

LEAN SOFTWARE DEVELOPMENT

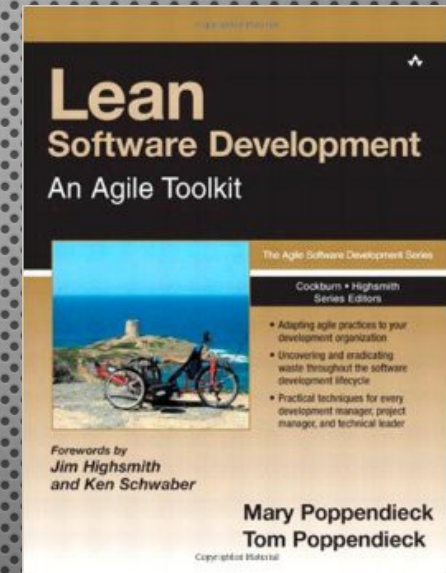
Análisis y diseño de software 1.



FREDDY ABAD
RONALDO RENDON

LEAN SOFTWARE DEVELOPMENT

- Método de desarrollo de software.
- Marco Teórico Sólido, basado en la experiencia, para las prácticas ágiles de gestión.
- Se origino en el Sistema de Producción de Toyota.
- Lean significa «fino» o «esbelto».
- Lean mejora la calidad y la eficiencia por la eliminación de residuos, minimizar capital



LA DISCIPLINA DE LAS 5S

Para lean software development

- **SORT** (EN CASO DE DUDA , A LA BASURA)
- **STRAIGHTEN** (ETIQUETADO , CODIFICACIÓN DE COLORES , PRESENTACIÓN)
- **SHINE** (MANTENER LA LIMPIEZA EN TODO PROCESO, SUSTITUIR TODO LO NO OPERATIVO).
- **STANDARDIZE** (CREAR DISEÑOS , ESCRIBIR LISTAS DE CONTROL)
- **SUSTAIN** (CREAR UN NUEVO HÁBITO DE TRABAJO , COMPROBAR REGULARMENTE LOS PROCESOS)

5S LEAN MANUFACTURING

seiri: clasificar

seiton: sistematizar

seiso: limpiar

seiketsu: estandarizar

shitsuke: disciplinar

Kaizen

- kaizen = "mejora continua".
- un evento de kaizen significa crear un cambio en un proceso/herramienta que desembocará en una mejora.
- Los cambios son mejores cuando son creados por la persona que realiza el trabajo.
- Se utiliza el sentido común y la intuición.



PRINCIPIOS del lean software development



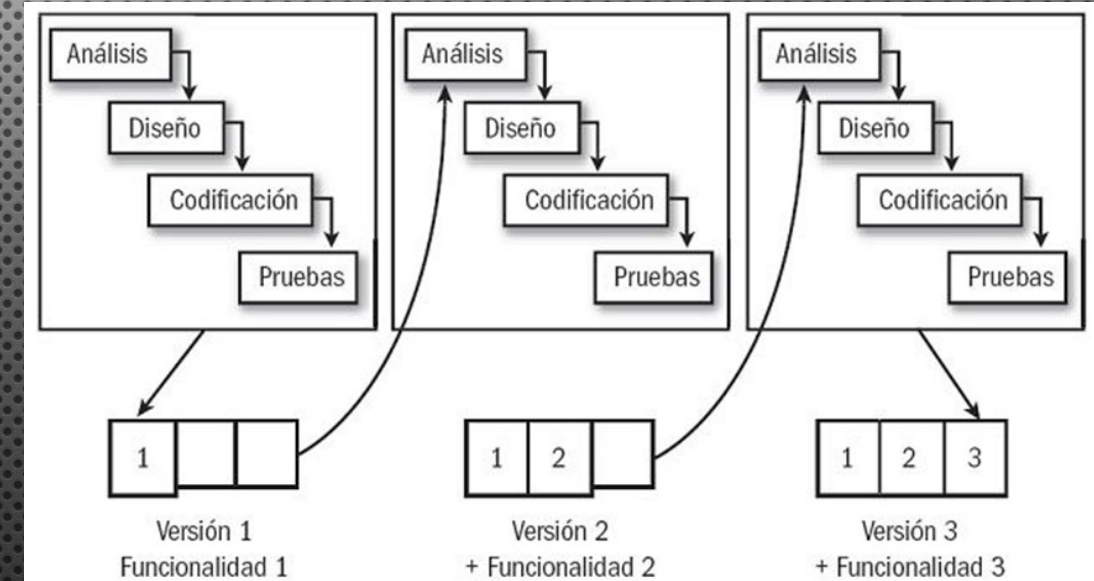
- I. Eliminar perdidas
- II. Aumentar el feedback
- III. Decidir lo mas tarde posible
- IV. Entregar rápidamente
- V. Crear integridad
- VI. Potenciar el equipo
- VII. Optimizar el todo

Eliminar las pérdidas

- CREAR VALOR PARA EL CONSUMIDOR.
- DESECHAR LO INNECESARIO/ POCO VALOR PARA EL CONSUMIDOR.
- EL CONSUMIDOR ESTARÍA IGUAL DE CONTENTO CON EL SOFTWARE SIN ESA CARACTERÍSTICA.

