Calidad, Validación y Verificación del Software

Conceptos Generales

Grado en Ingeniería Informática del Software

Javier Tuya, Raquel Blanco
Grupo de Investigación en Ingeniería del Software
http://giis.uniovi.es

Curso 2025-2026

Contenido

- Historias de fallos
- Causas de los defectos en el software
- Calidad. Contexto y cifras
- Verificación y Validación
- Pruebas. Definiciones
- Proceso de pruebas dinámicas
- Técnicas de prueba

¿Por qué son las pruebas necesarias?

Porque todos cometemos errores

Historias de fallos

ARIANE 5, Flight 501 Failure

Historias de fallos

ARIANE 5, Flight 501 Failure

“Consequently the **realignment function was not tested under simulated Ariane 5 flight conditions**, and the design error was not discovered”.

“The extensive reviews and tests carried out during the Ariane 5 Development Programme **did not include adequate analysis and testing** of the inertial reference system or of the complete flight control system, which could have detected the potential failure”.

Report by the Inquiry Board, Paris, 19 July 1996

The Chairman of the Board: Prof. J. L. LIONS

Historias de fallos

Contractors point fingers over ObamaCare botch, blame gov't for poor Testing. *FoxNews.com, October 24, 2013*

Obamacare Crashes Months in Coming Not Easily Repaired. Bloomberg, October 22, 2013

“officials **failed to complete exhaustive testing** of the program’s website in a push to begin signups by Oct. 1, according to people involved in the rollout”

Contractors: More testing of HealthCare.gov was needed. IT World, October 24, 2013

Historias de fallos

Software Bug Halts F-22 Flight

"The new US stealth fighter, the F-22 Raptor, was deployed for the first time to Asia earlier this month. On Feb. 11, twelve Raptors flying from Hawaii to Japan **were forced to turn back** when **a software glitch crashed all of the F-22s' on-board computers as they crossed the international date line**… if they had not been w

https://it.slashdot.org/story/07/02/25/2038217/software-bug-halts-f-22-flight

Historias de fallos

Software Bug Halts F-22 Flight

"The
first
from
glitc
cros
https
22-fl

International Date Line bug caused navigation problems

"The F-22A Raptors reportedly had to turn round and return to Hawaii using only visual contact with their tankers

…The Raptors returned safely, but the situation may have been disastrous if they had not been with their tankers or the weather had turned bad."

http://www.f-22raptor.com/news_view.php?nid=267

Historias de fallos

Software Bug Halts F-22 Flight

"The
first
from
glitc
cros
https
22-fl

International Date Line bug caused navigation problems

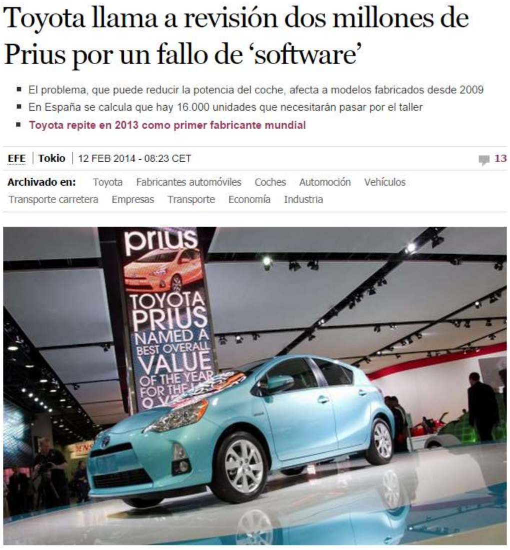
"Th
Ha
…T
dis
had
http

F-22 Raptor Cancelled.

"Slate reports that the F-22 Raptor has been cancelled by the Senate. At an estimated price tag of **$339 million per aircraft**, even the powerful military-industrial-congressional complex couldn't keep this Cold War program alive in these hard times."

https://tech.slashdot.org/story/09/07/22/1813236/f-22-raptor-cancelled?sdsrc=rel

Historias de fallos



Toyota llama a revisión dos millones de Prius por un fallo de 'software'

- El problema, que puede reducir la potencia del coche, afecta a modelos fabricados desde 2009
- En España se calcula que hay 16.000 unidades que necesitarán pasar por el taller
- Toyota repite en 2013 como primer fabricante mundial

