

MoFQA: Una propuesta para la generación automática de *tests* a partir de modelos siguiendo el proceso TDD

Linda Riquelme

Ing. Magalí González
Tutora

Dr. Luca Cernuzzi
Co-Tutor

MSc. Nathalie Aquino
Colaboradora

Ingeniería Informática

Proyecto Final de Carrera

Mayo 2017

Crisis de *software*

- Diversidad de módulos y sistemas
- Tiempos cortos
- Comunicación cliente-desarrollador
- Cambios en requerimientos

Errores en el *software*

Operación no permitida.

Has intentado ejecutar una acción que nuestro servidor ha señalado como malintencionada. En este momento no puede completarse tu petición.

Puedes ponerte en contacto con nuestro personal técnico para notificarle el problema enviando un correo electrónico a webmaster@zara.com.

IR A LA PÁGINA PRINCIPAL.

Fuente: <https://iuveaene.wordpress.com/2010/09/03/problemas-con-la-web-de-zara/>

Errores en el *software*

Delivery to the following recipient failed permanently:

webmaster@zara.com

Technical details of permanent failure:

Google tried to deliver your message, but it was rejected by the recipient domain. We recommend contacting the other email provider for further information about the cause of this error. The error that the other server returned was: 550 550 Invalid recipient <webmaster@zara.com> (#5.1.1) (state 14).

Fuente: <https://iuveaene.wordpress.com/2010/09/03/problemas-con-la-web-de-zara/>

Errores en el *software*

TRF / COLECCIÓN / FALDAS

14 artículos

CARACTERÍSTICAS ▼ COLOR ▼ TALLA ▼ PRECIO ▼

PÁGINA 1 DE 2 | VERTOOOS | ◀ ▶

FALDA MINI RETRO
1,00 EUR

FALDA LENTEJUELAS
25,95 EUR

FALDA LENTEJUELAS
25,95 EUR

FALDA FELPA
4€ 04€ 01€ 02

FALDA TULIPÁN
4 JVA 01 02

FALDA VUELO
4 JVA 01 02

Fuente: <http://www.ticbeat.com/tecnologias/error-permite-comprar-en-zara-online-por-euro/>

Errores en el *software*



Lo que pidió el cliente



Lo que entendió el jefe de proyecto



Lo que diseñó el analista



Lo que escribió el programador



Lo que el consultor describió



Lo que se documentó



Lo que instalaron los de operaciones



Lo que se llegó a pagar

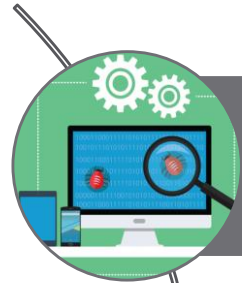


Lo que soporte llegó a hacer



Lo que el cliente necesitaba

Testing de Software



Identificar errores en el desarrollo

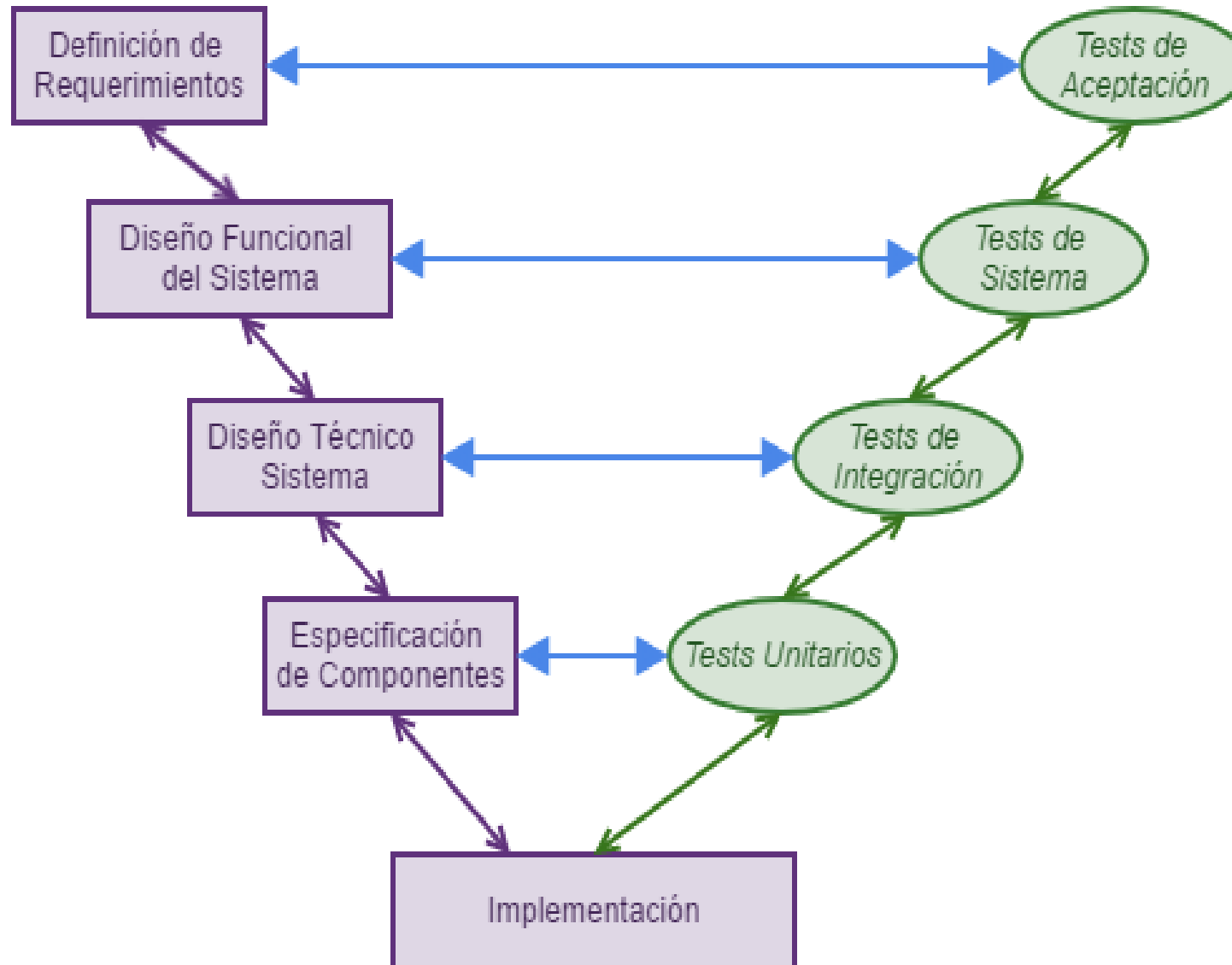


Factor de calidad del producto



Reducción en el costo de mantenimiento

Testing en el desarrollo de *software*



Modelo V para el desarrollo de *software* (Baker et al., 2007)

Costo del *Testing* de *Software*

Cantidad de combinaciones de datos de prueba

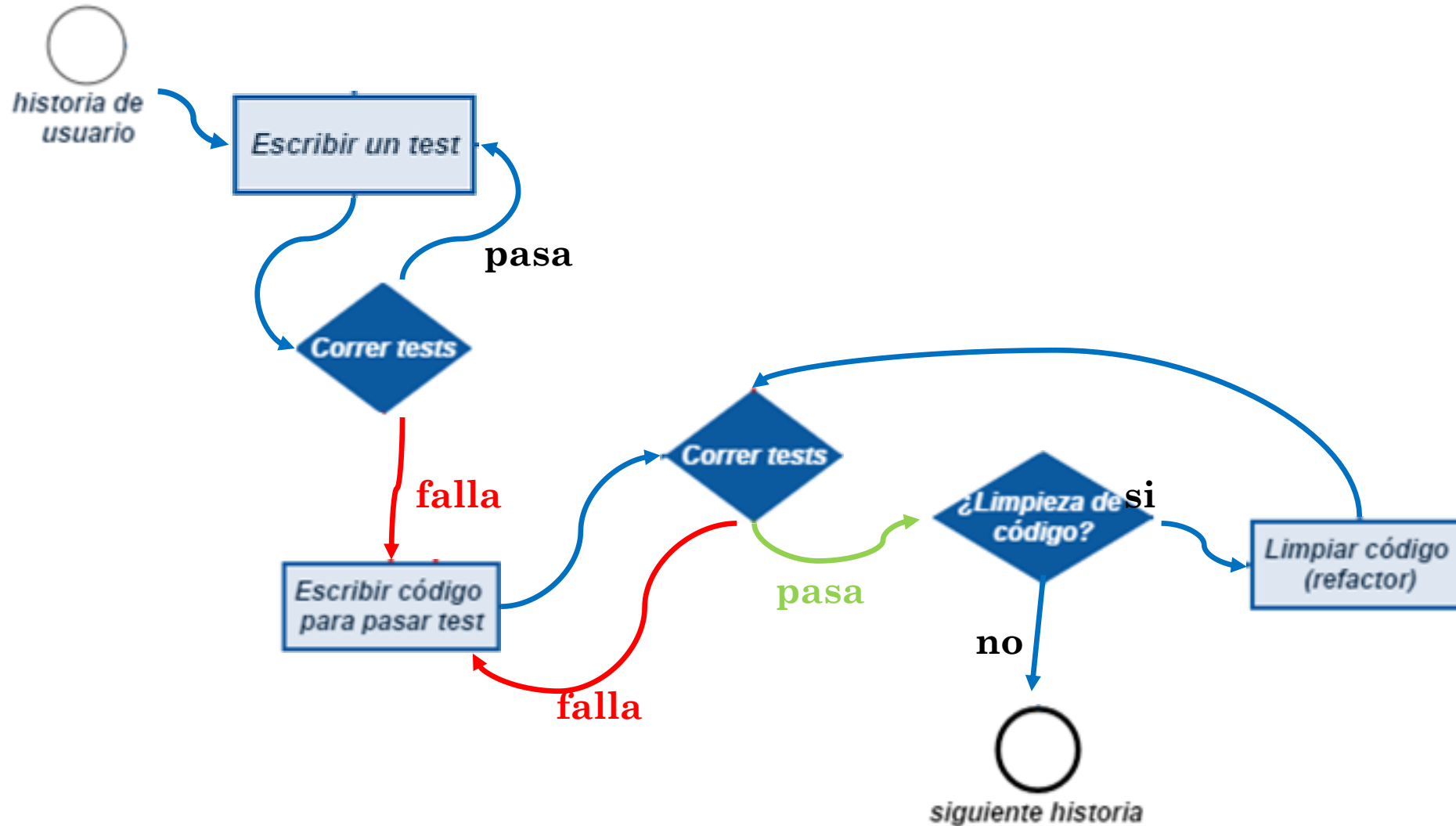
Orden de la ejecución afecta los resultados

Diversidad de entornos de prueba

Necesidad de *tests* de regresión

Tiempo y conocimientos técnicos

Test-Driven Development (TDD)



Test-Driven Development (TDD)

Ventajas

Confianza y
reducción de
errores.

