



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CURSO:
Métodos Formales de Pruebas - Testing de Software

Tema:
Introducción a Pruebas de Software

Docente: Mg. Wilder Inga

Reglas básicas

- 10 minutos de tolerancia
- Toma de asistencia al inicio de la clase
- Participación durante la clase
- Participación proactiva en los ejercicios grupales, se usaran herramientas colaborativas de G Suite.
- Las ausencias justificadas no dan derecho a nota (actividades en clase)
- Break 15 minutos

Video de Calidad de Software

Introducción

- ⑨ "All in all, coders introduce bugs at the rate of 4.2 defects per hour of programming. If you crack the whip and force people to move more quickly, things get even worse"
 - ③ Watts Humphrey (note),
<http://www.cs.usask.ca/grads/jpp960/490/BombSquad.html>
- ⑨ 20 por Día / 100 por Semana / 400 por Mes / 5000 por Año
- ⑨ "5000 Defect Project (not atypical for IBM)"
 - ③ Paul Gibson, Testing Challenges for IBM, UK Test 2005
Keynote, <http://www.uktest.org.uk>

Introducción.

Algunas cifras

- ⑨ Total de recursos empleados en pruebas:
 - ③ 30% a 90% [Beizer 1990]
 - ③ 50% a 75% [Hailpern & Santhanam, 2002]
 - ③ 30% a 50% [Hartman, 2002]
- ⑨ Mercado de herramientas: \$2,6 billion
 - ③ Coste por infraestructura inadecuada:
 - Transporte y manufactura: \$1,840 billion
 - Servicios financiero: \$3,342 billion
 - ③ Costes de reparación en función del instante en el ciclo de vida [Baziuk 1995]
 - Requisitos: x 1
 - Pruebas de sistema: x 90
 - Pruebas de instalación: x 90-440
 - Pruebas de aceptación: x 440
 - Operación y mantenimiento: x 880
 - ③ Fuente general: The economic impact of inadequate infrastructure for software testing. NIST Report – May 2002



El control de calidad se ocupa de las actividades que garantizan que el producto se desarrolle según los requisitos definidos. Se ocupa de todas las acciones que son importantes para controlar y verificar determinadas características del producto, incluidas las pruebas. La inspección y las pruebas de los productos es el aspecto más importante del control de calidad.

Software Testing VS Quality Assurance

- En la industria de TI, a menudo se observa que las personas generalmente no distinguen entre QA del software y las pruebas de software. Los tester a menudo se consideran profesionales de QA del software porque los objetivos de las pruebas de software y el aseguramiento de la calidad son los mismos, ejm. Garantizar que el software sea de la mejor calidad.
- Como su nombre indica, los procesos de QA se llevan a cabo para asegurar que la calidad del producto esté en línea con los requisitos del cliente. Los profesionales de QA trabajan en el desarrollo e implementación de todos los procesos necesarios para asegurar que todos los procedimientos necesarios del ciclo de vida del desarrollo de software se sigan correctamente.

Software Testing VS Quality Assurance

Atención proactiva de QA:

1. Prevención de defectos
2. Procesos
3. Mejora continua de procesos

Atención de testing:

Identificar o descubrir defectos y errores en el software. Implica una prueba rigurosa real del software para ver si hay algún defecto o variación del requisito del cliente que deba solucionarse.

Son parte de QC y se enfocan solo en actividades orientadas al producto. La prueba del software se lleva a cabo durante la fase de testing y solo se identifican los defectos y no se corrigen en este proceso. La reparación de defectos no forma parte de las pruebas de software.

Cost of poor quality

“The Cost of Poor Software Quality in the US: a 2018 report” **How much was spent on IT products, services, and labor?:**

- \$2 trillion+ in the United States (estimate by CISQ) – 6.2% vs 2017
- \$3.7 trillion globally (estimate by Gartner) – 11.1% vs 2017
- \$3.74 trillion in 2019 (0.6% vs 2018)
- \$6.3 trillion globally (estimate by Apptio)
- Aprox. 45% labor cost