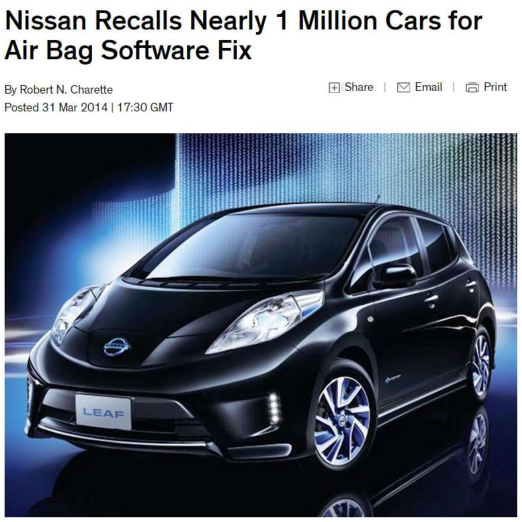
EFE | Tokio | 12 FEB 2014 - 08:23 CET 13

Archivado en: Toyota Fabricantes automóviles Coches Automoción Vehículos Transporte carretera Empresas Transporte Economía Industria

El País (12/2/2014)

Toyota Motor llamará a revisión más de 1,9 millones de unidades de su modelo Prius en todo el mundo por un problema de software que puede reducir la potencia del coche o, en el peor de los casos, hacer que se detenga, según ha explicado la compañía.

Historias de fallos



Nissan Recalls Nearly 1 Million Cars for Air Bag Software Fix

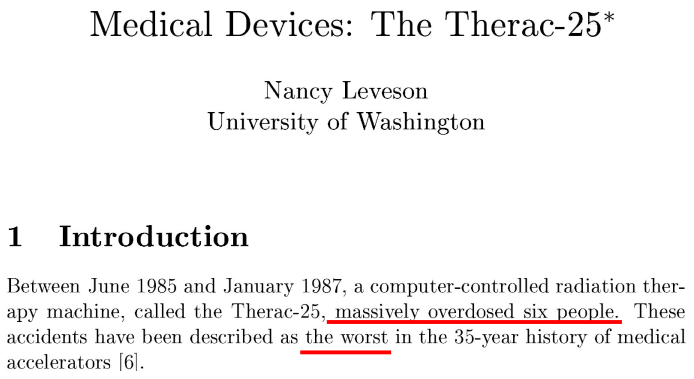
By Robert N. Charette
Posted 31 Mar 2014 | 17:30 GMT

IEEE Spectrum (32/3/2014)

"Unfortunately, the software installed on the vehicles…may incorrectly determine that the passenger seat is empty when it is, in fact, occupied. If that were to happen, and if the vehicle were subsequently involved in an accident, the passenger-seat airbags would fail to deploy, increasing the possibility of injury or death."

A *New York Times* article says that, "The automaker blamed the sensitivity of the software calibration, particularly when 'a combination of factors such as high engine vibration at idle when the seat is initially empty and then becomes occupied' or an 'unusual' seating posture are factors."

Historias de fallos



Medical Devices: The Therac-25*

Nancy Leveson
University of Washington

1 Introduction

Between June 1985 and January 1987, a computer-controlled radiation therapy machine, called the Therac-25, massively overdosed six people. These accidents have been described as the worst in the 35-year history of medical accelerators [6].

http://sunnyday.mit.edu/papers/therac.pdf

Medical Devices: The Therac-25*

Inadequate Software Engineering Practices. Some basic software engineering principles that apparently were violated in the case of the Therac-25 include the following:

- Software specifications and documentation should not be an afterthought.
- Rigorous software quality assurance practices and standards should be established.
- Designs should be kept simple and dangerous coding practices avoided.
- Ways to detect errors and and get information about them, such as software audit trails, should be designed into the software from the beginning.
- The software should be subjected to extensive testing and formal analysis at the module and software level; system testing alone is not adequate. Regression testing should be performed on all software changes.
- Computer displays and the presentation of information to the operators, such as error messages, along with user manuals and other documentation need to be carefully designed.