Aumentar el feedback

PARA PROBLEMAS DIFÍCILES ,
AUMENTAR LA RETROALIMENTACIÓN.
USAR UN MODELO ITERATIVO
INCREMENTAL.
“REPEAT STEPS UNTIL REQUIREMENTS
ARE MET”.



Decidir lo más tarde posible

- LA TECNOLOGÍA CAMBIA RÁPIDAMENTE.
- LA SITUACIÓN DEL NEGOCIO EVOLUCIONA.
- ¡EL SOFTWARE CAMBIARÁ!
- TOMAR DECISIONES BASÁNDOSE EN HECHOS/EXPERIENCIAS, NO EN PREDICCIONES.

ENTREGAR RAPIDAMENTE

- BASAR LA PRODUCCIÓN EN LA DEMANDA, NO EN UNA PLANIFICACIÓN.
- REALIZAR UN TRABAJO FÁCIL DE ENTENDER.
- DIVIDIR EL PROBLEMA EN PROBLEMAS PEQUEÑOS A RESOLVER.
- MINIMIZAR LA DEMANDA POR ERRORES, LOS RESIDUOS DE LA SOBRECARGA.
- MAXIMIZAR EL VALOR EVITANDO LOS COSTOS POR RETRASO Y POR APLAZAMIENTO.
- BUENA COMUNICACIÓN:
 - REUNIONES .
 - KANBAN .

Crear integridad

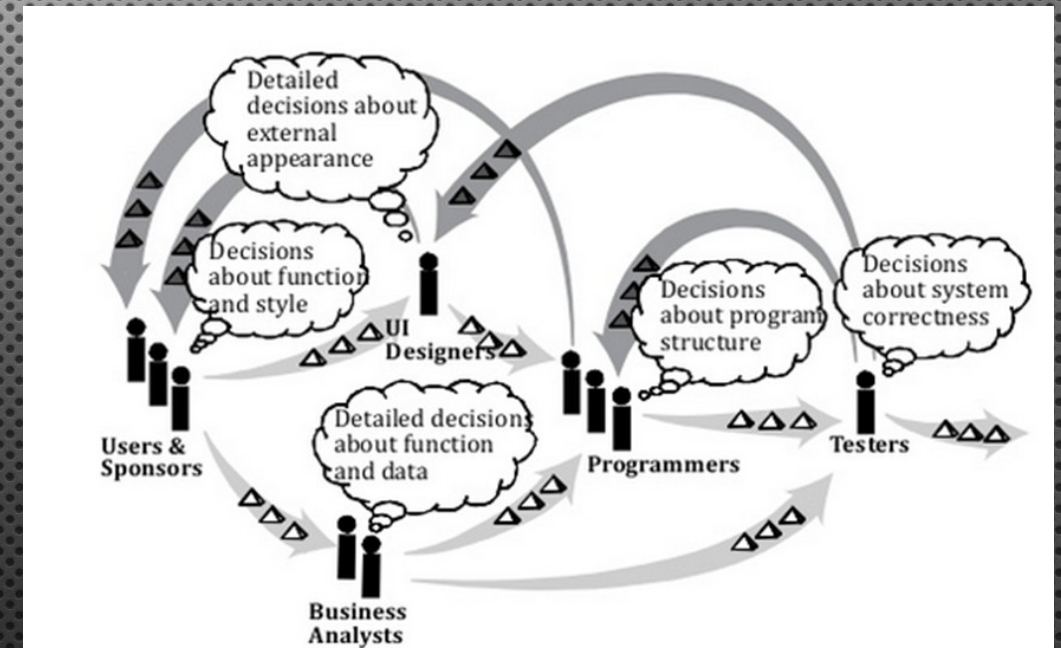
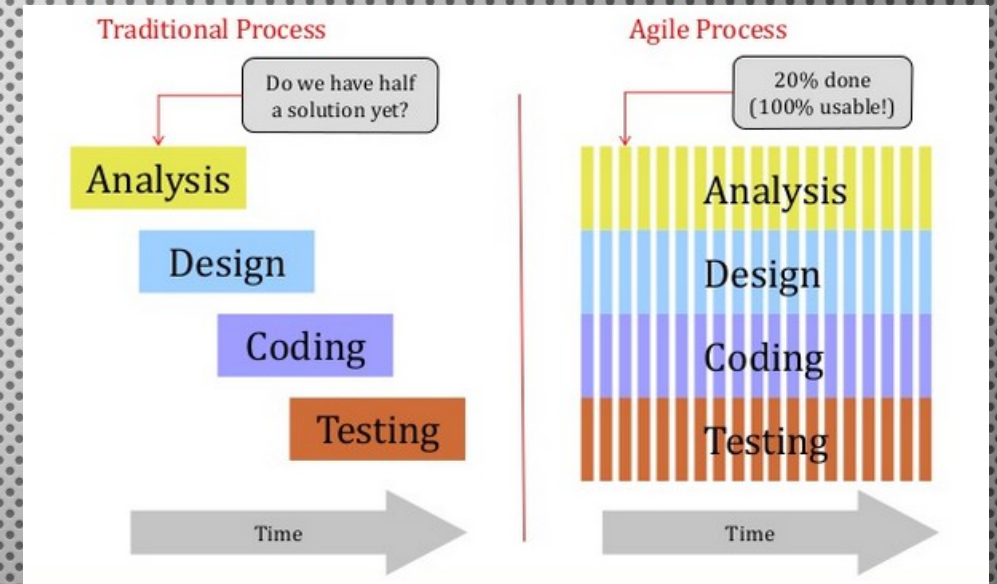
- SISTEMA TIENE INTEGRIDAD CUANDO SE LE DA AL CLIENTE EXACTAMENTE LO QUE QUIERE.
- ENTENDER LOS PROBLEMAS, DOMINARLOS Y RESOLVERLOS AL MISMO TIEMPO
- PEQUEÑOS ACTOS DE LIDERAZGO DE LOS TRABAJADORES GENERAN CASCADAS DE MEJORAS.
- “TRABAJADORES DE CONOCIMIENTO” = LOS TRABAJADORES TIENEN LA FACULTAD DE ORGANIZARSE POR SU CUENTA PARA COMPLETAR EL TRABAJO Y LOGRAR RESULTADOS.
- GENERAR MANERAS DE VISUALIZAR EL FLUJO DEL PROCESO
- CREAR DIRECTIVAS DEL PROCESO EXPLÍCITAS PARA TODO LOS PARTICIPES DEL PROYECTO.
- BENEFICIO: LAS CONVERSACIONES SOBRE POSIBLES MEJORAS PODRÁN SER COLABORATIVAS Y OBJETIVAS.