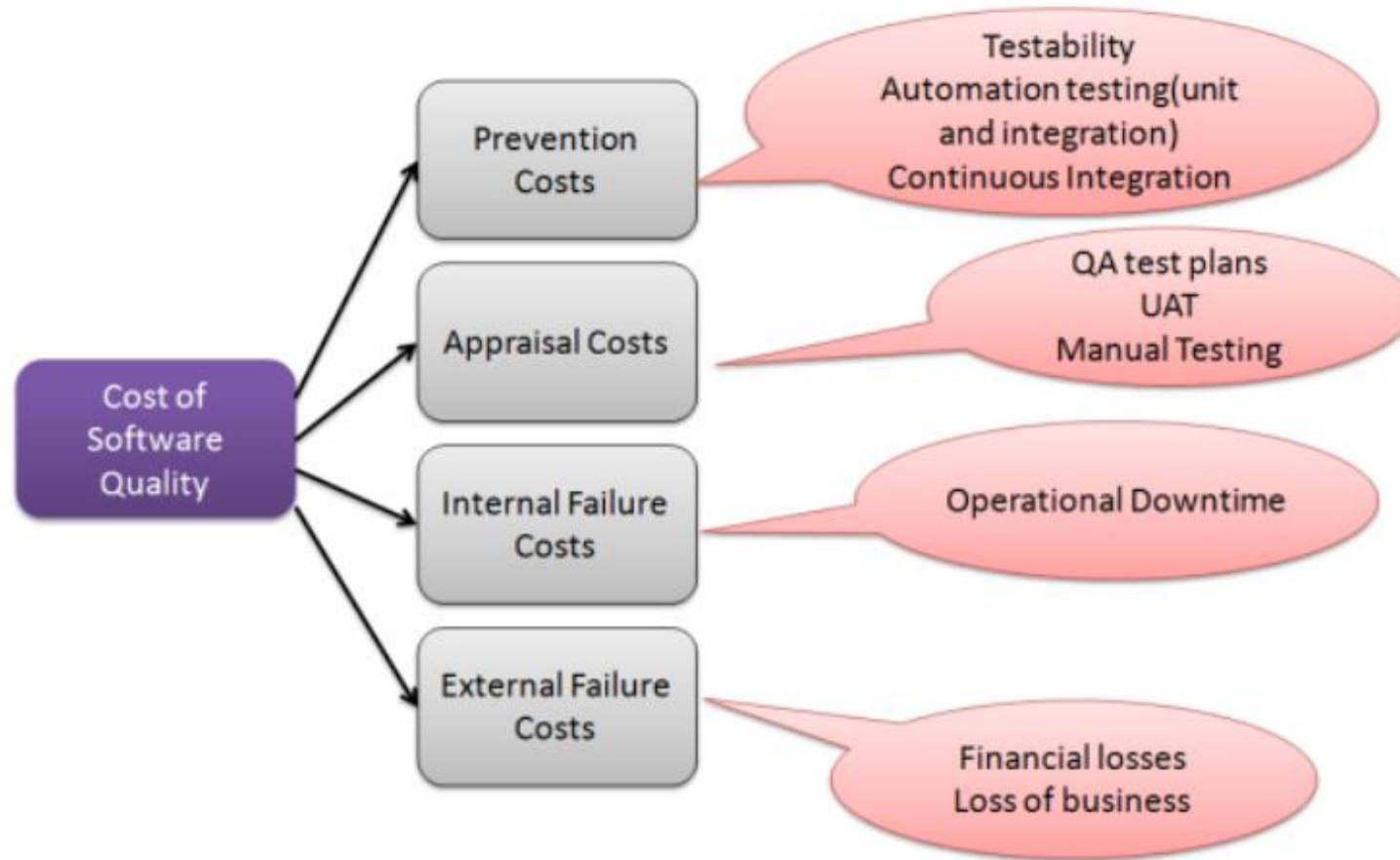
CPSQ (Cost of Poor Software Quality) for the United States in 2018:

- External failures and deficiencies – \$1.43 trillion
- Internal failures and deficiencies – \$.8 trillion
- Technical debt – \$.54 trillion
- Management failures – unknown contribution at this time

Cost of Software Quality

El costo de la calidad es importante porque cuando decide realizar pruebas de software para su producto, realmente va a invertir su tiempo, dinero y esfuerzo en realizar controles de calidad. Al realizar un análisis del costo de la calidad del software, sabría cuál es el retorno de esa inversión (ROI).

