

# Pruebas en el contexto de un proyecto

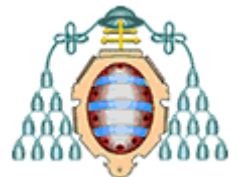
## Pruebas y Despliegue del Software

Grupo de Investigación en Ingeniería del Software -

<http://giis.uniovi.es>



Curso 2023-2024



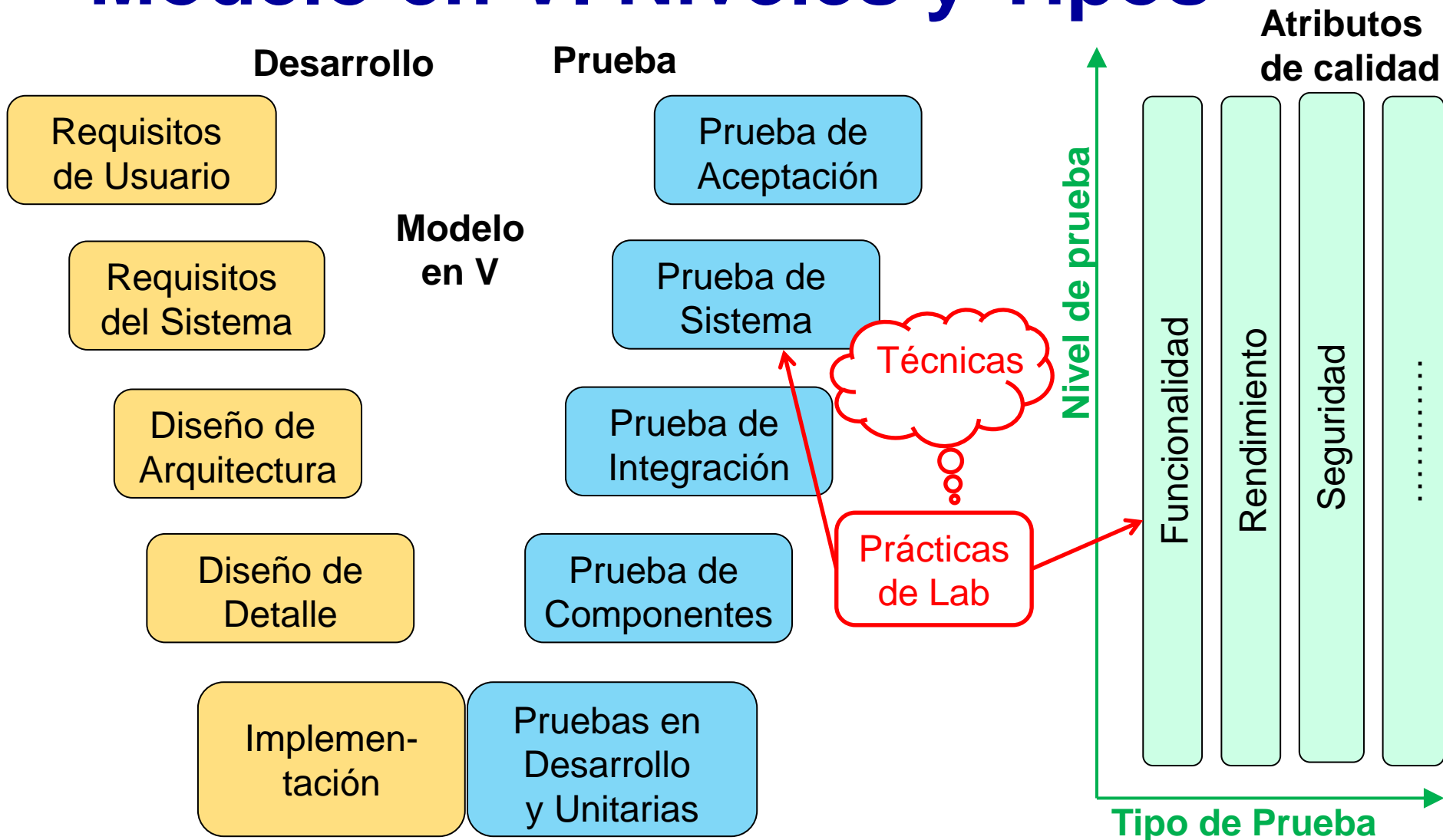
# Contenidos

- **Proceso**
  - Modelo en V
  - Niveles y Tipos de Pruebas
  - TDD
  - Procesos genéricos
- **Herramientas y Automatización**
- **Certificaciones, Metodologías y Estándares**