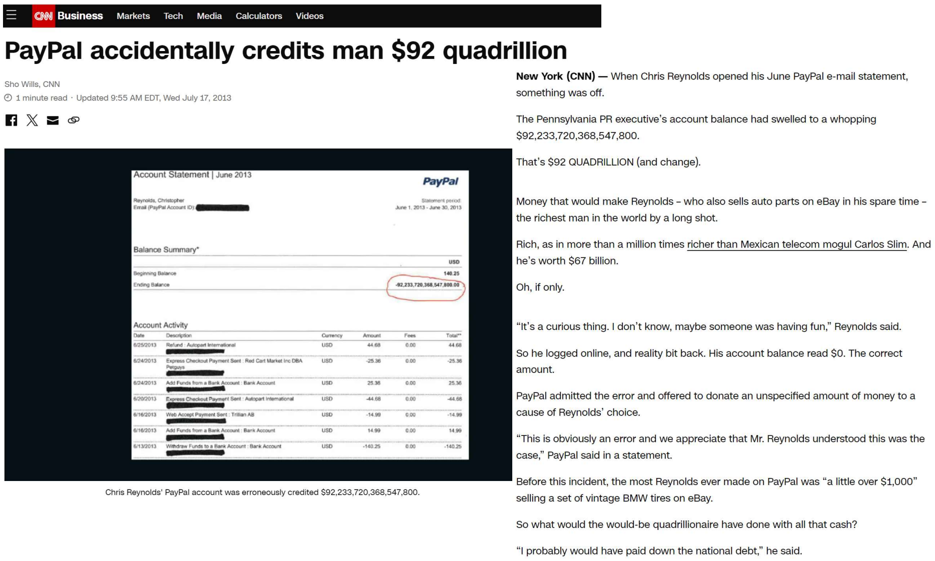
Medical Devices: The Therac-25*

Inadequate Software Engineering Practices. Some basic software engineering principles that apparently were violated in the case of the Therac-25 include the following:

- Software specifications and documentation should not be an afterthought.
- Rigorous software quality assurance practices and standards should be established.
- Designs should be kept simple and dangerous coding practices avoided.
- Ways to detect errors and and get information about them, such as software audit trails, should be designed into the software from the beginning.
- The software should be subjected to extensive testing and formal analysis at the module and software level; system testing alone is not adequate. Regression testing should be performed on all software changes.
- Computer displays and the presentation of information to the opera-
tors, such as error messag
mentation need to be care

The manufacturer said that the hardware and software were "tested and exercised separately or together over many years." In his deposition for one of the lawsuits, the quality assurance manager explained that testing was done in two parts. A "small amount" of software testing was done on a simulator, but most of the testing was done as a system. It appears that unit and software testing was minimal, with most of the effort directed at the integrated system test. At a Therac-25 user's meeting, the same man stated that the Therac-25 software was tested for 2,700 hours. Under questioning by the users, he clarified this as meaning "2700 hours of use." The FDA difficulty in getting an adequate test plan out of the company and the lack of regression testing are evidence that testing was not done well.

Historias de fallos



PayPal accidentally credits man $92 quadrillion

Sho Wills, CNN

1 minute read · Updated 9:55 AM EDT, Wed July 17, 2013

Account Statement | June 2013

PayPal

Reynolds, Christopher
Email (PayPal Account ID): [redacted]

Statement period:
June 1, 2013 - June 30, 2013

Balance Summary*

	USD
Beginning Balance	140.25
Ending Balance	-92,233,720,368,547,800.00

Account Activity

Date	Description	Currency	Amount	Fees	Total**
6/25/2013	Refund : Autopart International [redacted]	USD	44.68	0.00	44.68
6/24/2013	Express Checkout Payment Sent : Red Cart Market Inc DBA Petguys [redacted]	USD	-25.36	0.00	-25.36
6/24/2013	Add Funds from a Bank Account : Bank Account [redacted]	USD	25.36	0.00	25.36
6/20/2013	Express Checkout Payment Sent : Autopart International [redacted]	USD	-44.68	0.00	-44.68
6/16/2013	Web Accept Payment Sent : Trillian AB [redacted]	USD	-14.99	0.00	-14.99
6/16/2013	Add Funds from a Bank Account : Bank Account [redacted]	USD	14.99	0.00	14.99
6/13/2013	Withdraw Funds to a Bank Account : Bank Account [redacted]	USD	-140.25	0.00	-140.25

Chris Reynolds' PayPal account was erroneously credited $92,233,720,368,547,800.

New York (CNN) — When Chris Reynolds opened his June PayPal e-mail statement, something was off.

The Pennsylvania PR executive's account balance had swelled to a whopping $92,233,720,368,547,800.

That's $92 QUADRILLION (and change).

Money that would make Reynolds – who also sells auto parts on eBay in his spare time – the richest man in the world by a long shot.

Rich, as in more than a million times richer than Mexican telecom mogul Carlos Slim. And he's worth $67 billion.

Oh, if only.

"It's a curious thing. I don't know, maybe someone was having fun," Reynolds said.

So he logged online, and reality bit back. His account balance read $0. The correct amount.

PayPal admitted the error and offered to donate an unspecified amount of money to a cause of Reynolds' choice.

