro.
- En la demo mostrar datos realistas.
- Plan B si la presentación no va.
- Explicar bien las métricas de los problemas.
- Todo lo que quite protagonismo a la demo, quitarlo de la demo.

Contenido

- Estudio de la calidad del código (SonarQ o alternativas).
- Sugerencia: Gráficos de % de diferencia entre horas con fotos
- Establecer un rango de lo que se considera aceptable para cada métrica.
- Sugerencia: Gráfica Burnup/Burndown en forma de X: Los SP bajan, el presupuesto(gastado) sube.
- Poner explícitamente la fórmula del rendimiento

- Cuánto van a pagar y por qué al preguntar en el pilotaje.
- Unificar todas las gráficas que comparen cosas del grupo. Si se compara algo de todo el grupo, todos tienen que estar en la misma gráfica.
- Una tarea del sprint 3 tiene que ser aplicar el feedback del pilotaje.
- Medir el impacto global de la métrica y la solución del problema.
- El pilotaje se hace teniendo algo que mostrar.
- Los problemas no pueden ser ambiguos ni difusos, tienen que quedar claros y ser concretos y precisos.
- Para el próximo sprint: Pilotaje hecho y haciéndose, número creciente de posibles clientes y usuarios que han probado la app, haber recogido el feedback y qué vamos a hacer con él. Mínimo una incorporación del feedback del pilotaje en el proyecto y resultado al mostrárselo a los usuarios.
- Hay que decir por cojones si están de acuerdo con lo que van a pagar, y si no, cuánto están dispuestos. Si no lo sabemos, preguntar cuánto están dispuestos. No darles opciones, sino dejar que ellos digan lo que piensan.
- Si no usamos la base de conocimiento de la clase, la cagamos. Tiene que poder accederse rápidamente para responder lo que pregunten. Todo el mundo tiene que estar listo para lo que le pregunten.
- Gráfica Burnup/Burndown en forma de X: Los SP bajan, el presupuesto(gastado) sube. O lo ponemos o vamos mal.
- Si la demo no funciona, no apto.
- El sistema de la demo tiene que ser estable para probarlo en cualquier momento. Si no, es no apto.
-

Off-topic

- ¿Nepe? Ofc yes