Satisfacción de
requerimientos e
interfaz de usuario.

Al menos un test
por cada
funcionalidad.

Códigos más
simples y limpios.

Limitaciones

Tiempo para definir
y mantener tests.

Elaborados por el
desarrollador.

Testing basado en
tests unitarios.

Test-Driven Development (TDD)

Ventajas

Confianza y
reducción de
errores.

Satisfacción de
requerimientos e
interfaz de usuario.

Al menos un test
por cada
funcionalidad.

Códigos más
simples y limpios.

Limitaciones

Tiempo para definir
y mantener tests.

Elaborados por el
desarrollador.

Testing basado en
tests unitarios.

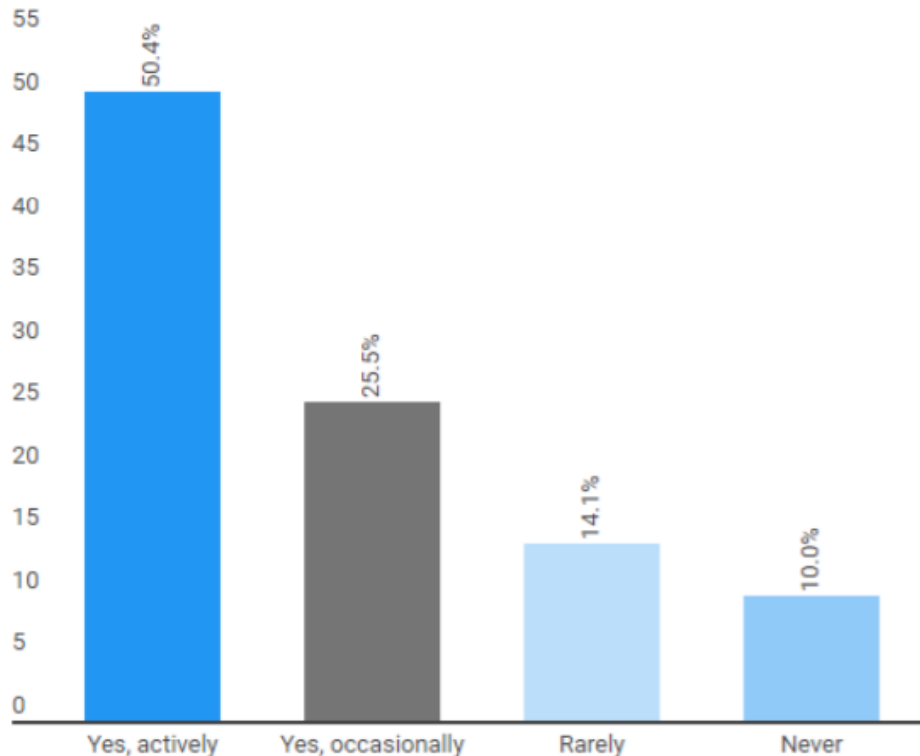
Automatización de *Testing*



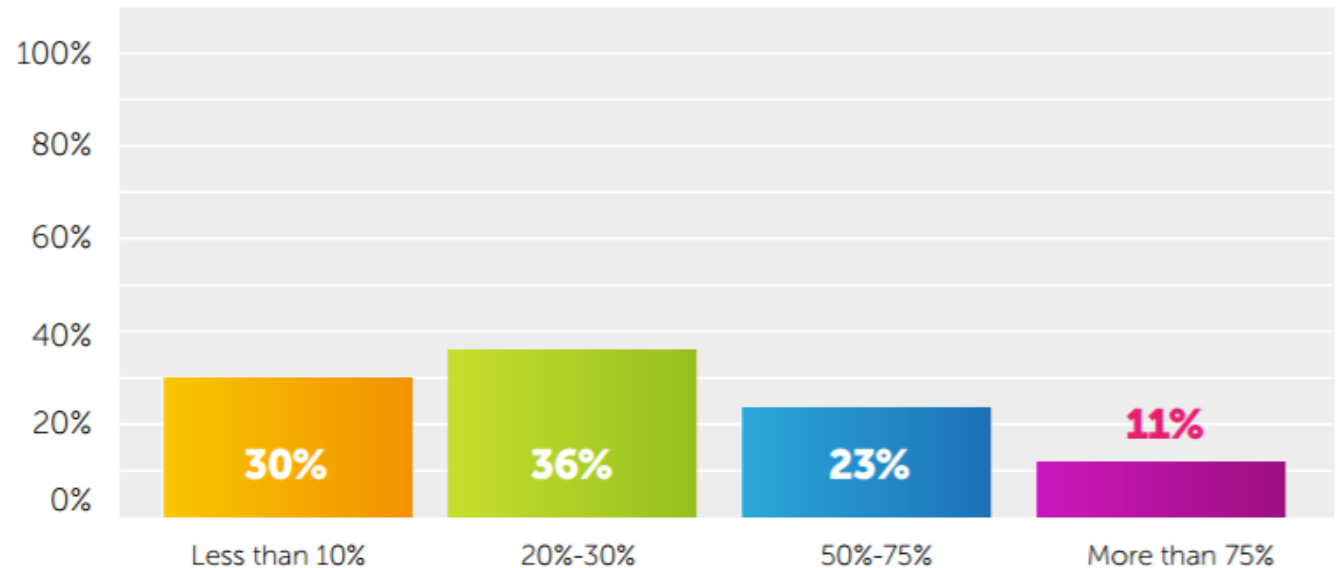
Script-based Testing (Utting, 2010)

Automatización de *Testing*

Do you use automated tests in the process of software testing?



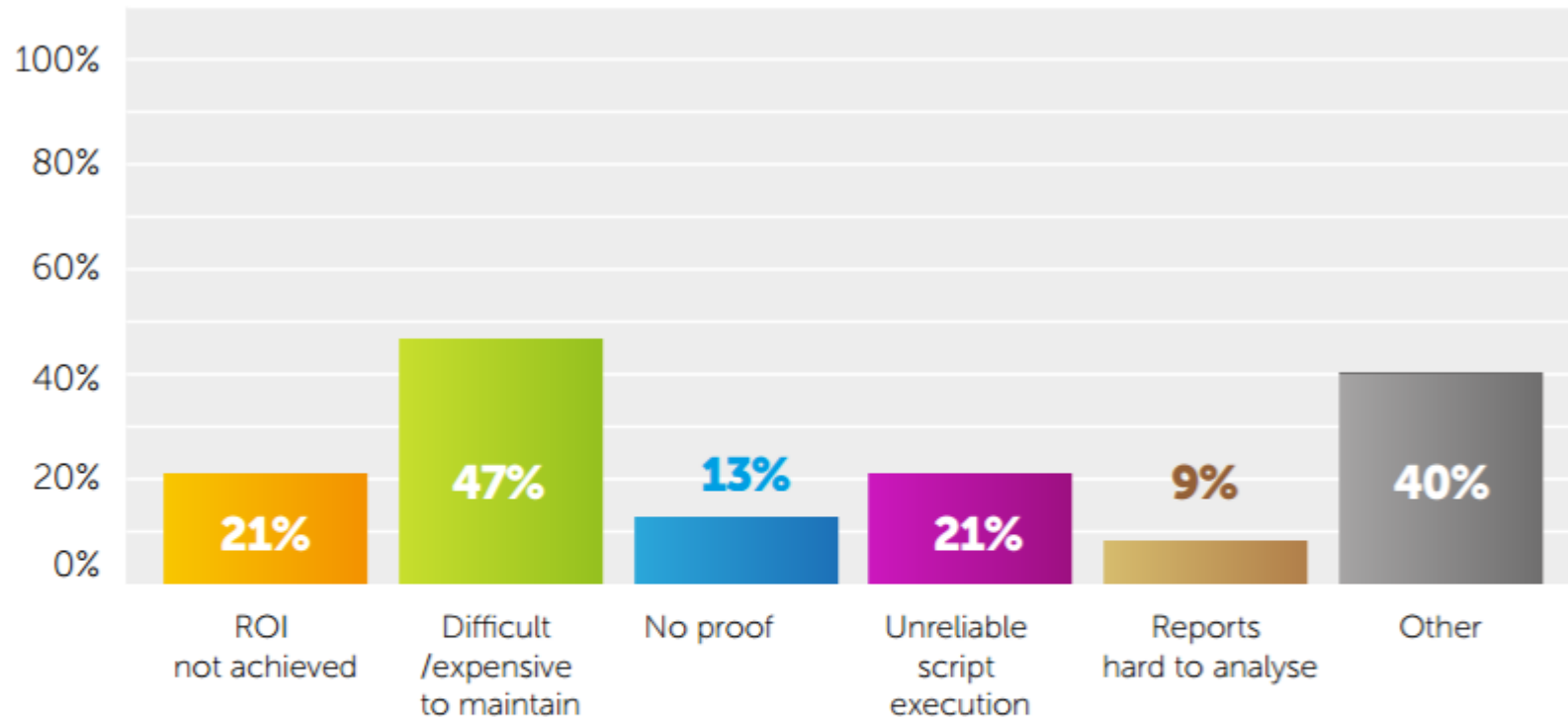
Modern State of Test Automation (*Applied Systems, 2016*)



WHAT % OF YOUR TEST CASES ARE AUTOMATED?

