



Pruebas de integración

Universidad Autónoma de Coahuila

Facultad de Sistemas

Carlos Nassif Trejo García




Sistema

- Colección de modulo interconectados con un único objetivo
- Subsistema: Sistema provisional que no esta integrado completamente

Importancia

- Errores de interfaz
- Limitaciones de las pruebas unitarias
- Identificar módulos con errores

Unit test vs. Integration test



Crear una
versión
funcionable
del sistema

Juntar los módulos de forma
incremental

Asegurarse que cualquier
modulo adicional funcione sin
afectar las funcionalidades ya
implementadas

Interfaces

- Procedure call
- Shared Memory
- Message Passing



Put them
together
errors

Construcción

Funcionalidad
inadecuada

Ubicación de
la
funcionalidad

Cambios en la
funcionalidad

Funcionalidad
agregada

Uso incorrecto
de la interfaz

Mal entendido
de la interfaz

Alteración en
estructura de
datos

Manejo
incorrecto de
errores

Etc...

Ventajas



Defectos son detectados en una etapa temprana



Mas fácil de corregir



Feedback temprano sobre las métricas de los módulos individuales y el sistema en general



Agendar arreglos de defectos es flexible y puede llevarse junto con la etapa de desarrollo

Tipos de Integration Testing



Intrasistema

Intersistema

Por pares

Intrasystem Testing

Integración de bajo nivel

Combinar módulos para construir un sistema

Ejemplo: Cliente servidor

Intersystem Testing

Integración de alto nivel

Sistemas no dependiente de interfaces

Todos los sistemas se conectan y se hacen pruebas end-to-end

Asegurar interconexión del sistema

Solo se prueba una funcionalidad a la vez

Pairwise Testing

Integración de nivel medio

Solo un par de sistemas interconectados son probados

Probar que 2 sistemas en construcción funcionan como lo esperado

Técnicas de integración

No es necesario esperar a que todos los módulos estén terminados

Incremental

Top Down

Bottom Up

Sandwich

Big Bang

Incremental

Ciclos incrementales creando builds

En cada ciclo se prueba, arreglan errores y se pasa al siguiente ciclo

Por capas

- Self-Contained: Contiene el código necesario
- Stable: El subsistema puede estar en servicio por 24hrs sin anomalías