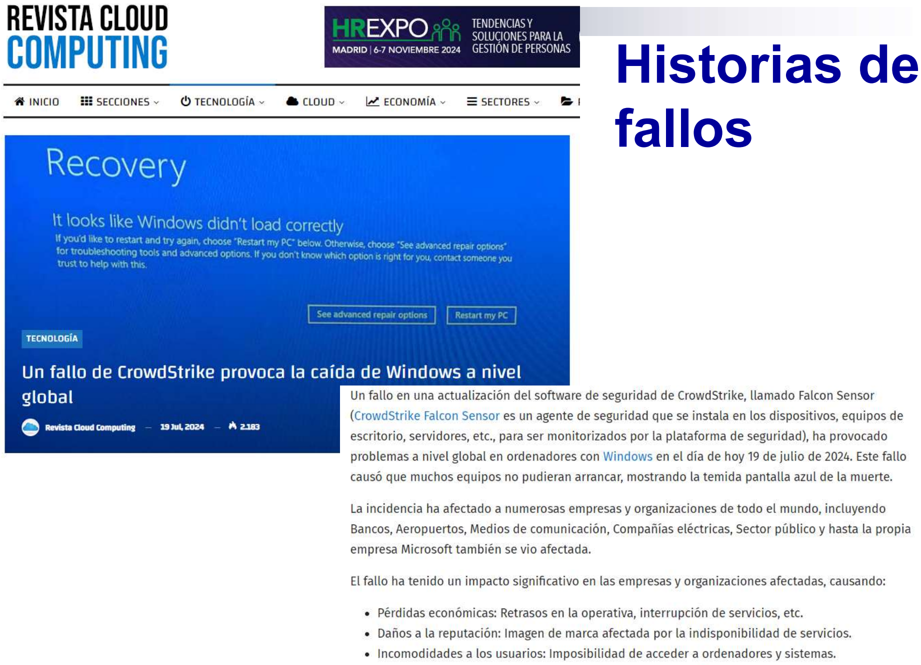
"This is obviously an error and we appreciate that Mr. Reynolds understood this was the case," PayPal said in a statement.

Before this incident, the most Reynolds ever made on PayPal was "a little over $1,000" selling a set of vintage BMW tires on eBay.

So what would the would-be quadrillionaire have done with all that cash?

"I probably would have paid down the national debt," he said.

Historias de fallos


REVISTA CLOUD COMPUTING
HREXPO TENDENCIAS Y SOLUCIONES PARA LA GESTIÓN DE PERSONAS
MADRID | 6-7 NOVIEMBRE 2024
INICIO
SECCIONES
TECNOLOGÍA
CLOUD
ECONOMÍA
SECTORES
Recovery
It looks like Windows didn't load correctly
If you'd like to restart and try again, choose "Restart my PC" below. Otherwise, choose "See advanced repair options" for troubleshooting tools and advanced options. If you don't know which option is right for you, contact someone you trust to help with this.
See advanced repair options
Restart my PC
TECNOLOGÍA

Un fallo de CrowdStrike provoca la caída de Windows a nivel global

Revista Cloud Computing — 19 Jul, 2024 — 2.183

Un fallo en una actualización del software de seguridad de CrowdStrike, llamado Falcon Sensor (CrowdStrike Falcon Sensor es un agente de seguridad que se instala en los dispositivos, equipos de escritorio, servidores, etc., para ser monitorizados por la plataforma de seguridad), ha provocado problemas a nivel global en ordenadores con Windows en el día de hoy 19 de julio de 2024. Este fallo causó que muchos equipos no pudieran arrancar, mostrando la temida pantalla azul de la muerte.

La incidencia ha afectado a numerosas empresas y organizaciones de todo el mundo, incluyendo Bancos, Aeropuertos, Medios de comunicación, Compañías eléctricas, Sector público y hasta la propia empresa Microsoft también se vio afectada.

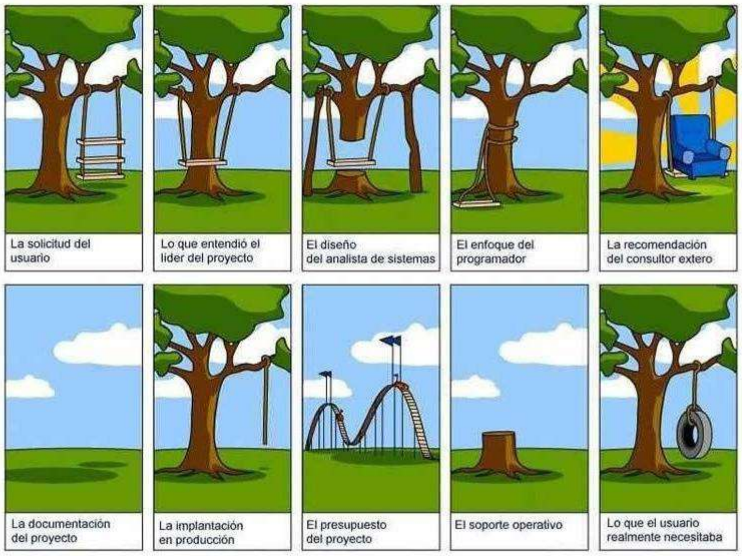
El fallo ha tenido un impacto significativo en las empresas y organizaciones afectadas, causando:

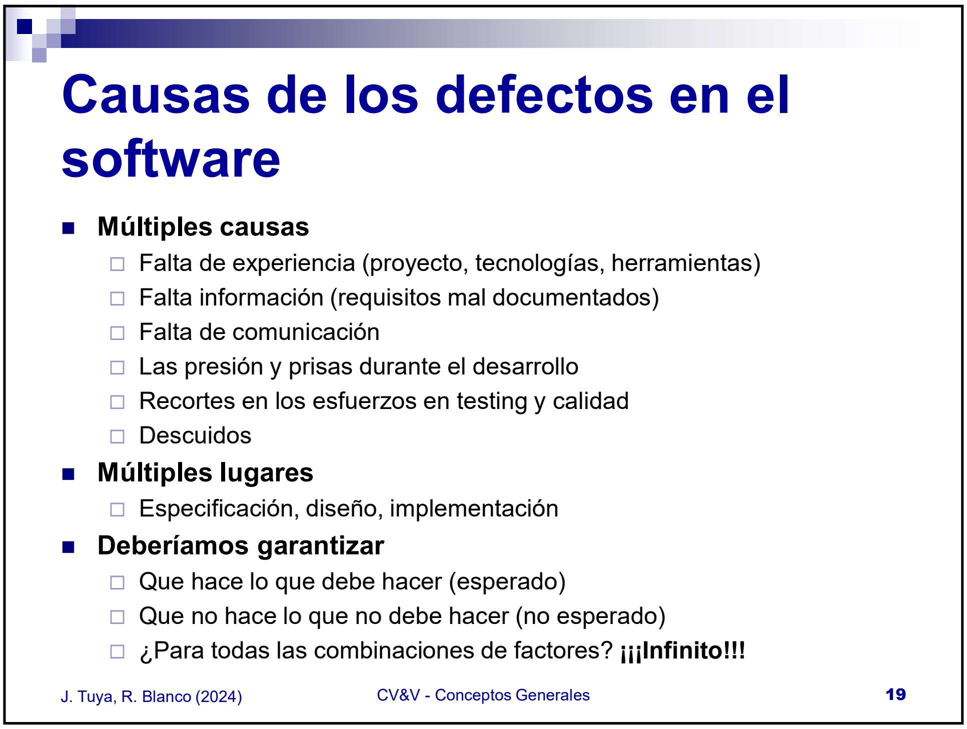
- Pérdidas económicas: Retrasos en la operativa, interrupción de servicios, etc.
- Daños a la reputación: Imagen de marca afectada por la indisponibilidad de servicios.
- Incomodidades a los usuarios: Imposibilidad de acceder a ordenadores y sistemas.

Seguridad

- SQL injection flaw in Wall Street Journal database led to breach
  - □ A vulnerability in a web-based graphics system led to a breach of The Wall Street Journal's network by a hacker, the newspaper acknowledged late Tuesday.
  - □ IT World (July 23, 2014)
- TalkTalk gets record £400,000 fine for failing to prevent October 2015 attack
  - □ The attacker accessed the personal data of 156,959 customers including their names, addresses, dates of birth, phone numbers and email addresses. In 15,656 cases, the attacker also had access to bank account details and sort codes
  - □ Ico.org.uk (Oct 5th, 2016)