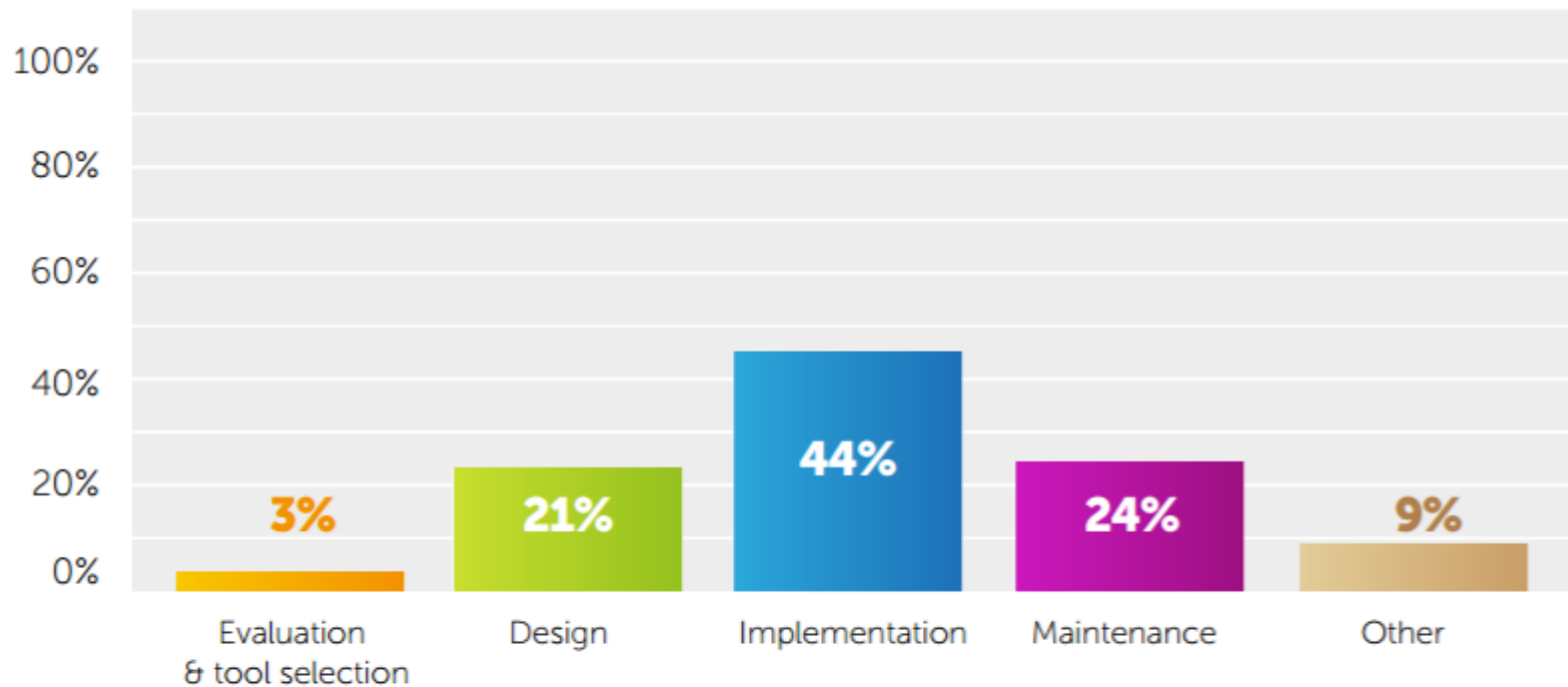
European Software Testing Benchmark Report (*2016*)

Automatización de *Testing*



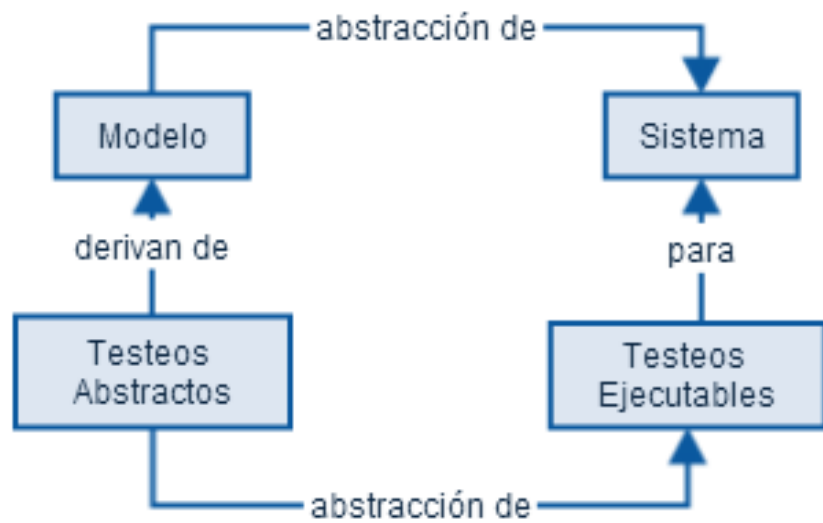
WHAT ARE THE MAIN ISSUES WITH THE QUALITY OF YOUR AUTOMATED TESTING?

Automatización de *Testing*

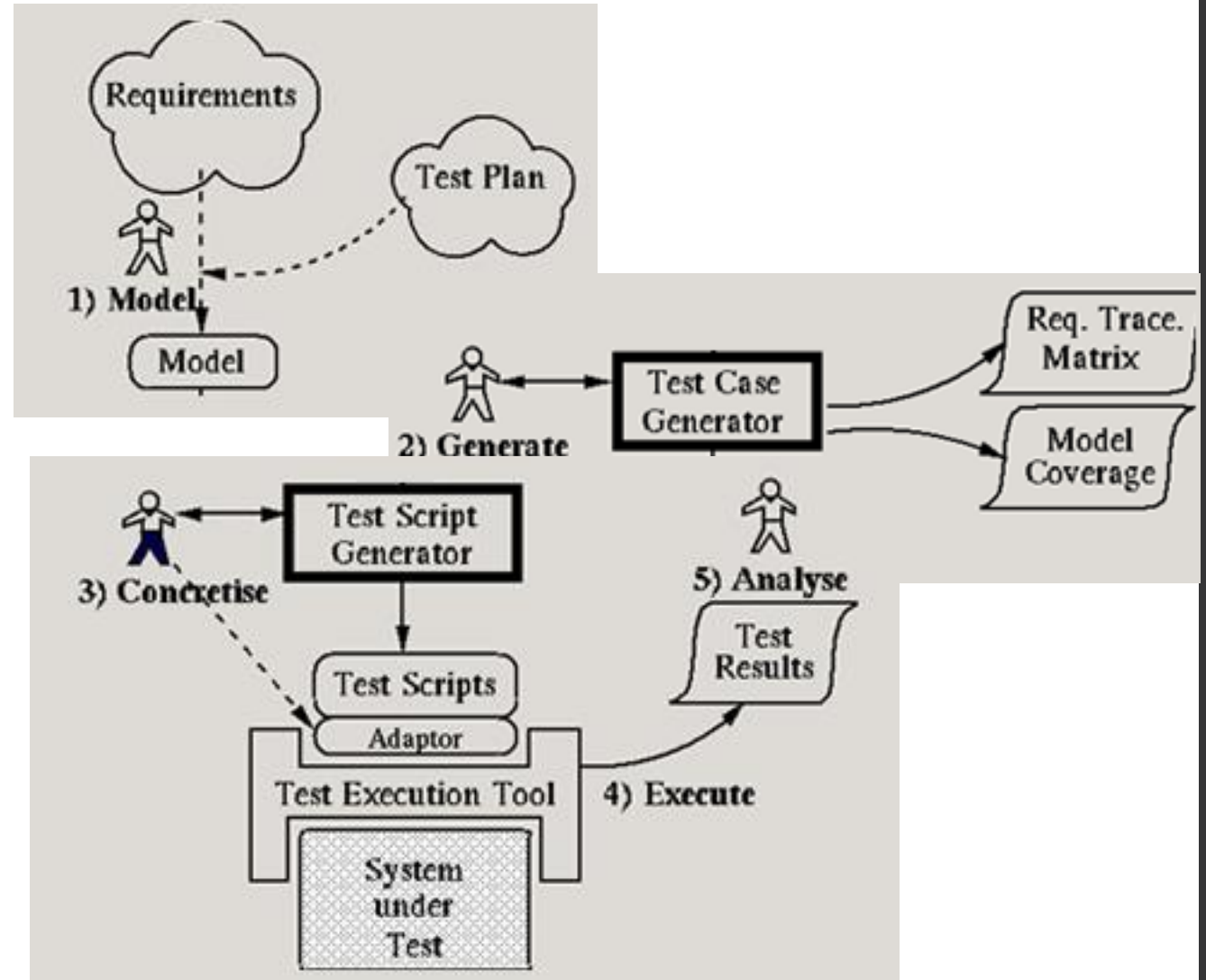


WHERE IS MOST OF YOUR AUTOMATED TESTING TIME SPENT?

Model-based Testing

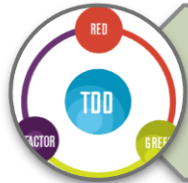


Model-Based Testing (Brambilla et al., 2012)

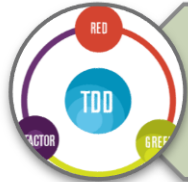


Model-based Testing (Utting, 2010)

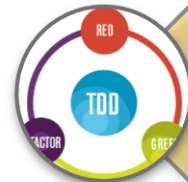
Integración de MBT y TDD



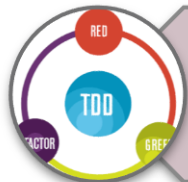
Desarrollo dirigido por tests.



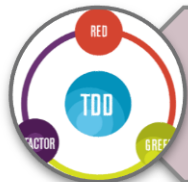
Automatización en la ejecución de tests.



Definición de tests por medio de código y palabras claves.



Muchas líneas de código para tests.



Cambio de requerimientos introduce problemas de mantenimiento.



Mayor abstracción de tests.



Seguimiento de los requerimientos del sistema.



Generación automática o semi-automática de tests.



Reducción de costos de mantenimiento.



Independencia de plataforma.

Estado del Arte

Título	Tipo de Informe	Año	Tipo de Estudio	Objeto de Estudio
Test First Model-Driven Development	Tesis de Msc. en Informática	2012	Presentación de una herramienta y validación mediante casos de estudio.	Se presenta la herramienta TFMDD que permite generar tests y software a partir de un modelado único, basado en restricciones a partir de la definición de pre y post condiciones.
Alignment of requirements specification and testing: A systematic mapping study	Artículo de Conferencia	2011	Mapeo sistemático, incluye el análisis de 35 trabajos seleccionados.	Analiza la relación que guarda el proceso de testing con respecto a las especificaciones de requerimientos funcionales y no funcionales del software.
A survey on model-based testing approaches: a systematic review	Artículo de Workshop	2007	Revisión sistemática de técnicas MBT, se analizan 78 trabajos seleccionados.	Se realiza comparaciones con respecto a aspectos como: modelos utilizados, soporte de herramientas, criterios de cobertura que soportan, niveles de automatización de pasos de testing, complejidad. Sus resultados permiten conocer qué tipos de tests están mayormente cubiertos y cuáles son algunas limitaciones pendientes en el área.
Industrial-strength model-based testing-state of the art and current challenges	Artículo de Workshop	2013	Estudio primario. Utilizando una herramienta existente (RT-Tester) como referencia, se describen aspectos de MBT aplicados en la industria.	Utilizando la herramienta RT-Tester, se ilustran aspectos de MBT en la práctica y métodos que dieron buenos resultados en testing aplicado a problemas del mundo real. El campo de aplicación considerado es: sistemas embebidos de tiempo real aplicados a la aviación, industria automotriz y vías férreas. Las técnicas y métodos presentados pueden ser utilizados como benchmarks para este campo de aplicación.
Towards Testing Future Web Applications	Artículo de Conferencia	2011	Se presenta una metodología que implementa nuevos paradigmas para el testing de webs futuras.	En el marco del proyecto FITTEST se realiza un análisis de las limitaciones de las técnicas de testing actuales para su aplicación sobre sistemas de plataforma web. Se presenta nuevos métodos y paradigmas que pueden ayudar a hacer frente a los problemas mencionados.
Integrating Model-Based Testing in Model-Driven Web Engineering	Artículo de Conferencia	2011	Estudio primario. Se presenta una técnica práctica para aplicar MBT, reutilizando modelos de desarrollo para la generación de tests.	Discusión sobre las ventajas y desventajas de la reutilización de modelos para generación de tests. Se presentan además detalles de web testing. Finalmente, se implementa una técnica para la aplicación de MBT mediante un ejemplo práctico.
Model-Based Testing of Community-Driven Open-Source GUI Applications	Artículo de Conferencia	2006	Estudio primario. Descripción de un proceso ideado para llevar a cabo el testing de aplicaciones elaboradas por una comunidad de desarrollo.	Se describen los problemas de testing en aplicaciones GUI, especialmente en comunidades de desarrollo. Se propone una técnica para llevar a cabo el testing de comunidades Open Source de desarrolladores, interconectados por la WWW.
Model Based Testing in Web Applications	Artículo de Journal	2014	Documento informativo, recopila varios trabajos y se analiza los modelos que utilizan para generar tests de aplicaciones web.	Discusión de conceptos principales y modelos utilizados para generar tests para aplicaciones web, se acompaña la discusión con un caso de estudio ilustrativo.