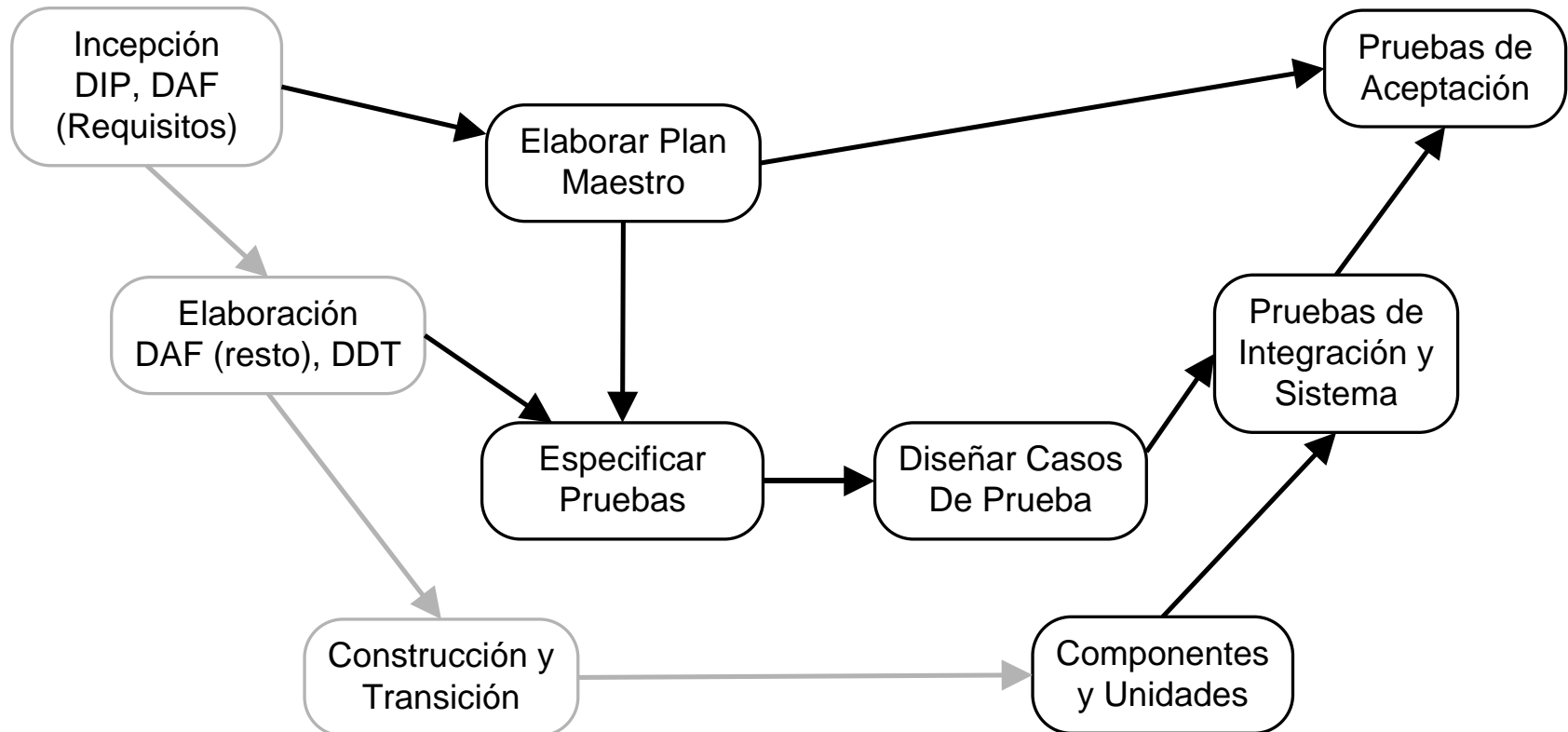
# Modelo en V. Niveles y Tipos (recordamos SINP)

