



Aproximación cuantitativa a través de pruebas unitarias de caja negra a partir de un desarrollo iterativo utilizando Rational Rose para la generación de test

Alberto Díaz-Maroto Martínez
Consultor de Ingeniería del Software
Alberto.diaz@dvc.es



daVinci Consulting

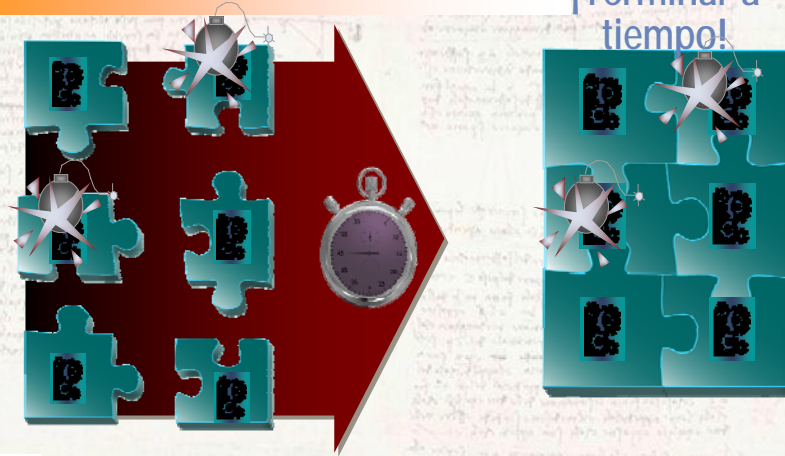
ÍNDICE

- ***El problema***
- ***La solución***
- ***Modelado UML***
- ***Los pasos a seguir***



El problema

¡Terminar a tiempo!



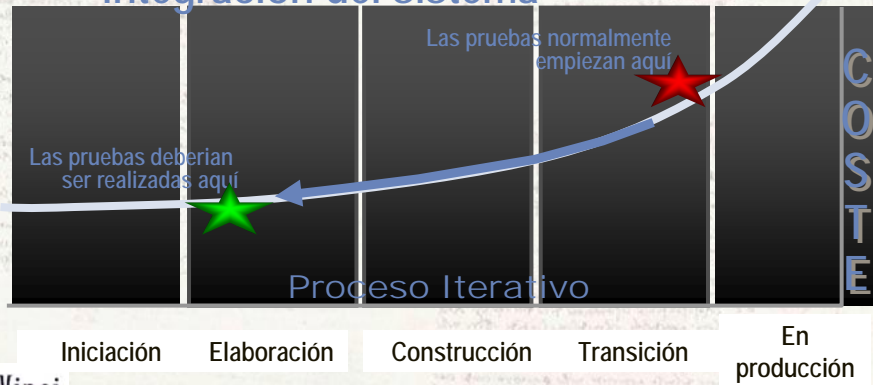
ÍNDICE

El problema

- La solución
- Modelado UML
- Los pasos a seguir

La solución

Empezar a probar antes de la integración del sistema

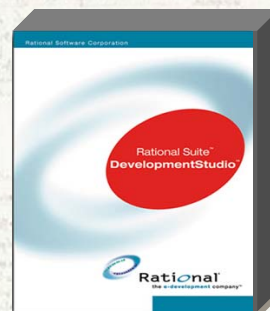


La solución: Rational QualityArchitect (RQA)