Estado del Arte: Desafíos

- La integración de técnicas MBT al proceso de desarrollo de *software* basado en TDD.
- Limitación de herramientas MBT integradas para el proceso completo de *testing*.
- Mayor cercanía de *tests* a los requerimientos del *software*.
- Necesidad de reducir los conocimientos requeridos para la utilización de herramientas MBT.

Objetivos

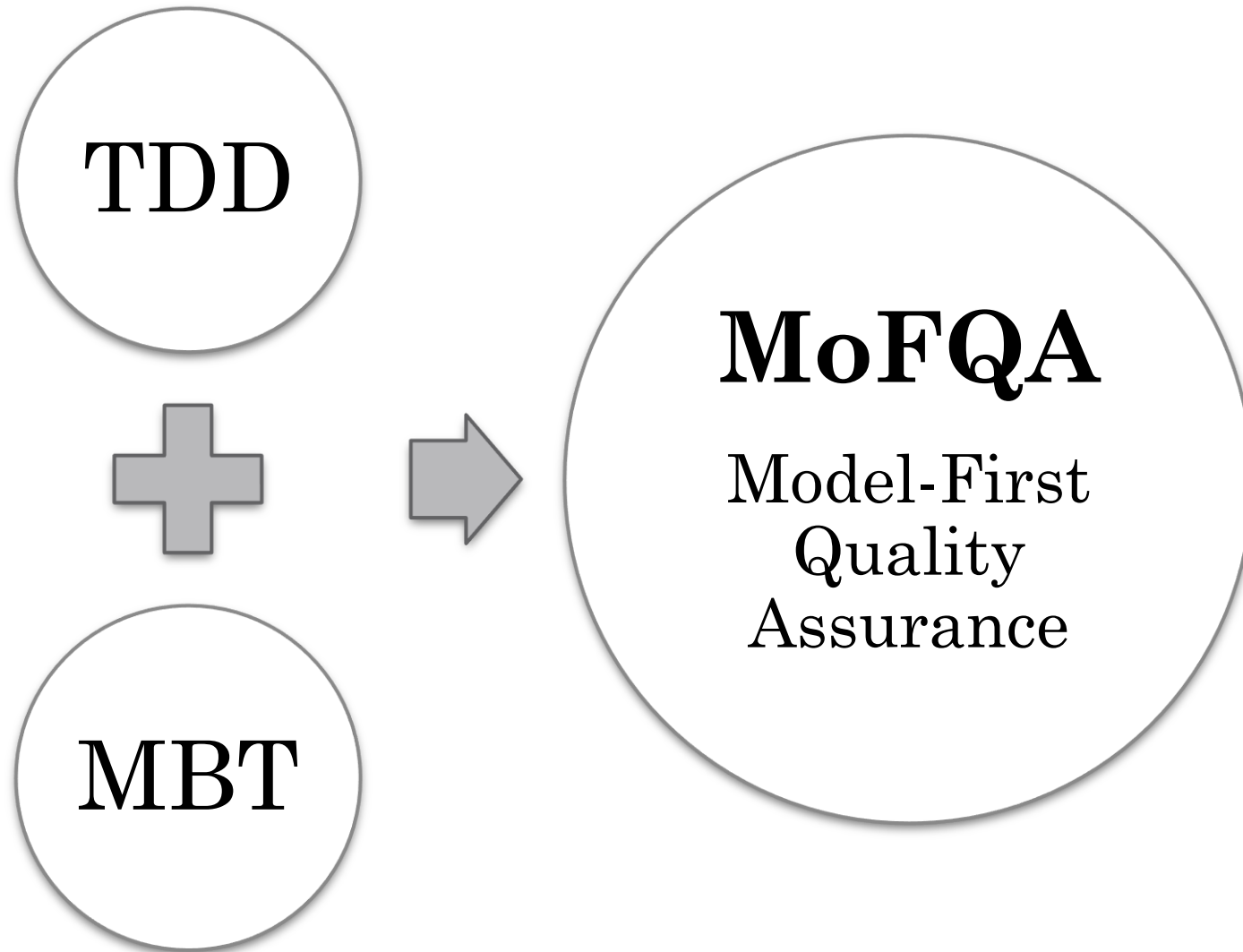
General

- Definición de un método de desarrollo de *software* basado en la creación de *tests* a partir de modelos (apoyados por herramientas MBT) definidos por el usuario y el desarrollador, siguiendo los pasos y prácticas descritas en TDD.

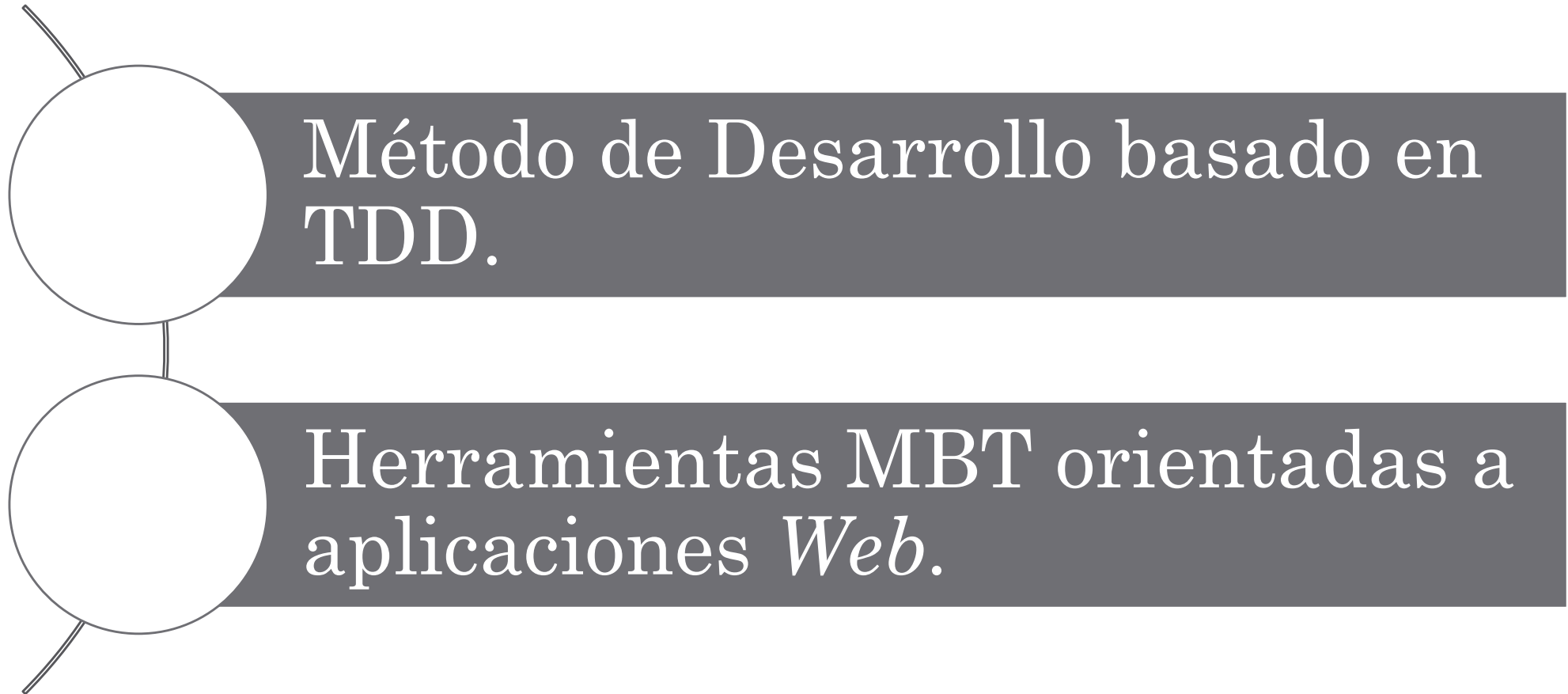
Específicos

- Definición de un método de trabajo que integre los pasos de TDD, siguiendo las prácticas definidas por MBT.
- Definición de perfiles UML para la representación de *tests* unitarios y de aceptación aplicables a sistemas de plataforma *Web*.
- Definición de reglas de transformación para la generación de *tests* ejecutables a partir de modelos UML, enriquecidos con los elementos del perfil.
- Desarrollo de una herramienta para el modelado de *tests* de aceptación, a ser utilizada por usuarios finales de los sistemas a verificar.