Cost of Software Quality



Cost of Software Quality

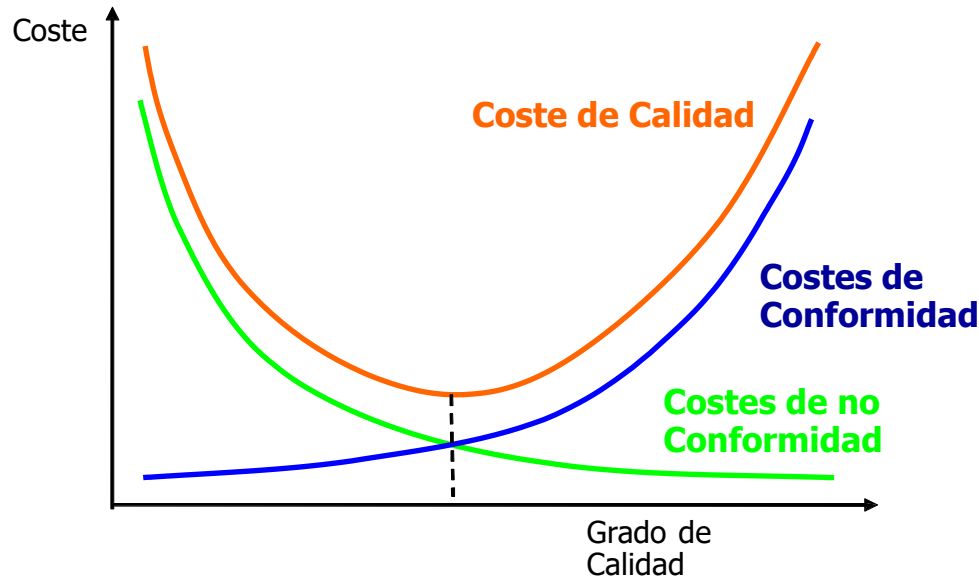
Costos de conformidad

- Costos de prevención: monto gastado para garantizar que todas las prácticas de garantía de calidad se sigan correctamente. Esto incluye tareas como capacitar al equipo, revisiones de código y cualquier otra actividad relacionada con el control de calidad, etc.
- Costos de evaluación: es la cantidad de dinero gastada en planificar todas las actividades de prueba y ejecutarlas, como la elaboración de casos de prueba y luego ejecutarlos.

Costos de no conformidad

- A nivel internos: es el gasto que ocasiona la subsanación de los errores. Costo de las pruebas unitarias por subsanación.
- A nivel externos: es el gasto que se produce cuando el defecto es encontrado por el cliente. Estos gastos son mucho más de los que surgen a nivel interno, especialmente si el cliente no está satisfecho o el error se disemina afectando colateralmente a otros componentes de un sistema.

Costes



⑨ Invertir en pruebas (y en general en calidad) es rentable

③ (Krasner)

- En manufactura: 5-25%
- En software: 10-70%

③ En Empresas de Automoción:

- 4% Excelencia
- 4-8% Buenos
- >10% No deseables

⑨ $C = C_{\text{conformidad}} + C_{\text{noconformidad}}$

③ $C_{\text{conformidad}} = C_{\text{prevención}} + C_{\text{evaluación}}$

③ $C_{\text{noconformidad}} = C_{\text{internos}} + C_{\text{externos}}$

⑨ Costes de inactividad en cliente (N. Donfrio, 2002):

- ③ Cadena Suministro: 300K\$/hora
- ③ ERP/e-comm: 480K\$/hora

Introducción.