Check-in request

Release note

Creando build

Tomar la ultima versión aprobada de los módulos

Compilar

Push al repo

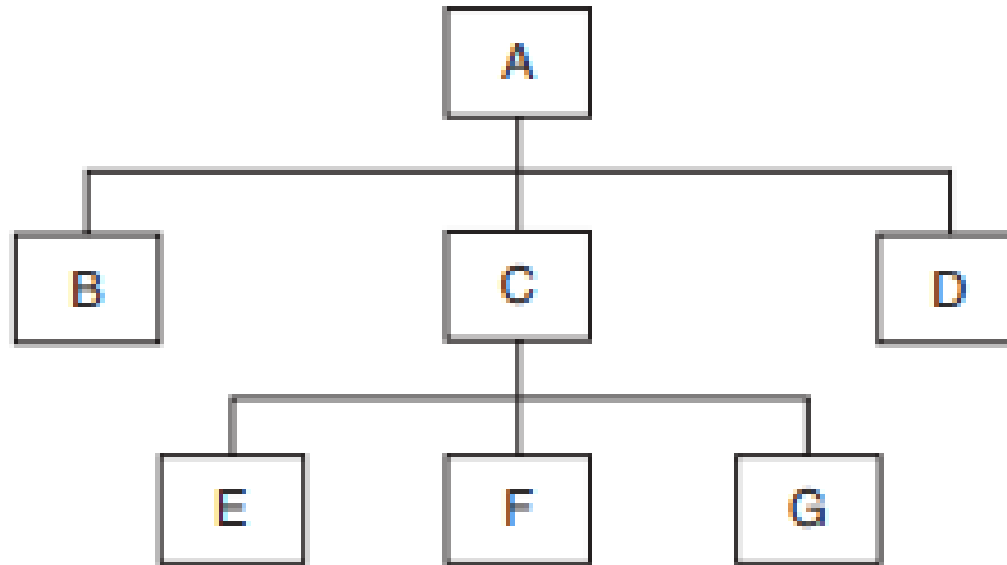
Enlazar los módulos a subbassemblies

Verificar que sean correctos

Control de versiones

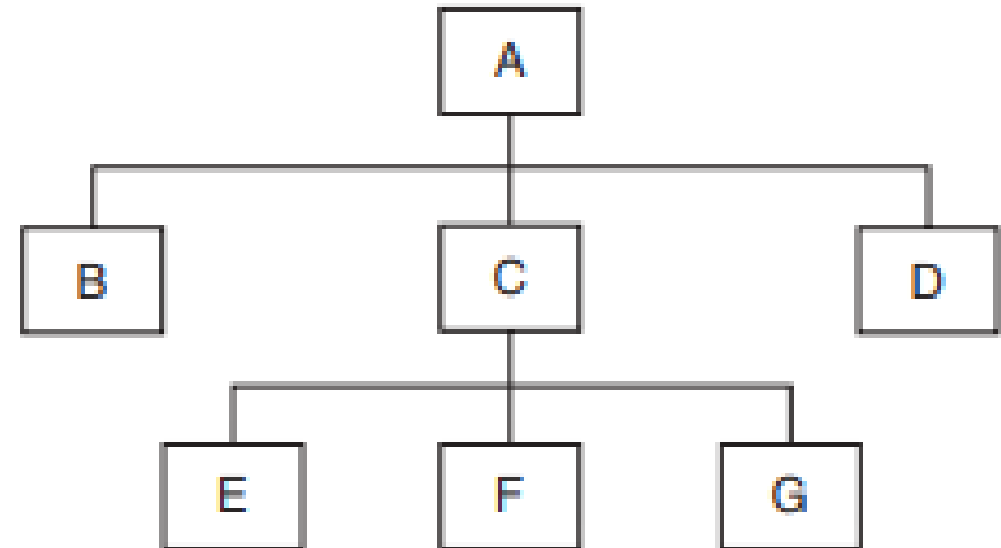
Top Down

- Sistema jerárquico: Módulos principales se descomponen en módulos secundarios



Pasos

- Tomar como base el modulo raíz con sus stubs
- Escoger un stub para remplazarlo con el modulo
- Hacer pruebas
- Repetir



Ventajas

Se puede observar la funciones del nivel de sistema en lo que la integración se va desarrollando

Encontrar problemas de interfaz se vuelve sencillo por la naturaleza de integración

Las pruebas desarrolladas en integración, se pueden reutilizar en pruebas de regresión

Es natural que las pruebas del nodo principal correspondan a funciones de sistema

Limitaciones

Normalmente no se puede apreciar funciones del sistema en una etapa temprana

Selección entre casos de prueba y diseño de stubs se vuelve compleja por la distancia entre el modulo principal y el stub