Rational Rose Enterprise

Rational Suite DevelopmentStudio

Rational Suite Enterprise

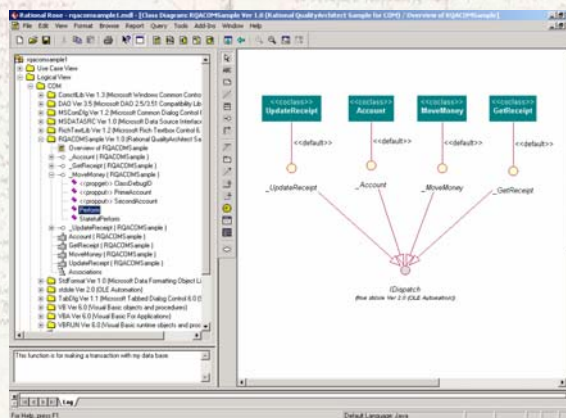


Utilizar el modelo visual para generar pruebas de componentes

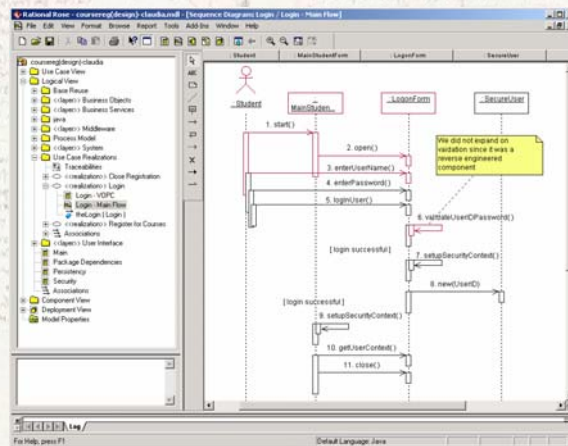
ÍNDICE

- El problema
- La solución
- Modelado UML
- Los pasos a seguir

La arquitectura



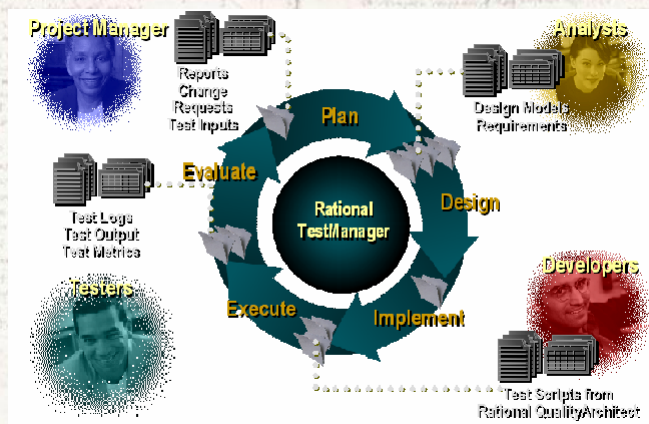
Los modelos



Pruebas dirigidas por el diseño

1. **Rapidez en la generación de pruebas, eliminando el trabajo extra de los desarrolladores.**
2. **Pruebas para la lógica de servicios de negocio: EJB y COM.**
3. **Si no se dispone del modelo lógico podemos utilizar la ingeniería inversa.**
4. **Metodología de pruebas integrada con el ciclo de desarrollo.**

Metodología de pruebas



ÍNDICE

- El problema
- La solución
- Modelado UML
- Los pasos a seguir

Los pasos a seguir

1. Generar el componente
2. Dos tipos de pruebas:
 - 2.1 Pruebas básicas
 - 2.2 Pruebas de escenario
3. Evaluación de los resultados

1. Generar el componente

Ingeniería directa

Modelo

Los desarrolladores
completan el código

Generar el
"esqueleto"
del código

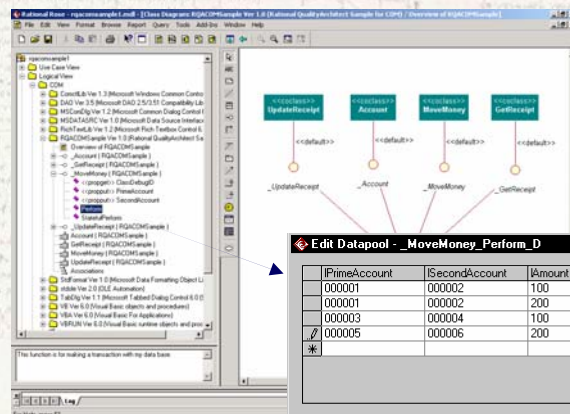
Crear
componentes

2. Dos tipos de pruebas

En ambos tipos de pruebas el script que se genera es en Java o en Visual Basic dependiendo del tipo de componente que estemos probando:

1. Si es una **EJB** entonces el script es en **Java**.
2. Si es un **COM** el script es en **Visual Basic**, independientemente del lenguaje en el que esté compilado el COM.