- Las técnicas son una ayuda esencial para identificar situaciones a cubrir y derivar los casos de prueba
- Pero, cuál va a ser el alcance de las pruebas en un contexto dado? (p.e. en las prácticas de laboratorio de PDS)
  - ¿En qué momento del desarrollo del sistema vamos a probar? -> Niveles de prueba: modelo en V
  - ¿Qué tipo de características del software vamos a probar? -> tipos de prueba

# Modelo en V. Niveles y Tipos



# Modelo en V (metodología METESPA)



# Niveles y Tipos de Pruebas

- Nivel de prueba: Grupo de actividades de prueba organizadas y gestionadas en conjunto [Tmap, ISTQB]
  - En el mismo momento.
  - Ej. Integración, sistema, aceptación.
- Tipo de prueba: Grupo de actividades de prueba para un componente o sistema enfocadas en un objetivo específico de prueba [Tmap, ISTQB]
  - Un tipo de prueba puede utilizarse en uno o más niveles de prueba
  - Característica de calidad: Ej. Funcional, rendimiento, etc.
  - Relativo a cambios: Ej. Regresión

# Niveles de prueba

- Componentes: por separado, principalmente funcionalidad.
- Integración: interfaces entre componentes, interacciones con otras partes del sistema
  - A nivel de componentes
  - A nivel de sistema
- Sistema: comportamiento del sistema/producto global, funcional y no funcional, suele ser el último paso de verificación
- Aceptación: Determinar si el sistema está listo para ser liberado.  
Validación
  - Aceptación de usuario: principalmente funcional
  - Aceptación operacional: recuperación de desastres, tareas de mantenimiento, etc.
  - FAT (Factory Acceptance Testing) y SAT (Site Acceptance Testing)
  - En COTS: alpha/beta