Bottom Up

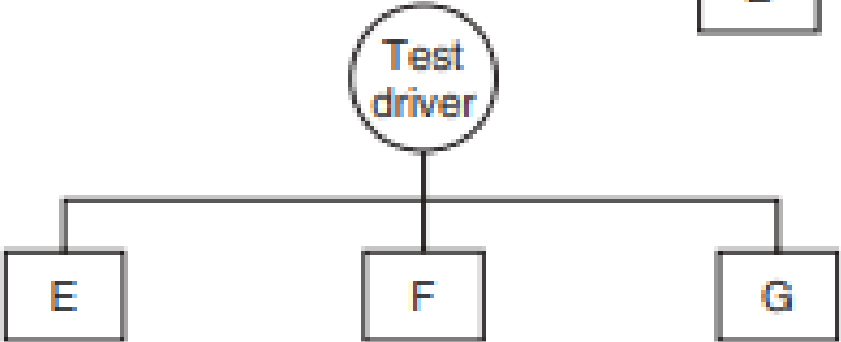
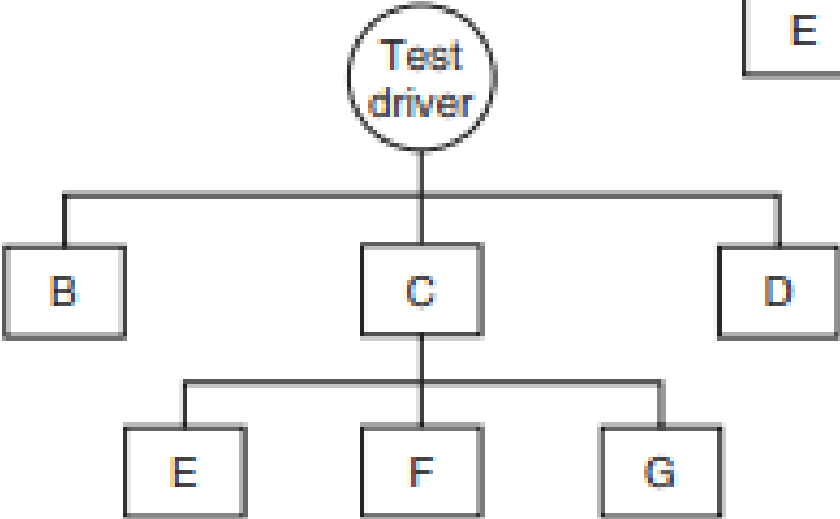
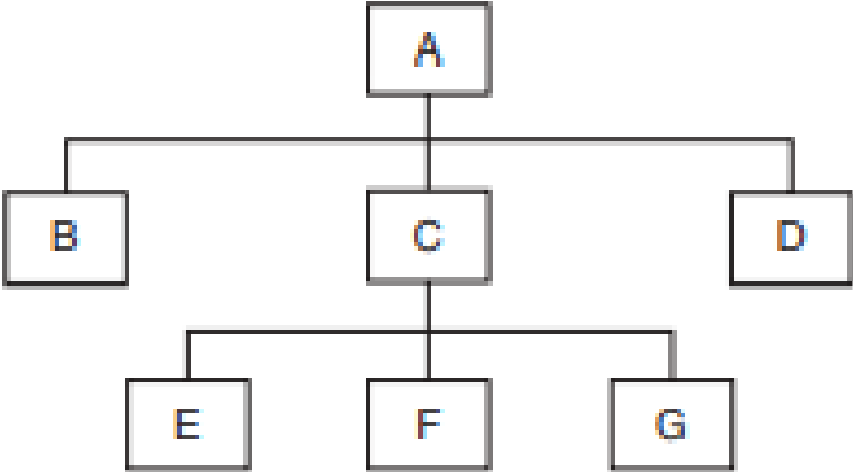
Empieza con módulos de bajo nivel

- No llaman a otro modulo

Construir un test driver que llame a esos módulos

Probar con el test driver

Si las pruebas son satisfactorias, remplazarlo con el modulo real





Ventajas

- Si los módulos de bajo nivel normalmente son llamadas por otros módulos, tiene mas sentido integrar de esta manera

Desventajas

No se pueden observar funcionalidades del sistemas hasta que se integre módulos de alto nivel

Los errores suelen ocurren en módulos de alto nivel, por lo que su detección es mas lenta

Comparación

Validación en decisiones de diseño:
Top-Down

Observar funciones de sistema: Top-Down

Dificultad en diseñar casos de prueba: Bottom-Up

Reusar casos de prueba: Top-Down

Sandwich

- Capas:
 - Top Layer: Top-Down
 - Middle Layer: Big Bang
 - Bottom Layer: Bottom Up



Big Bang

Pruebas los módulos de forma individual

Juntas todos los módulos para construir el sistema

Pruebas!