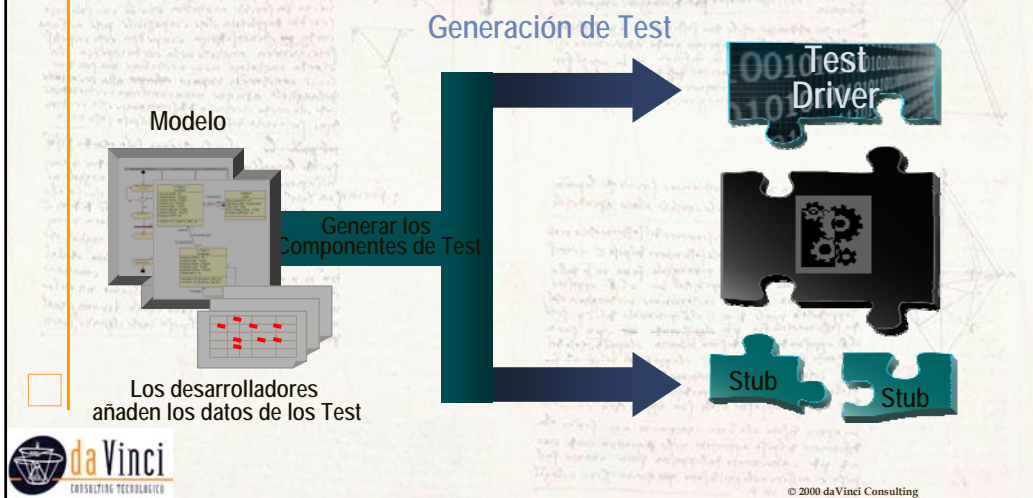
2.1 Pruebas básicas



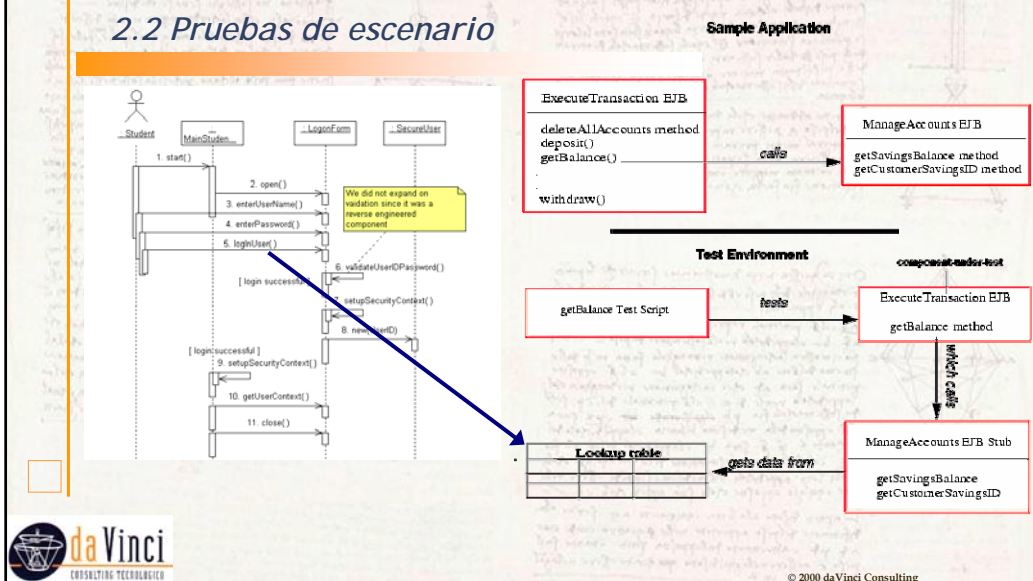
Generación de un Datapool:
tabla donde cada registro es un conjunto de valores de entrada y valores esperados de salida para la función para la que deseamos crear el test.

PrimeAccount	SecondAccount	Amount	ITranType	expectedReturn	expectedError
000001	000002	100	2	Bob	
000001	000002	200	2	Tom	
000003	000004	100	2	Henry	
000005	000006	200	1		
*					

2.2 Pruebas de escenario



2.2 Pruebas de escenario



3. Evaluación de las pruebas: Test Manager

Event Type	Result	Date & Time	Failure Reason	Computer Name
Computer Start	Fail	28/06/2001 17:13:37		
Script Start (MoveMoneyPerform)	Fail	28/06/2001 17:13:37		
Message	Pass	28/06/2001 17:13:37		
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Message	Fail	28/06/2001 17:13:56	See Description	
Script End (MoveMoneyPerform)	Fail	28/06/2001 17:13:56	See Description	
Computer End	Fail	28/06/2001 17:13:56		



GRACIAS

Alberto Díaz-Maroto Martínez
Consultor de Ingeniería del Software
Alberto.diaz@dvc.es