# Pruebas de componentes vs integración

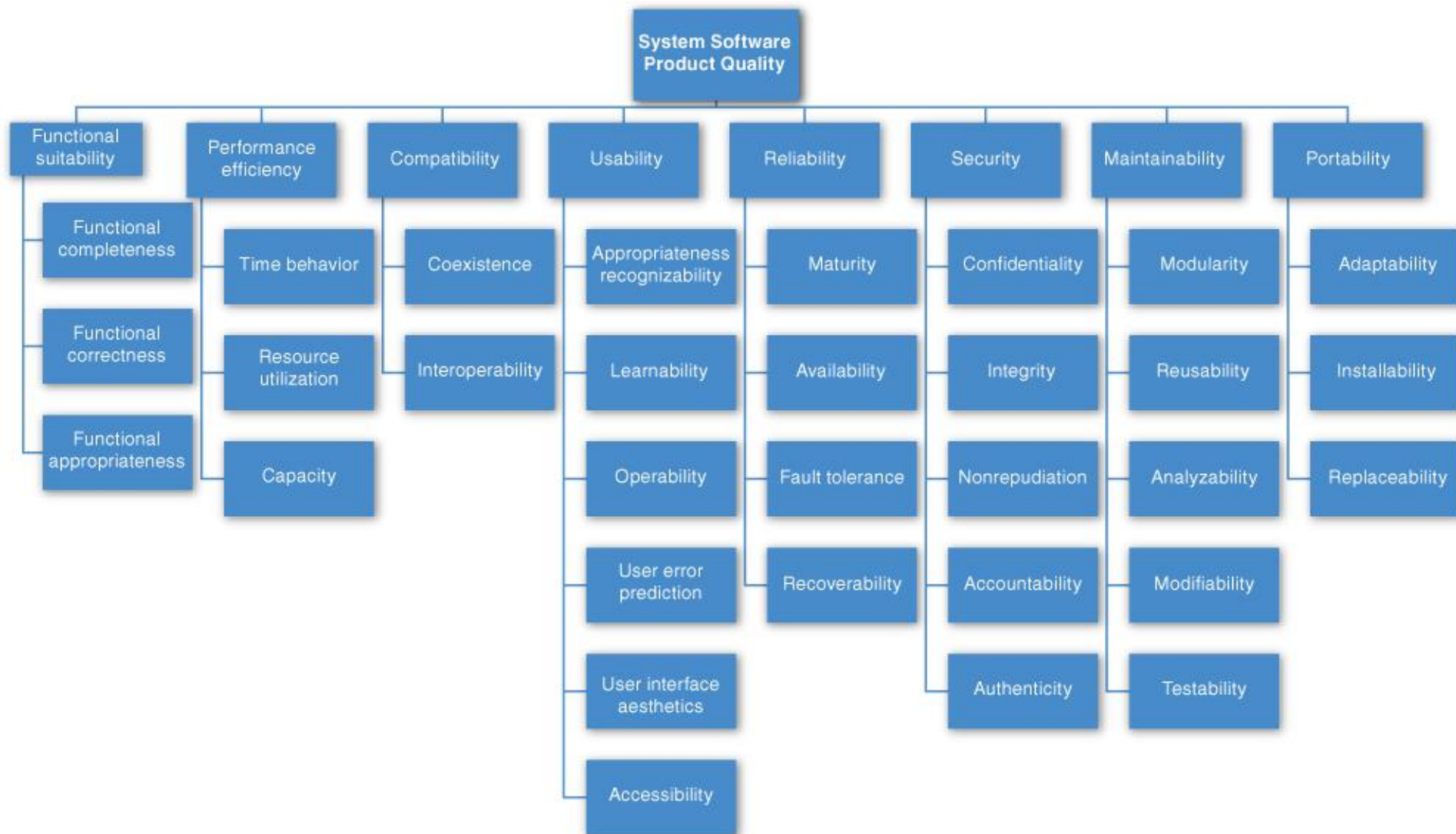




# Tipos de Pruebas

- Funcionales
- No funcionales
  - Interoperabilidad
  - Seguridad
  - Rendimiento, Carga, estrés
  - Usabilidad, Accesibilidad
  - Fiabilidad (tolerancia a fallos, recuperación), Eficiencia, Portabilidad
- Relativas a cambios
  - Confirmación (retest): Defectos han sido solucionados
  - Regresión: Los cambios no han afectado a otras partes

# ISO/IEC 25010: System and software quality models



# ISO/IEC/IEEE 29119: Tipos de prueba (extracto)

Type of Testing	Quality Characteristic	Sub-Characteristics
Accessibility Testing	Usability	Accessibility
Backup/Recovery Testing	Reliability	Maturity
		Fault tolerance
		Recoverability
Compatibility Testing	Compatibility	Co-existence
Conversion Testing	Functional Suitability	Functional completeness
		Functional correctness
		Functional appropriateness
Disaster Recovery Testing	Reliability	Maturity
		Fault tolerance
		Recoverability
Functional Testing	Functional Suitability	Functional completeness
		Functional correctness
		Functional appropriateness
Installability Testing	Portability	Installability
Interoperability Testing	Compatibility	Interoperability
Maintainability Testing	Maintainability	Modularity
		Reusability
		Analysability
		Modifiability
		Testability
Performance-Related Testing	Performance efficiency	Time-behaviour
		Resource utilisation
		Capacity
Portability Testing	Portability	Adaptability
		Installability

# Ejemplo: Pruebas de rendimiento

- Planificar y Analizar
  - Definir diferentes escenarios de prueba
  - Infraestructura y objetivo de la prueba
- Diseñar (Elaborar scripts para ejecución automática)
  - Registrar sesiones
  - Variar número de usuarios y escenarios que ejecutan
  - Tiempo de espera entre cada acción (thinking time)
- Ejecutar (Introducir carga en el sistema, no olvidar datos)
  - Límites del sistema (aumentar usuarios hasta saturación/fallo)
  - Carga continuada (a un porcentaje del límite)
  - Estrés (aumento muy rápido de usuarios)
- Medir
  - Tiempos de respuesta, memoria, cursores abiertos
  - Fallos encontrados