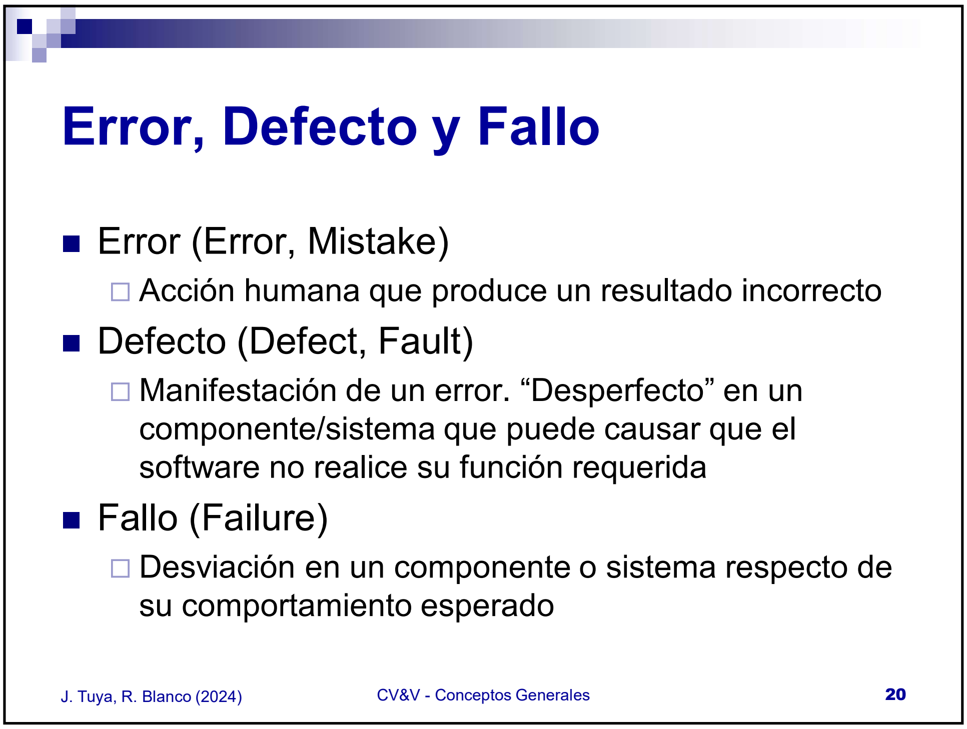
Problemática General


La solicitud del usuario
Lo que entendió el líder del proyecto
El diseño del analista de sistemas
El enfoque del programador
La recomendación del consultor extero
La documentación del proyecto
La implantación en producción
El presupuesto del proyecto
El soporte operativo
Lo que el usuario realmente necesitaba



Causas de los defectos en el software

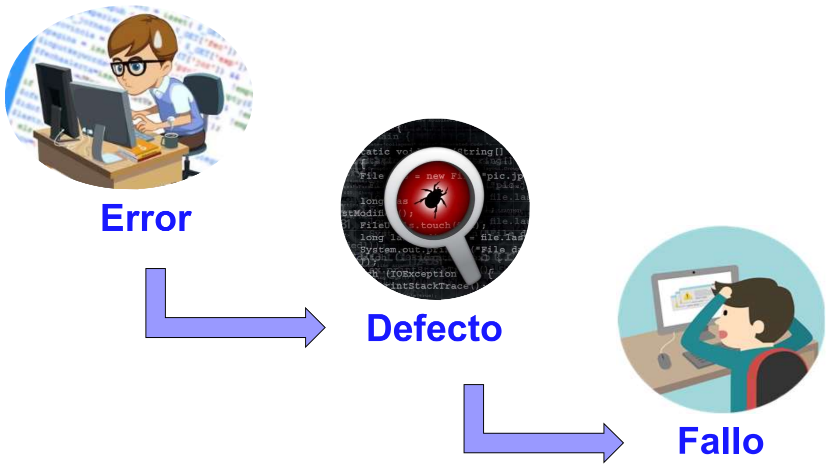
- **Múltiples causas**
  - □ Falta de experiencia (proyecto, tecnologías, herramientas)
  - □ Falta información (requisitos mal documentados)
  - □ Falta de comunicación
  - □ Las presión y prisas durante el desarrollo
  - □ Recortes en los esfuerzos en testing y calidad
  - □ Descuidos
- **Múltiples lugares**
  - □ Especificación, diseño, implementación
- **Deberíamos garantizar**
  - □ Que hace lo que debe hacer (esperado)
  - □ Que no hace lo que no debe hacer (no esperado)
  - □ ¿Para todas las combinaciones de factores? **¡¡¡Infinito!!!**



Error, Defecto y Fallo

- Error (Error, Mistake)
  - □ Acción humana que produce un resultado incorrecto
- Defecto (Defect, Fault)
  - □ Manifestación de un error. “Desperfecto” en un componente/sistema que puede causar que el software no realice su función requerida
- Fallo (Failure)
  - □ Desviación en un componente o sistema respecto de su comportamiento esperado