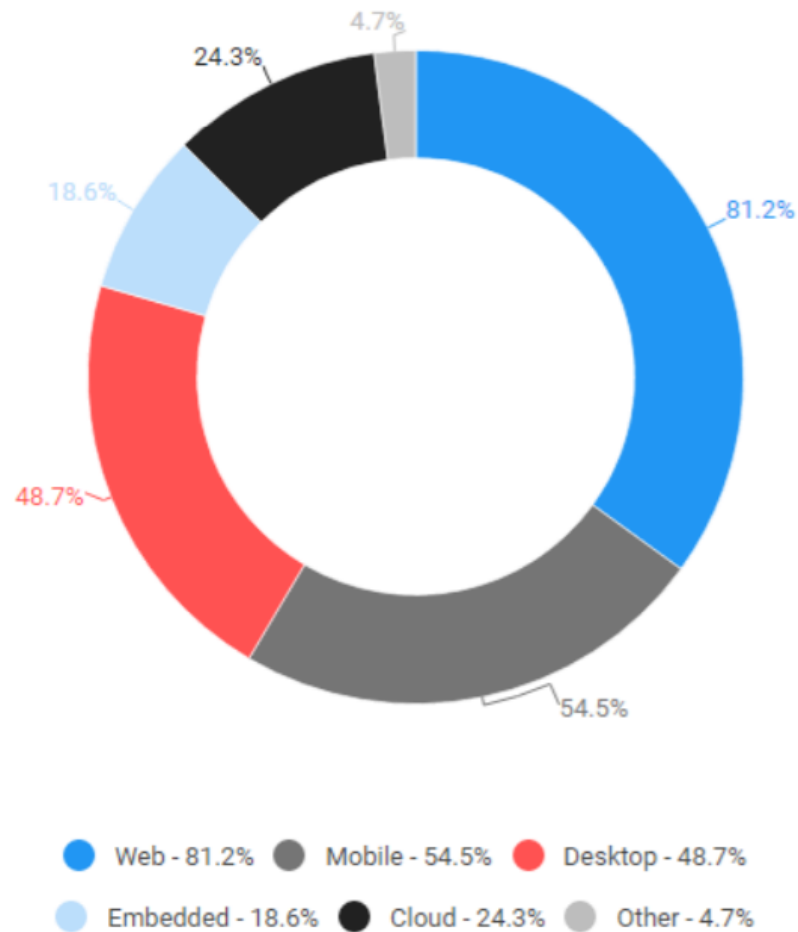
Propuesta: MoFQA



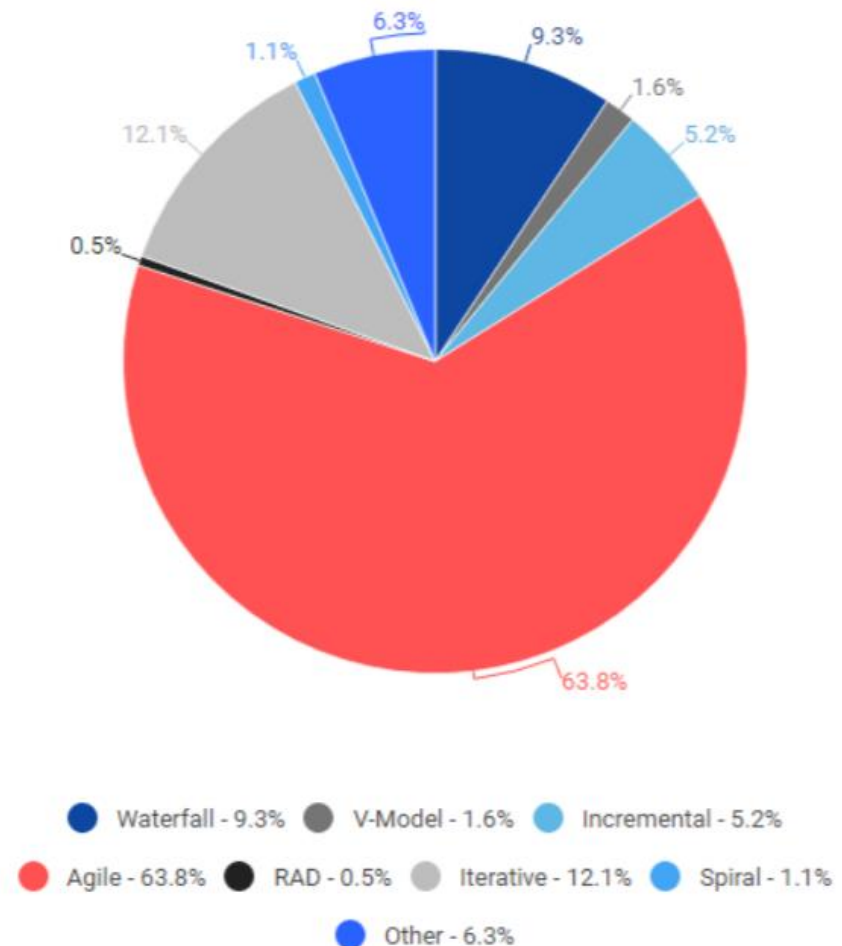
Propuesta: MoFQA



Herramientas MoFQA

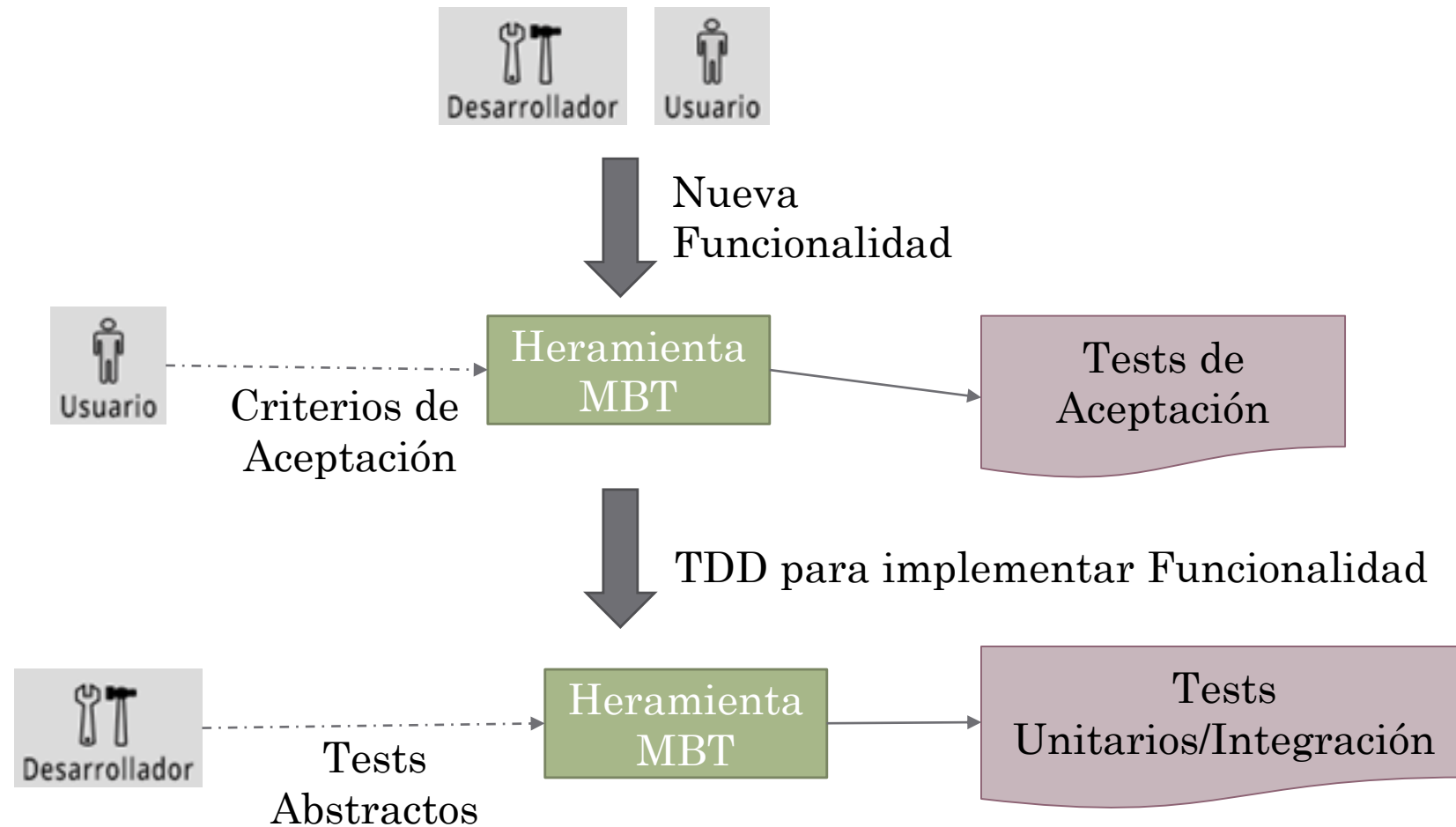


Modern State of Test Automation (*Applied Systems, 2016*)

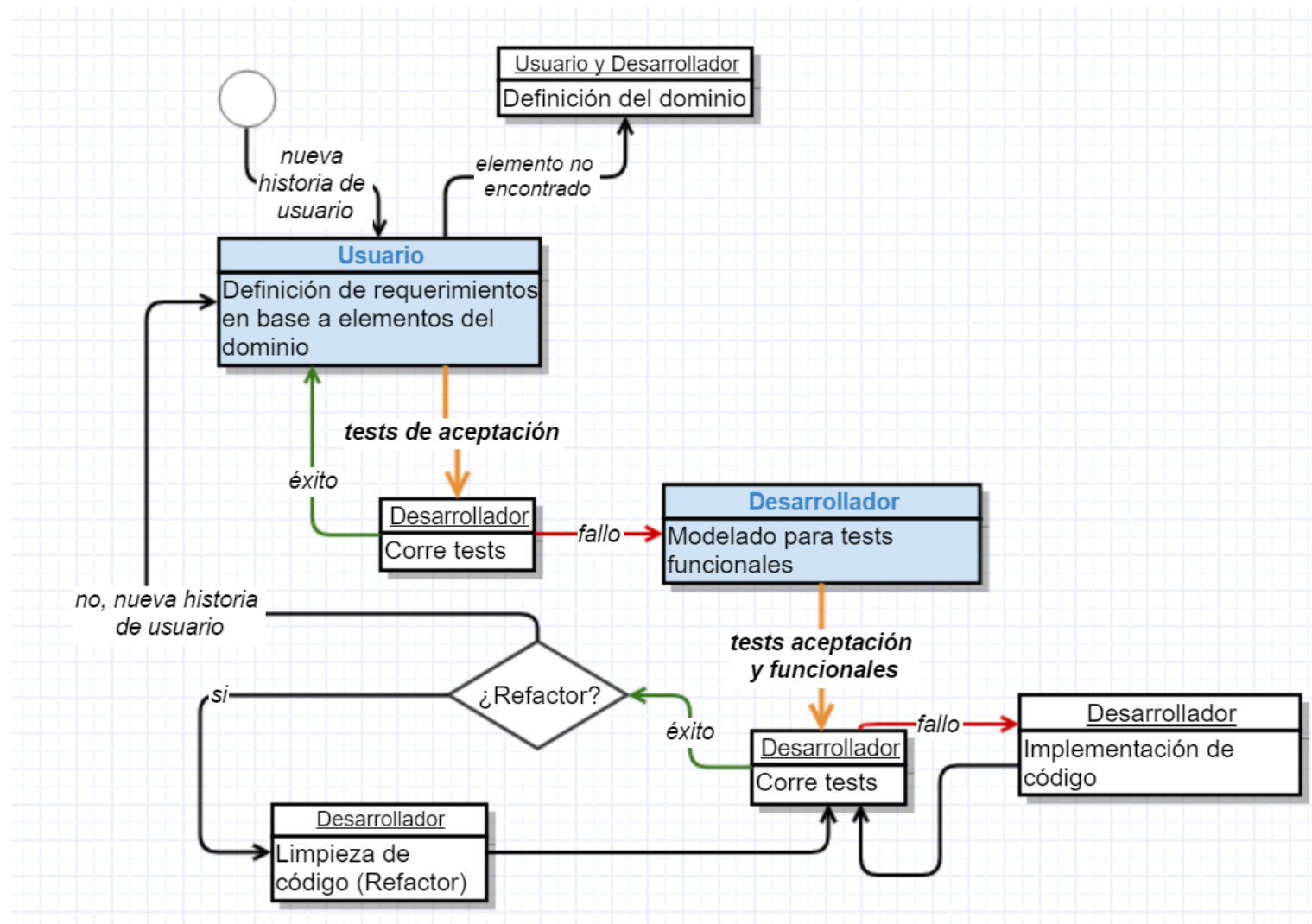


Modern State of Test Automation (*Applied Systems, 2016*)