# Desarrollo Dirigido por las Pruebas

- Test Driven Development (TDD)
  - Definir las pruebas antes de codificar
  - Muy incremental: Ir codificando para ir pasando las pruebas
  - Regresión y refactorización
- Típico en métodos ágiles, ciclos muy cortos
  - Requiere disciplina
  - Cambio de forma de pensar de los programadores
  - Requiere automatización
  - Muy útil en pruebas unitarias

# Procesos Genéricos

- Planificar
  - Estimar
  - Realizar **plan**
- Preparar
  - Especificación y Diseño de las pruebas
- Realizar
  - Ejecución
  - Reporting (reporte)
- Monitorizar y Controlar
  - Medir
  - Aplicar acciones

# Plan de pruebas

- Resumen del contenido (ISO/IEC/IEEE 29119):
  - Contexto de la prueba: identificar proyecto y partes interesadas, test ítems y características a ser probadas
  - Comunicación entre testing y resto de organización
  - Riesgos (de producto y de proyecto)
  - Estrategia de prueba:
    - Niveles y tipos de prueba que se realizarán, entregables, técnicas a utilizar, criterios de finalización, métricas.
    - Requisitos del entorno de test y de los datos.
    - Pruebas de regresión y retest. Criterios de suspensión/reanudación.
  - Actividades de prueba, estimaciones, personal (roles y responsabilidades), necesidades de formación y contratación
  - Planificación temporal

# ISO/IEC/IEEE 29119:

## Procesos de gestión





# Automatización y Herramientas

- Valorar **rentabilidad**
- Generales:
  - Básicas: Scripts SQL, Excel, Macro HojaPruebas
  - Enterprise Architect
  - Suites comerciales: HP Mercury, IBM Rational, Borland
- Específicas
  - Gestión requisitos (comerciales)
  - Gestión defectos (Bugzilla, Mantis, Jira, GForge)
  - Funcionales (Record/Playback). Web: Selenium, Canoo WebTest
  - Código/BD/Servicios (JUnit/DBUnit/soapUI)
  - Cobertura Código (Clover)
  - Análisis Estático (FindBugs, SonarQube)
  - No funcionales: Accesibilidad (TAW), Seguridad (Security Compass), Rendimiento (Jmeter, OpenSTA)

## The Cost of Automation

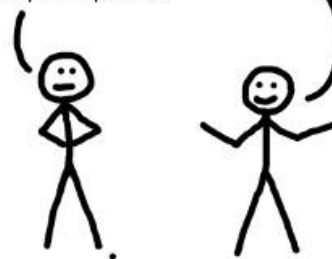
How much money have you spent on this test automation?

Just \$200,000.



Oh, so how many manual testing hours have you replaced?

None



hmmm, how many bugs have you found?

None

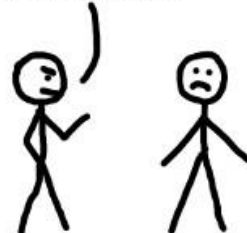


What are you doing then?

Leveling risks.



If you want to level the risk on being unemployed, then go and find some bugs!!



Cartoon designed by Raimond Sinivee

AG

Andy Glover [cartoontester.blogspot.com](http://cartoontester.blogspot.com) Copyright 2011

# Certificaciones, Metodologías y Estándares

- Certificaciones
  - Emitidas por organizaciones independientes
  - Cada una puede tener diversos niveles
- Certificaciones de Testing (Europa)
  - ISTQB (Intl. Software Testing Qualifications Board)
  - TMap Next® - Business Driven Test Management
- Metodologías
  - TMap Next®
  - METESPA: Metodología del Proceso de Pruebas del Principado
- Estándares:
  - IEEE 829 y 2008, BSI 7925
  - ISO/IEC 29119 – Software Testing