POTENCIAR EL EQUIPO

- NADIE ENTIENDE MEJOR LOS DETALLES QUE LAS PERSONAS QUE TRABAJAN EN ELLO.
- ENCONTRAR PERSONAS QUE HAGAN BIEN SU TRABAJO Y DEJARLOS ACTUAR.
- EQUIPO PEQUEÑO.
- COMPROMISO DEL EQUIPO DE TRABAJO.
- MISION CLARA / CORTO PLAZO .
- PERSONAL EXPERTO Y CON FACULTADES DE DOMINAR.
- RECOGER INFORMACION PARA DETERMINAR LA VIABILIDAD.
- PROPORCIONAR LOS RECURSOS NECESARIOS.
- LIBERTAD PARA TOMAR DECISIONES.
- MANTENER ESTANDARES DE CODIFICACION , CONTROL DE VERSIONES, TEST AUTOMATICOS.

Optimizar el todo

- ENFOQUE EN RENDIMIENTO GENERAL DEL SISTEMA.
- FOCUS ON THE TEAM EFFORT, NOT AS A GROUP OF INDIVIDUALS.
- THINK BIG, ACT SMALL, FAIL FAST, LEARN QUICKLY.
- SISTEMAS DE SOFTWARE= PARTES +INTERACCIONES.
- ADAPTACIÓN BASADA EN DATOS EMPÍRICOS NO ASUNCIÓN.
- DEFECTOS SE ACUMULAN EN EL PROCESO DE DESARROLLO.
- “EMBRACE MISTAKES THEN LEARN FROM THEM”.
- TENER BIEN DEFINIDAS LAS RELACIONES.
- ÉXITO=PRINCIPIOS LEAN + SENTIDO COMÚN.



Links recomendados

- <http://www.slideshare.net/SmartBizVN/introduction-to-agile-and-lean-software-development>
- <http://www.computer.org/csdl/mags/so/2012/05/mso2012050022.pdf>
- <https://www.youtube.com/watch?v=063muGMJ3ME>
- <https://www.youtube.com/watch?v=tEscnbjv2UM>
- <http://es.slideshare.net/davidfm/lean-software-development-5242248>