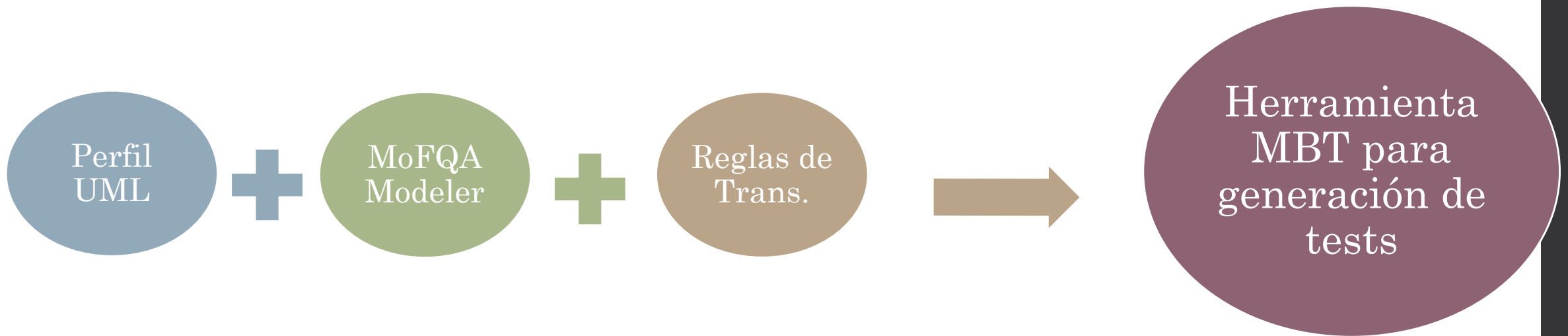
MoFQA: Modelo para el desarrollo de *software*



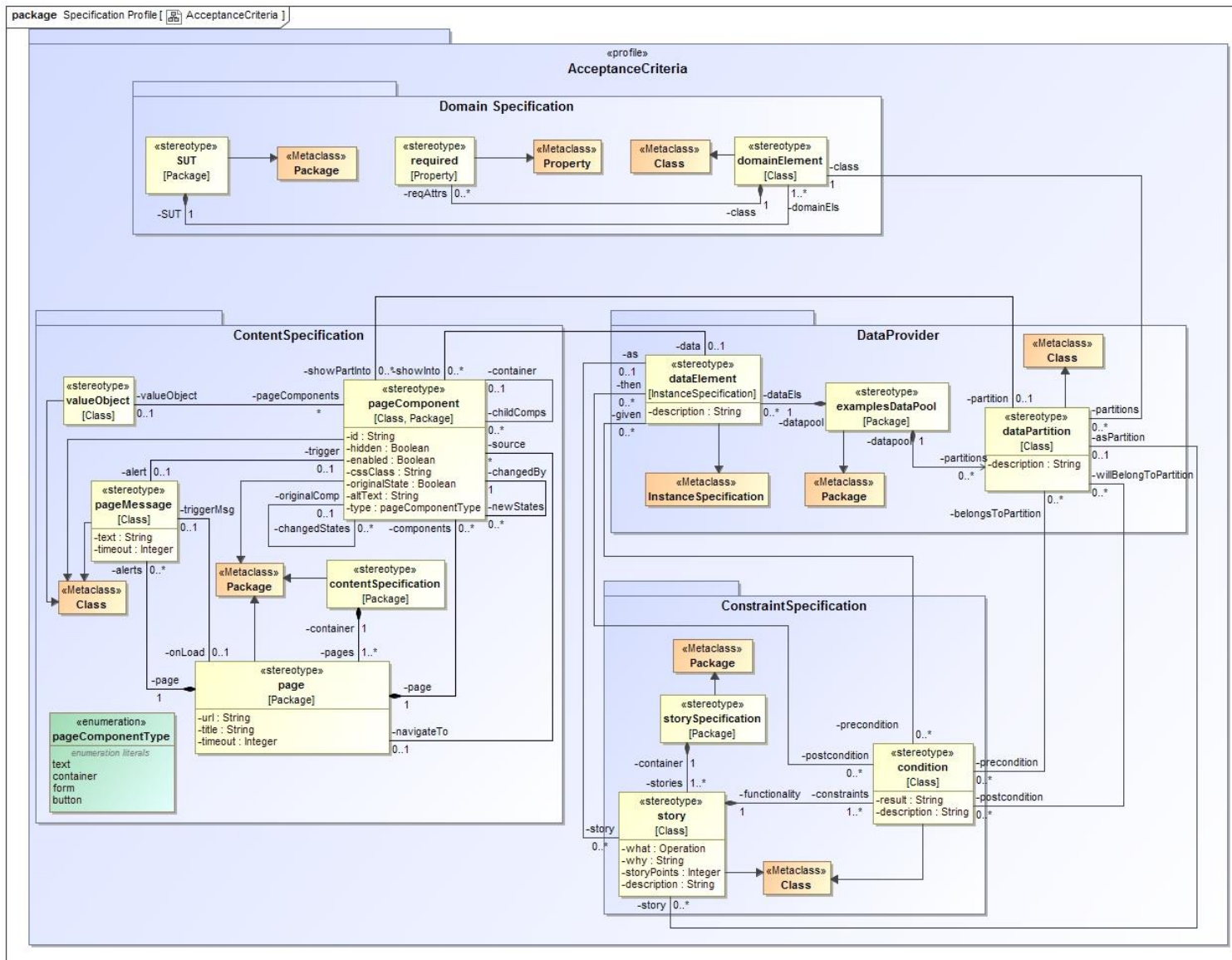
MoFQA: Modelo para el desarrollo de *software*



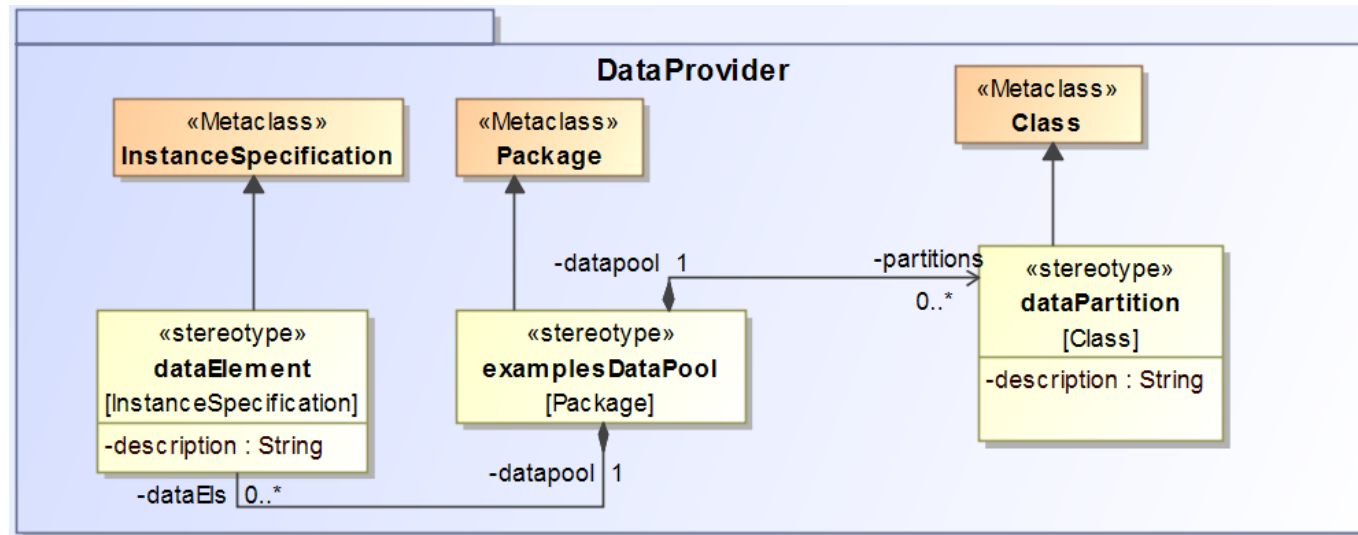
MoFQA: Herramientas MBT



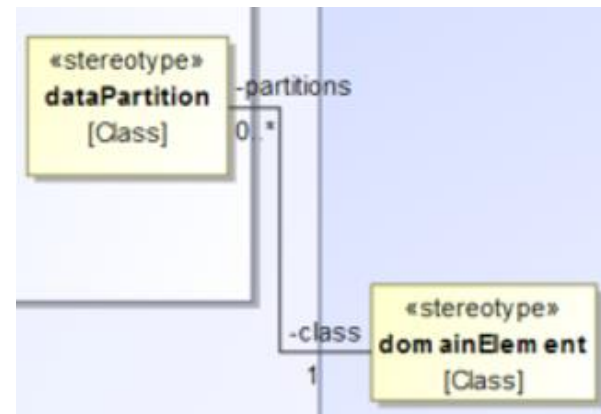
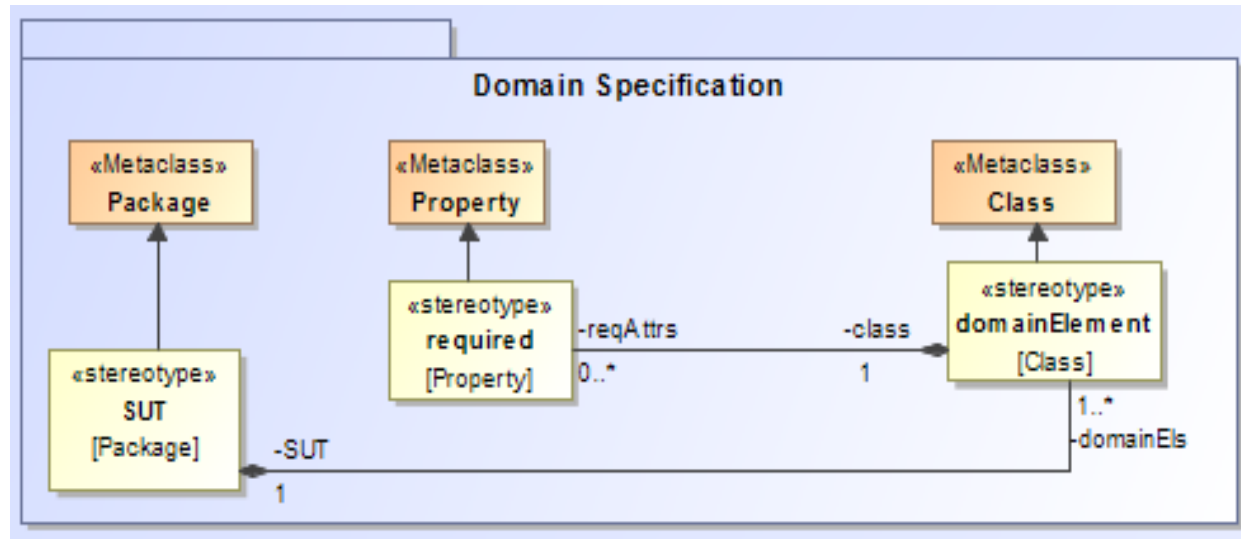
MoFQA: Perfil *Acceptance Criteria*



