Calidad

* Superar las expectativas del cliente

¿qué espera el cliente de mi?

1. Que cumpla lo que esperaba (cero defectos)
   1. Aquí hablamos de expectativas, que quede satisfecho
2. En la fecha y costo prometidos

* En otras palabras se habla de calidad, productividad y funcionalidad

Empresas nivel 5 entregan 1.05 defectos por cada 1000 líneas de código

* Calidad se mide como defectos por cada mil líneas de codigo

Deuda técnica

* Es pedir tiempo prestado
* El cliente me exige que termine rápido el software, entonces yo le pido tiempo prestado al tiempo. Si ahora no tengo tiempo para poner comentarios o hacer documentación porque tengo que acabar el proyecto, entonces esas son cosas que me comprometo a hacer después (en el mantenimiento). En otras palabras, le pido prestado tiempo al tiempo, para pagarle el tiempo después de la entrega de software (con intereses, osea me voy a tardar más en hacer lo que no hice)

Los defectos surgen en su mayoría en la etapa de requerimientos, en promedio inyectamos 100 defectos por cada mil líneas en la etapa de codificación.

Las pruebas (el testing) es bueno, pero no es para encontrar defectos

* Es un cambio de paradigma porque ahorita yo pruebo (hago testing) para ver si hay defectos en mi programa cuando en verdad el testing debe ser solo para confirmar que no hay defectos.

¿cómo le hacemos para que a la fase de testing le llegen 0 defectos?

* Revisar lo que ya hiciste, así como cuando un examen lo terminas antes y haces el double check para asegurar que todo este bien
* En otras palabras, verificación y validación
* Verificación hace referencia a la forma (el proceso) mientras que la validación se refiere al producto en sí
  + Tenemos que asegurarnos que estamos desarrollando el producto correcto (validación) de la forma correcta (verificación)
  + Existe la VyV dinámica (pruebas) y la VyV estática (revisiones)

Un defecto es aquello que tengo que cambiar en el programa porque no quedo como el cliente lo quería

* Es igual a un retrabajo

Para medir la densidad de defectos = # de defectos / KLCD (K líneas de código)

Exam tips

* Densidad de defectos: cantidad de defectos por cada mil líneas de código

**11/01/2018**

Cuatro métricas necesarias para evaluar un software

* Tiempo
  + Administración de fecha y costo
* Calendario
  + Administración de fecha y costo
* Defectos
  + Administración de la calidad
* Tamaño
  + Para estimar y comparar

La única manera en la que los datos sean reales es si los recolectamos al mismo tiempo que trabajamos

La única manera en la que puedo comparar dos datos es si vienen de la misma fuente o son comparables (se recolectaron siguiendo un mismo estándar)

Un small Project se define como menos de 50,000 líneas de código, uno grande es mayor de 500,000 líneas de código

* En small projects no importa si usan el método ágil o el de cascada

Las tres cosas que vas a aprender en el laboratorio:

* Recolectar métricas
* Revisar antes de probar
* Diseñar antes de codificar

La calidad se mide por cuantos defectos se te escaparon hasta testing

* El chiste es encontrar los defectos antes de llegar a la etapa de testing

Todo mundo habla sobre cómo mejorar la funcionalidad y la fecha de entrega, pero lo que se les olvida es cómo entregar menos defectos (como seguir el proceso de desarrollo)

**15/01/2018**

Proceso organizacional

* Secuencia de pasos para definir como una organización va a hacer las cosas
* Son pasos muy generales, grandes
* No te dice cómo fabricar un producto

Administrar significa estimar, planear, dar seguimiento y ajustar.

Proceso operacional

* Procesos de fabricación
* Te dicen cómo hacer un requerimiento

Enfoque a procesos

* Se refiere a que es más importante la forma en la que haces un producto, que el producto mismo

Para continuar con las practicas del PSP en nuestra vida hay que recolectar métricas de tamaño, tiempo (esfuerzo), defectos y calendario.

Elementos de un proceso

* Scripts
  + Guían al programador en el uso de un proceso
* Métricas
  + Miden el proceso y el producto
* Formatos
  + Una manera conveniente y consistente de capturar datos
* Estándares
  + Proveen definiciones consistentes

Criterios de entrada

* Precondiciones

Criterios de salida

* Todo aquello que debe ser cierto para verificar que terminamos de hacer el proceso

Existen dos enemigos de la productividad

* El retrabajo (los defectos)
* Las interrupciones

Cuando corrijo un defecto, no me muevo de fase, me quedo en la fase donde estoy.

**18/01/2018**

PSP1

Los programas 1 y 2 son acerca de recolectar métricas

Estimación de tamaño y tiempo con PROBE (PROxy Based Estimation)

1. Realizar un diseño conceptual
2. Identificar cada proxy y su tamaño (cada ítem o clase y sus LDC)
3. Ajustar el tamaño basado en la historia (método PROBE A, B, C o D)
4. Calcular el tiempo basado en la historia (método PROBE A, B, C o D)

Maneras de estimar cuántas líneas de código voy a generar en mi programa

Es mejor usar siempre la correlación linear a usar el promedio, pero no siempre se puede usar la correlación lineal. Para poder usarlo, r cuadrado debe ser más de 0.5

* A, B
  + Correlación lineal
    - Es la mejor línea que minimiza la distancia entre los puntos
* C
  + Promedio
* D
  + Educated guess

Diseño conceptual

* Calcular las líneas de código de tu programa, estimando todo lo que vas a hacer de líneas agregadas, modificadas, reusadas, borradas, etc.

PSPS1

* tamaño

PSP2

* revisiones

PSP2.1

* diseño

Exam tips

* Que algo sea preciso se refiere a que algo no sea ambiguo, que dos personas lo entiendan igual
* Todos los procesos deben de tener etapas de planning y post-mortem
* Encuentra los defectos lo más cercano a donde los inyectas
* Mientras trabajas, recolecta métricas para tener datos históricas y poder hacer estimaciones y demás

**22/01/2018**

TSP

El TSP es una disciplina de procesos

* Para un seguimiento del proyecto se necesitan planes detallados y datos precisos y exactos
* Para maximizar la productividad, el equipo se enfoca primero en la calidad
* La gente que hace el trabajo debe recolectar los datos

El team software process (TSP) no funciona sin el personal software process (PSP)

* Asume que las personas ya saben el PSP

Proceso operacional

* Objetivo es fabricar algo en especifico
* Nivel micro detallado

Proceso organizacional

* Organizar muchas gentes

Métricas consolidadoras

* Métricas que se pueden comparar entre diferentes proyectos, aun y cuando se hicieron en diferentes lenguajes
* Ex: puntos función

Métricas control de producción

* Métricas que ayudan a controlar la producción

Equipos aglutinados

* Grupo de personas muy unidas

**25/01/2018**

El TSP consta de 2 procesos

* Proceso para construir un equipo (un lanzamiento)
  + Se definen metas y se asignan roles
  + Estima el tamaño del producto
  + Define la estrategia y los procesos a seguir
  + Se negocia con la alta gerencia
* Proceso para trabajar en equipo
  + El quipo se auto-administra
  + Siempre se enfoca primero a la calidad y no a la fecha de entrega

Objetivos del lanzamiento (dura entre 4-5 días)

* Tener un equipo aglutinado
* Tener metas agresivas pero realistas







