Retrospective Starfish

Estampa.la

Más De

1. **Utilizar la Wiki como centro de documentos.**

   Ejemplo: En la wiki se encuentran todos los documentos actualizados asociados al proyecto.

2. **Continuar usando buenas prácticas para la definición de las APIs**

   Ejemplo: Todos los servicios se definieron de forma uniforme, estandar y predecibles.

Más De

---

3. Continuar diseñando antes de programar.

Ejemplo: Solamente cuando estamos de acuerdo en el diseño implementamos.

4. Dividir responsabilidades específicas entre front y back.

Ejemplo: se establecieron contratos específicos con los servicios lo que permitió avanzar en la implementación de forma paralela de front y back.

Comenzar a hacer

---

1. Pruebas funcionales cuando se crean o modifican features.

Ejemplo: cuando se crean features nuevos se encontraban partes que no funcionaban, lo que atrasaba a los demás integrantes del equipo que dependían del feature.

2. Seguimiento a las actividades y Responsabilidades que tiene asignado cada integrante.

Ejemplo: Cuando se asignan las actividades no se hace un seguimiento sobre los avances o impedimentos que se han tenido, lo cual afecta el cronograma cuando se presentan errores y se tiene poco tiempo para reaccionar.

Comenzar a hacer

---

3. Refactorización del código.

Ejemplo: con el tiempo el código ha ido creciendo y algunos módulos presentan problemas de rendimiento ya que los diseños iniciales fueron evolucionando y algunas partes del código quedaron rezagadas.

4. Establecer criterios de aceptación de los features que se Desarrollan.

Ejemplo: se debe tener claro cuando un feature está terminado y está listo para ser probado.

Menos De

---

1. **Cambiar la arquitectura o diseños.**

   Ejemplo: la arquitectura ya se encuentra estable y debemos evitar cambiarla.

2. **Menos trasnochos para cumplir entregas.**

   Ejemplo: si se mejora la organización no se requerirá trasnochar para cumplir con entregas.

Menos De

---

3. Menos pruebas de tecnologías nuevas o librerías.

Ejemplo: Ya se llegó a una arquitectura estable con tecnologías definidas y probadas, lo que disminuye la incertidumbre sobre el desarrollo de los features faltantes.

4. Cambios sin notificar.

Ejemplo: Una vez se decide en grupo el funcionamiento de un feature, éste no se debe modificar sin consultarlo con los demás integrantes del equipo. Estos cambios sin previo aviso provocan inconsistencias o incongruencias al momento de entender el funcionamiento del sistema.

Dejar de hacer

---

1. **Dejar la preparación del del demo para lo último.**

   Ejemplo: El demo no se está preparando con el tiempo requerido, lo que provoca contradicciones al momento de presentar.

2. **Modificaciones al código el día de la presentación.**

   Ejemplo: Modificaciones en caliente al código el día de la presentación genera efectos de borde ya que ese código no se prueba.

Dejar de hacer

---

3. Perder el toco de lo importante.

Ejemplo: Se dejó de lado lo funcional por preocuparnos de problemas técnicos y de arquitectura de muy bajo nivel que es difícil plasmarlo en una presentación.

4. Perder el ritmo de trabajo después de una entrega.

Ejemplo: Después de cada entrega se baja el ritmo, teniendo una falsa sensación de haber terminado. Se debe mantener siempre un ritmo constante y equilibrado.