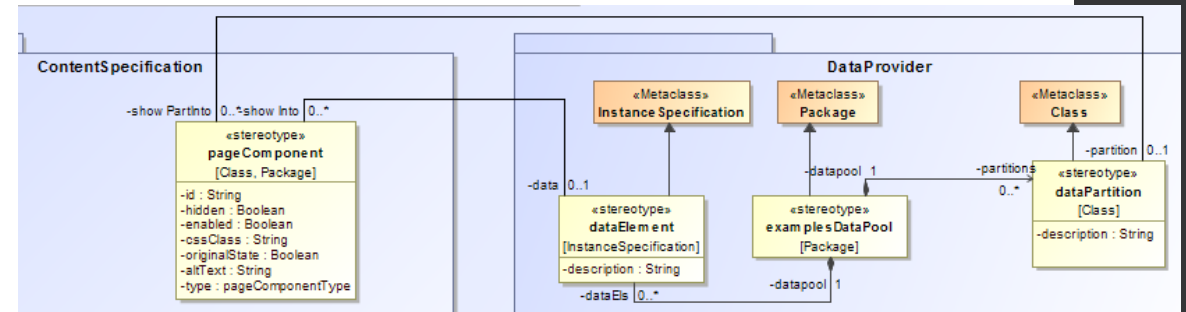
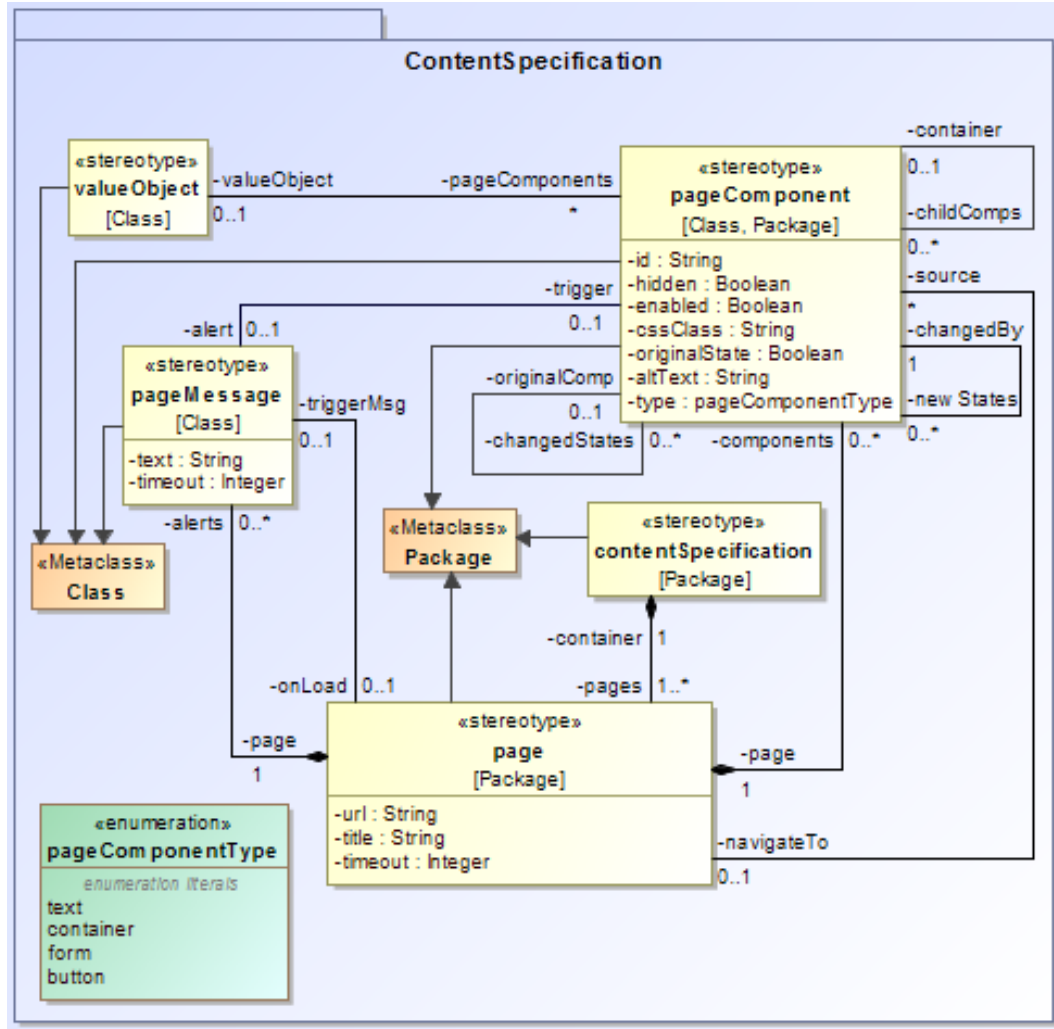
MoFQA: Perfil *Acceptance Criteria*



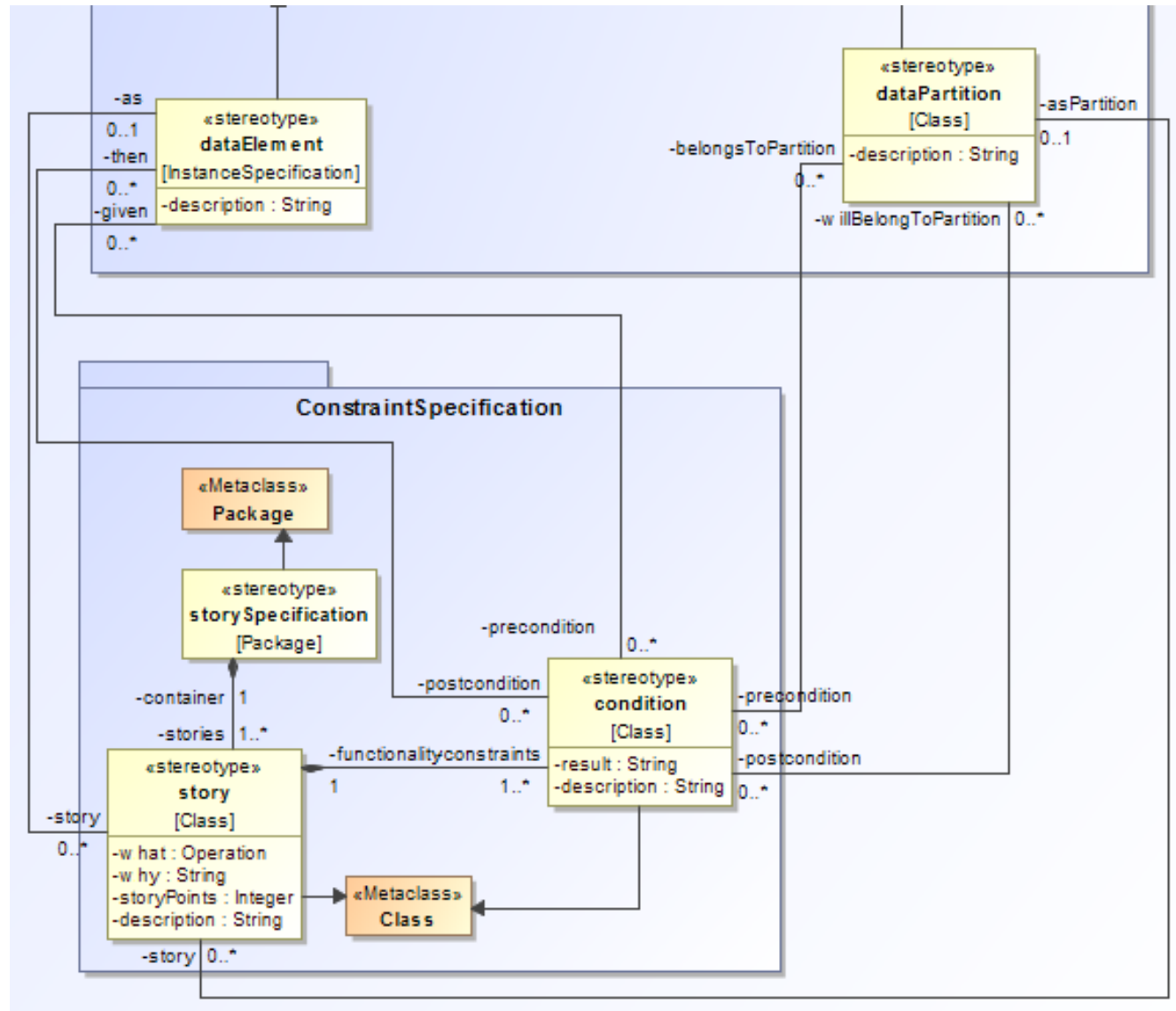
MoFQA: Perfil *Acceptance Criteria*



MoFQA: Perfil *Acceptance Criteria*



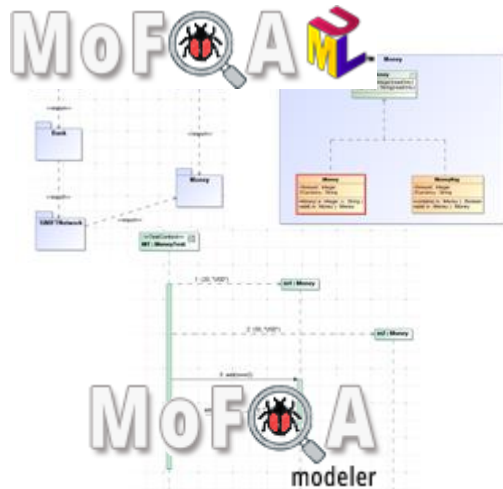
MoFQA: Perfil *Acceptance Criteria*



MoFQA: MoFQA Modeler

- Herramienta de modelado de criterios de aceptación para requerimientos de aplicaciones de plataforma Web.
- Permite modelar:
 - Páginas con sus componentes (texto, campos de formularios, botones, mensajes de alerta).
 - Datos de ejemplo.
 - Resultados de las interacciones.
- Dirigida al usuario final.
- Disponible en <http://www.dei.uc.edu.py/proyectos/mddplus/herramientas/mofqa/modeler/>