Error, Defecto y Fallo



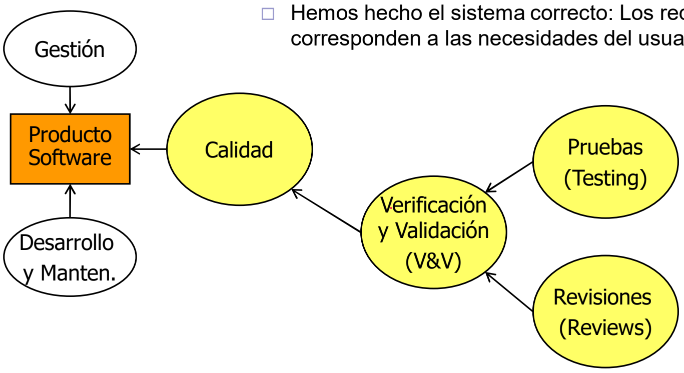
Error

Defecto

Fallo

CV&V - Contexto

- Conseguir calidad implica:
  - □ Hemos hecho las cosas correctamente: El producto satisface los requisitos: Funcionales y de rendimiento (explícitos), de Calidad (implícitos)
  - □ Hemos hecho el sistema correcto: Los requisitos corresponden a las necesidades del usuario



Gestión

Producto Software

Desarrollo y Manten.

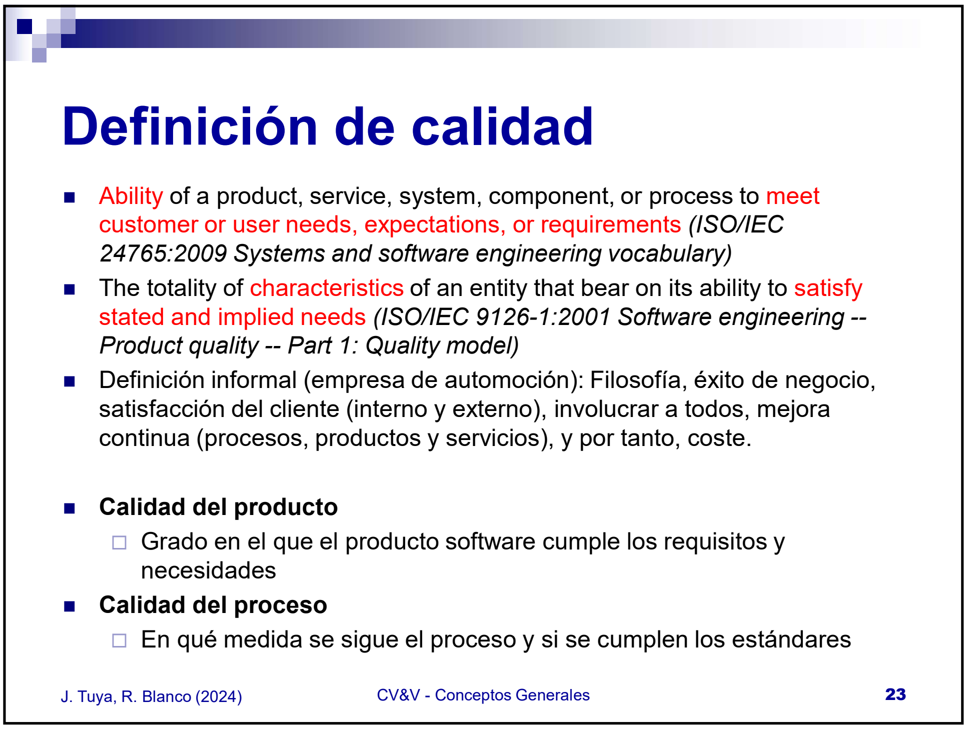
Calidad

Verificación y Validación (V&V)

Pruebas (Testing)

Revisiones (Reviews)