Definiciones

- ⑨ Definición 1: La prueba (testing) es el proceso de ejecutar un programa con la intención de encontrar fallos [Glenford J. Myers]
 - ③ Un buen caso de prueba es el que tiene alta probabilidad de detectar un nuevo error
 - ③ Un caso de prueba con éxito es el que detecta un error nuevo
- ⑨ Definición 2 [Cem Kaner]:
 - ③ Una investigación técnica del producto bajo prueba
 - ③ ...para proporcionar a los interesados (stakeholders)
 - ③ ...información relacionada con la calidad

Deuda técnica

- La Deuda Técnica es una excelente metáfora (creada por Ward Cunningham) que nos ayuda a pensar sobre algunos problemas del desarrollo de software. Según la metáfora, hacer las cosas rápido y mal nos incrementa la deuda técnica, la cual es similar a la deuda financiera. Al igual que la deuda financiera, la deuda técnica tiene pago de intereses, que vienen en la forma del esfuerzo extra que será necesario hacer en el futuro por una elección rápida y mala de diseño. Podemos decidir seguir pagando el interés, o podemos pagar el capital al hacer un refactor del diseño hacia un diseño mejor. Aunque hay un costo de pagar este capital, nos ahorramos el pago de intereses en el futuro.

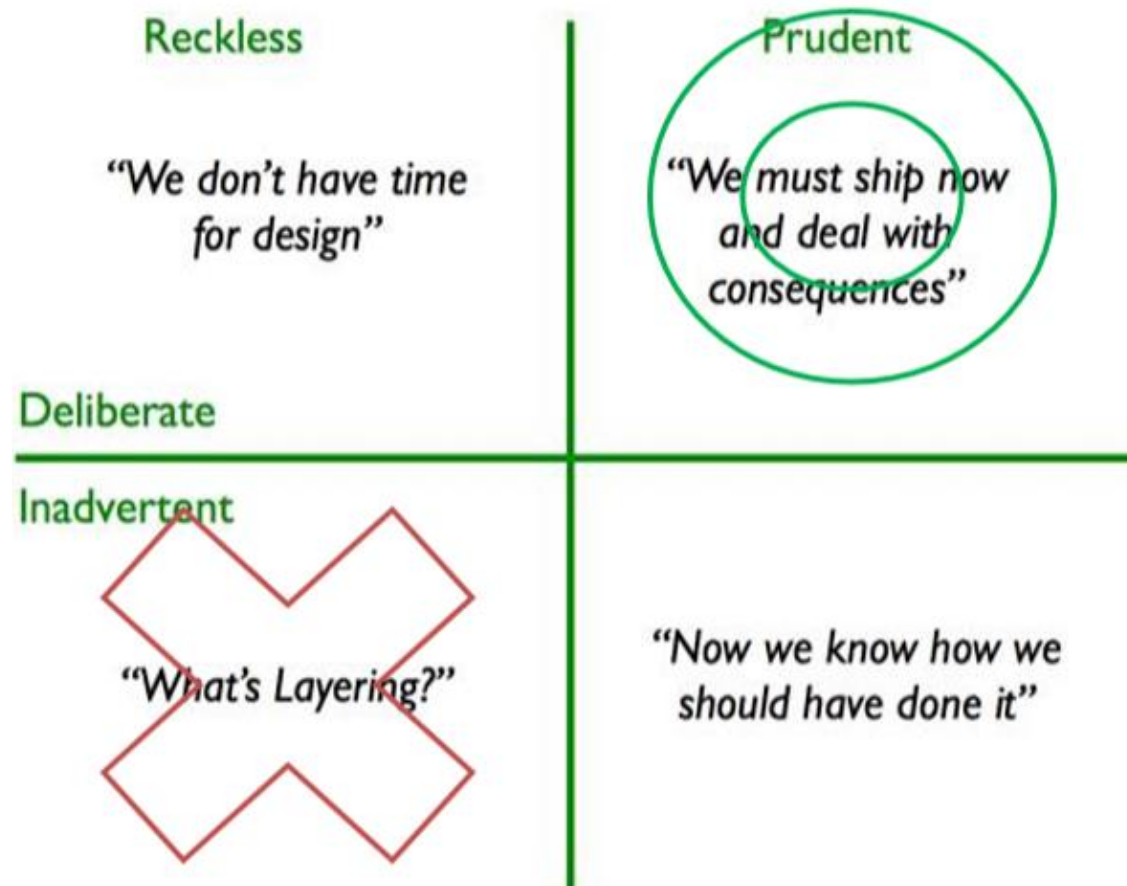
Deuda técnica

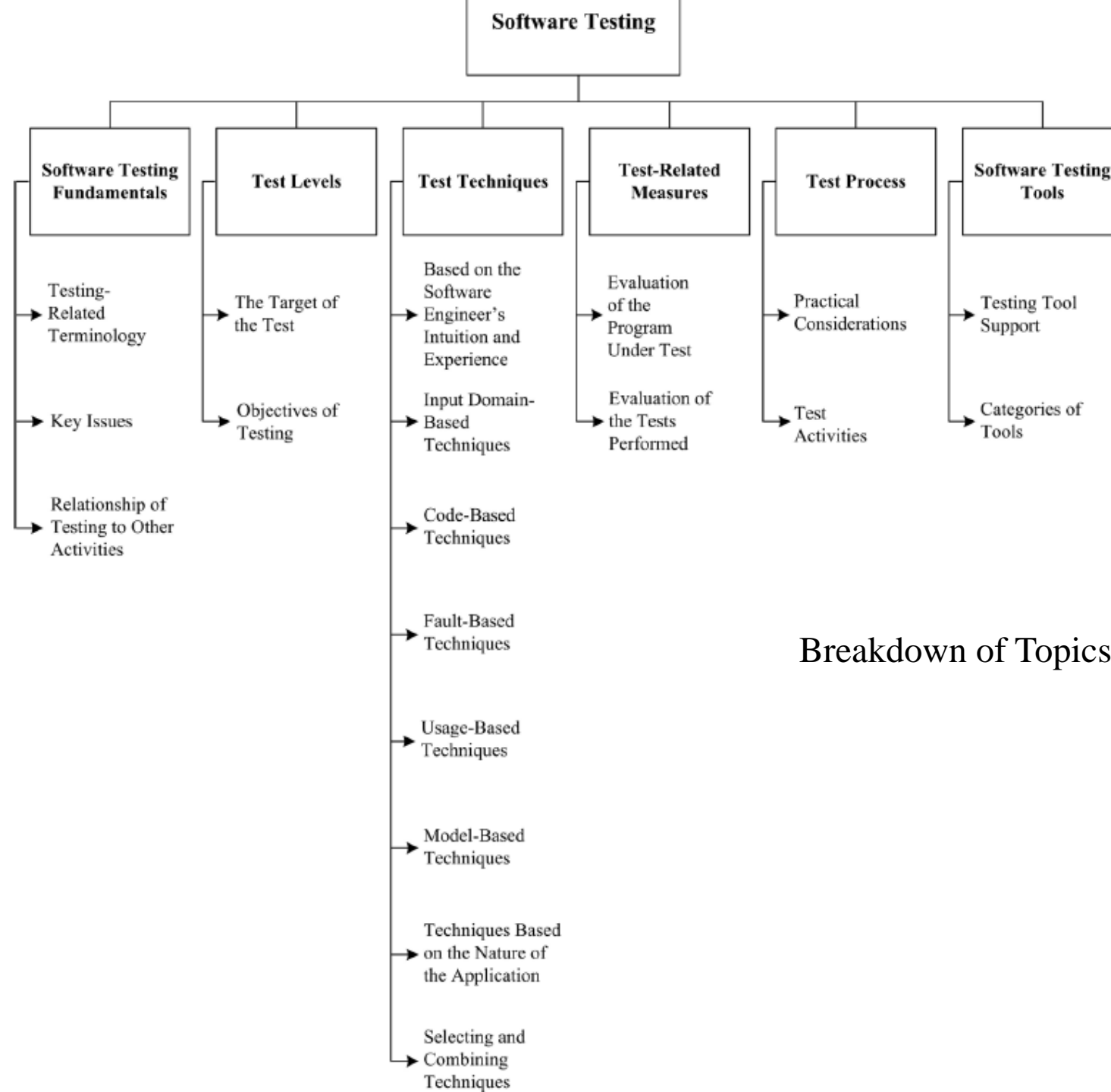
La deuda técnica es el coste y los intereses a pagar por hacer mal las cosas.

La deuda técnica al final siempre alguien la paga. O la paga el proveedor que desarrolla el software o la paga el cliente que lo usa o compra.