MoFQA: Reglas de Transformación



Transformación
M2T



MoFQA
transformation rules

```
package example;\nimport org.testng.annotations.*;\n\npublic class SimpleTest {\n    @Test(description="launches the WordPress site")\n    public void launchSite(){\n        selenium.open("");\n        selenium.waitForPageLoad("30000");\n        assertEquals(selenium.getTitle(), "demo | Just another WordPress\n    }\n\n    @BeforeClass\n    public void setup() {\n        // code that will be invoked\n    }\n\n    @Test(groups = { "Fast" })\n    public void aFastTest() {\n        System.out.println("Fast t\n    }\n\n    @Test(groups = { "slow" })\n    public void aSlowTest() {\n        selenium.open("wp-admin");\n        selenium.waitForPageLoad("30000");\n        assertEquals(selenium.getTitle(), "Demo WP Log In");\n    }\n\n    @Test(description="enters valid login data")\n    public void loginAsAdmin() {\n        selenium.type("user_login", "admin");\n        selenium.type("user_pass", "demo123");\n        selenium.click("wp-submit");\n        selenium.waitForPageLoad("30000");\n        assertTrue(selenium.isTextPresent("howdy, admin"));\n    }\n\n    @Test(description="navigates to the New Post screen")\n    public void navigateNewPost() {\n        selenium.click("//a[contains(text(),'Posts')]//following::a[contains\n        selenium.waitForPageLoad("30000");\n        assertTrue(selenium.isTextPresent("Add New Post"));\n    }\n}
```



Tests: 0/1. Methods: 7/7 (52324 ms)

- Knorrium.info - Wordpress Demo (7/0/0/0)
- Write new post (7/0/0/0)
- test.Wordpress.launchSite()
- test.Wordpress.openAdminPage()
- test.Wordpress.loginAsAdmin()
- test.Wordpress.navigateNewPost()
- test.Wordpress.writeBlogPost()
- test.Wordpress.publishBlogPost()
- test.Wordpress.verifyBlogPost()

Command	Target	Value
open	/	selenium
type	q	selenium
clickAndWait	btnG	
clickAndWait	link=Selenium web applic...	
assertTextPresent	elvis	

Log Reference: UI-Element Rollup

```
[info] Executing: |open | / | |\n[info] Executing: |type | q | selenium | |\n[info] Executing: |clickAndWait | btnG | |\n[info] Executing: |clickAndWait | link=Selenium web application\n  testing system | |\n[info] Executing: |assertTextPresent | elvis | |\n[error] false
```

Validación

- Testing utilizando las herramientas **MoFQA** vs *testing* completamente manual
- Utilidad de la herramienta **MoFQA Modeler**.

Validación 1

- Se llevó a cabo una primera experiencia, en una clase de laboratorio con alumnos del segundo semestre de las carreras de Ing. Informática e Ing. Electrónica de la Universidad Católica “Nuestra Señora de la Asunción”.
- Objetivos:
 - Comparación entre el *testing* manual y el *testing* usando MoFQA Modeler. Parámetros de comparación:
 - Tiempo requerido para la definición y ejecución de *tests*.
 - Cantidad de verificaciones realizadas en cada proceso de *testing*.
 - Medir usabilidad de la herramienta MoFQA Modeler

Validación 1

- 24 alumnos participantes.
- Los 3 trabajos más completos fueron seleccionados para el análisis

	Requerimiento	Sub-Req.	T. Manual	T. Mod.	Cob. M.	Cob. G.	TM	TG	LOC	SUS
1	Mi Escritorio	7	4 min.	5 min.	100%	100%	9	22	109	67.5
2	Texto bienvenida (usuario no autenticado)	2	5 seg.	3 min.	100%	100%	2	6	73	82.5
3	Login no válido	3	6 min.	5 min.	100%	100%	9	15	341	47.5

Validación 1: Resultados

Métrica para MoFQA Modeler	Resultado
Cobertura de requerimientos	100%
Promedio de verificaciones adicionales	8
Promedio de modelado por requerimiento	4.3 minutos
Valor SUS Promedio	65.8

Validación 2

- Ilustración mediante el modelado de requerimientos para el portal Web Amazon.es.
- Objetivos:
 - Comparar el tiempo de modelado con MoFQA Modeler vs el modelado usando el perfil y un editor UML.
 - Verificar cobertura de requerimientos utilizando las herramientas MoFQA.

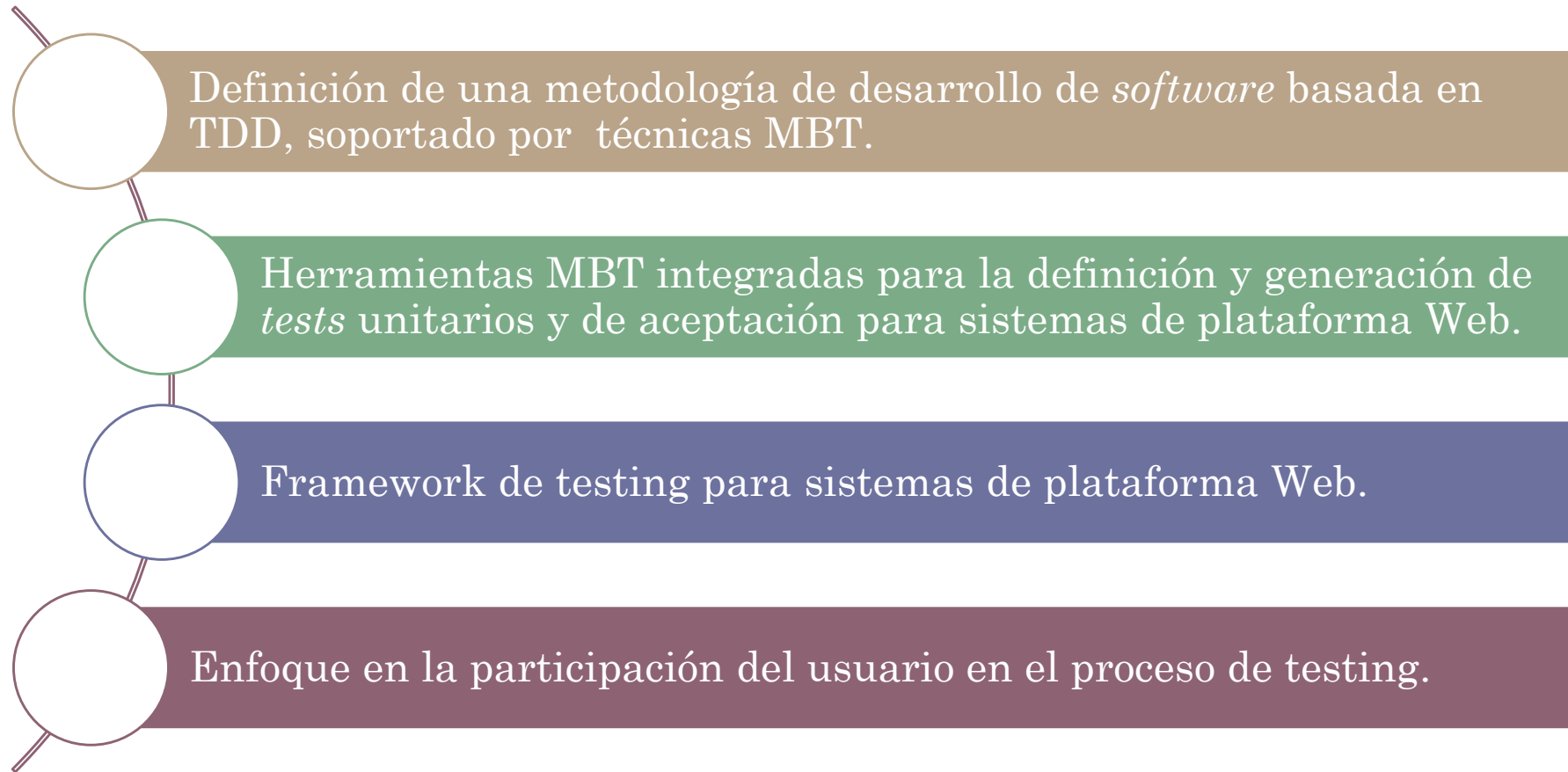
Validación 2

Requerimientos	Perfiles UML y MagicDraw	MoFQA Modeler
R1	18 min.	5 min.
R2	19 min.	6 min.
R3	20 min.	8 min.
R4	14 min.	3 min.
R5	14 min.	3 min.


Validación 2: Cobertura de requerimientos

Requerimiento	Sub-requerimientos	Cant. Verif.
R1: elementos en la Home no autenticada	Carga de la página	18
	Debe aparecer el texto "Bienvenido"	
	Debe aparecer el texto "Hola, identifícate"	
	Debe aparecer el enlace "Todos los departamentos"	
	Debe mostrarse un botón "Inicia sesión de forma segura"	
R2: "Todos los departamentos"	Enlace "Todos los departamentos" re-direcciona a una lista de departamentos y sub-departamentos.	9
R3: Elementos de inicio de sesión	Presencia del texto "Iniciar sesión"	8
	Formulario con campo e-mail	
	Formulario con campo contraseña	
	Botón con texto "Iniciar sesión"	
R4: Inicio de sesión no válido	Intento de inicio de sesión con usuario no válido debe actualizar el texto "Iniciar sesión" en pantalla por "Inténtalo nuevamente"	13
R5: Inicio de sesión válido	Inicio de sesión con usuario válido re-direcciona a la Home.	19
	Se visualiza el texto "Hola,"	
	El nombre del usuario que inició sesión aparece tras el texto "Hola,"	

Contribuciones

- 
- Definición de una metodología de desarrollo de *software* basada en TDD, soportado por técnicas MBT.
 - Herramientas MBT integradas para la definición y generación de *tests* unitarios y de aceptación para sistemas de plataforma Web.
 - Framework de testing para sistemas de plataforma Web.
 - Enfoque en la participación del usuario en el proceso de testing.

Trabajos Futuros



Definición y ejecución de un experimento o caso de estudio para validar el modelo de desarrollo de *software* propuesto.

Evaluar en forma más rigurosa la usabilidad de la herramienta, a través de experiencias adicionales.

Generalización de las reglas de transformación de forma a que los *tests* no estén únicamente orientados a la plataforma destino seleccionada en este trabajo.

Extensión de perfiles UML y las reglas de transformación para la definición de *tests* unitarios más enriquecidos.

Inclusión de nuevos elementos en los perfiles y reglas de transformación para la definición de *tests* de integración.

Extensión de herramientas de modelado y reglas de transformación para otros dominios de desarrollo, no necesariamente de plataformas *Web*.



¡Gracias por la
atención!

¿Preguntas?

Linda Riquelme

linda.riquelme@uc.edu.py