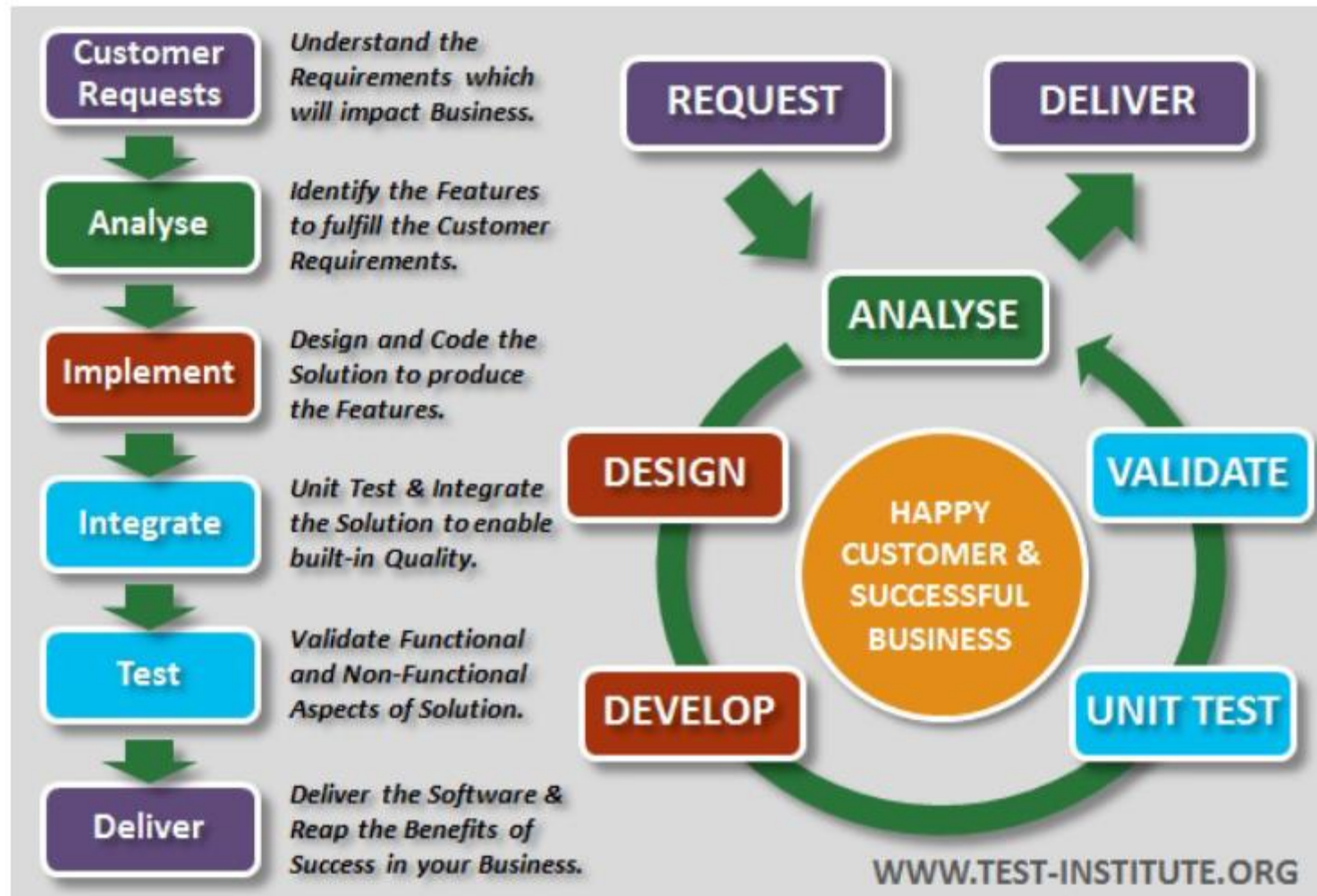
La deuda puede ser prudente o imprudente, y también hay una diferencia entre una deuda deliberada y otra inadvertida.

Cuadrante de Fowler:





Breakdown of Topics for the Software Testing KA



Software Testing Methodology in Software Engineering

Actividad

Tarea para la semana

- Definir el grupo de 5 miembros
- Describa el proyecto de software que usará para el curso. Debe estar desarrollado.
- Valide que el software que va a usar para el curso es de una calidad aceptable