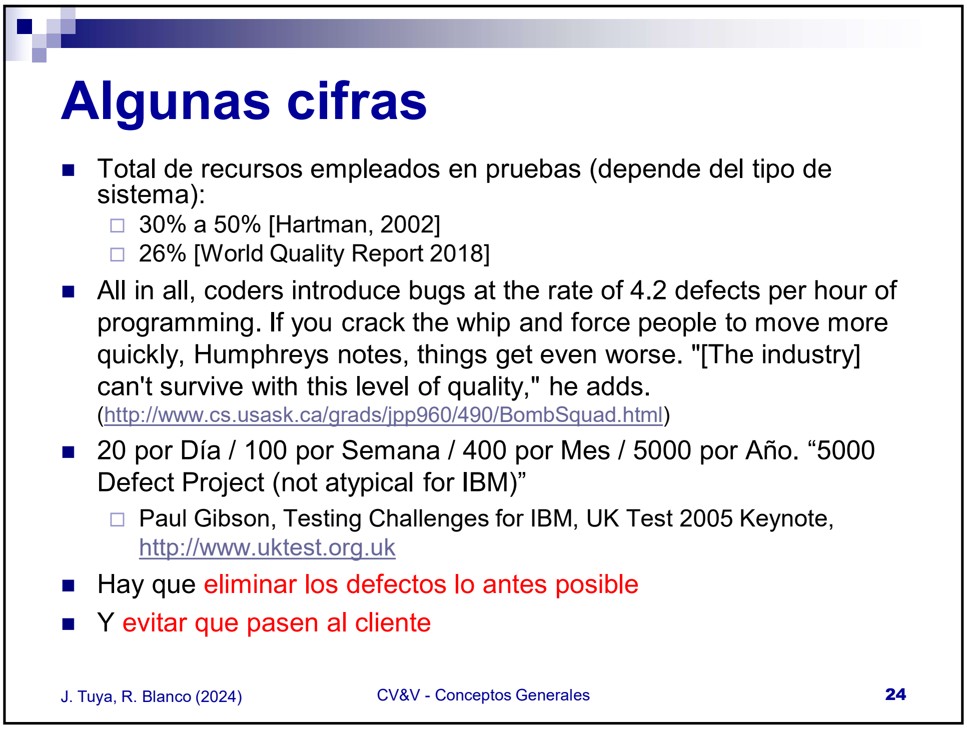
- Para quién?
  - □ Usuario
  - □ Cliente
  - □ Desarrollador
  - □ Tester



Definición de calidad

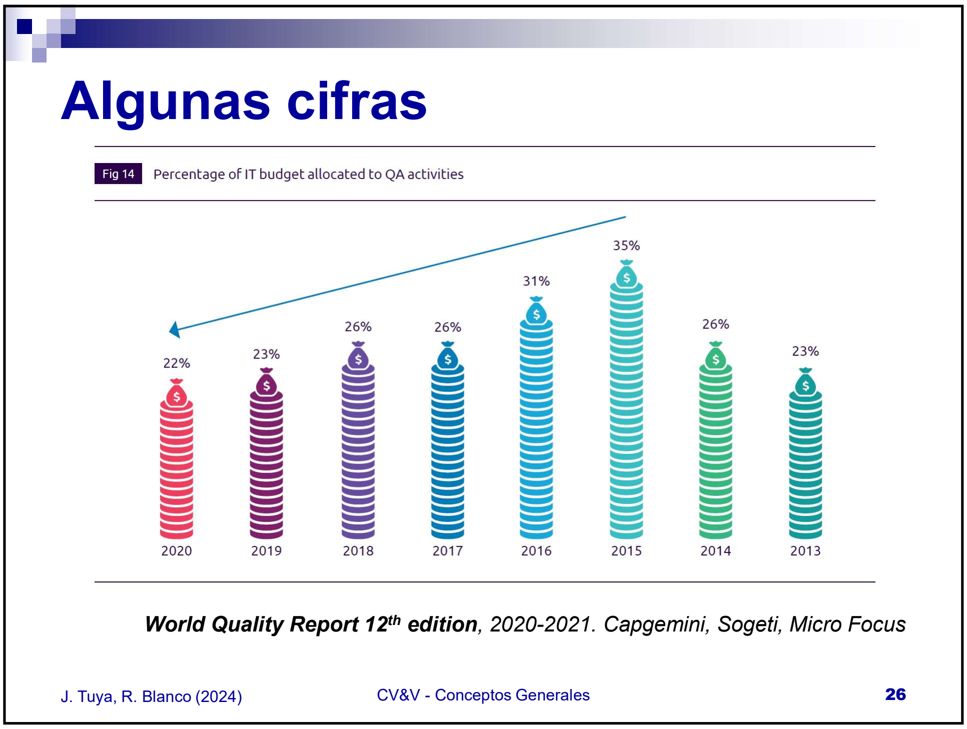
- Ability of a product, service, system, component, or process to meet customer or user needs, expectations, or requirements *(ISO/IEC 24765:2009 Systems and software engineering vocabulary)*
- The totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs *(ISO/IEC 9126-1:2001 Software engineering -- Product quality -- Part 1: Quality model)*
- Definición informal (empresa de automoción): Filosofía, éxito de negocio, satisfacción del cliente (interno y externo), involucrar a todos, mejora continua (procesos, productos y servicios), y por tanto, coste.

- **Calidad del producto**
  - □ Grado en el que el producto software cumple los requisitos y necesidades
- **Calidad del proceso**
  - □ En qué medida se sigue el proceso y si se cumplen los estándares



Algunas cifras

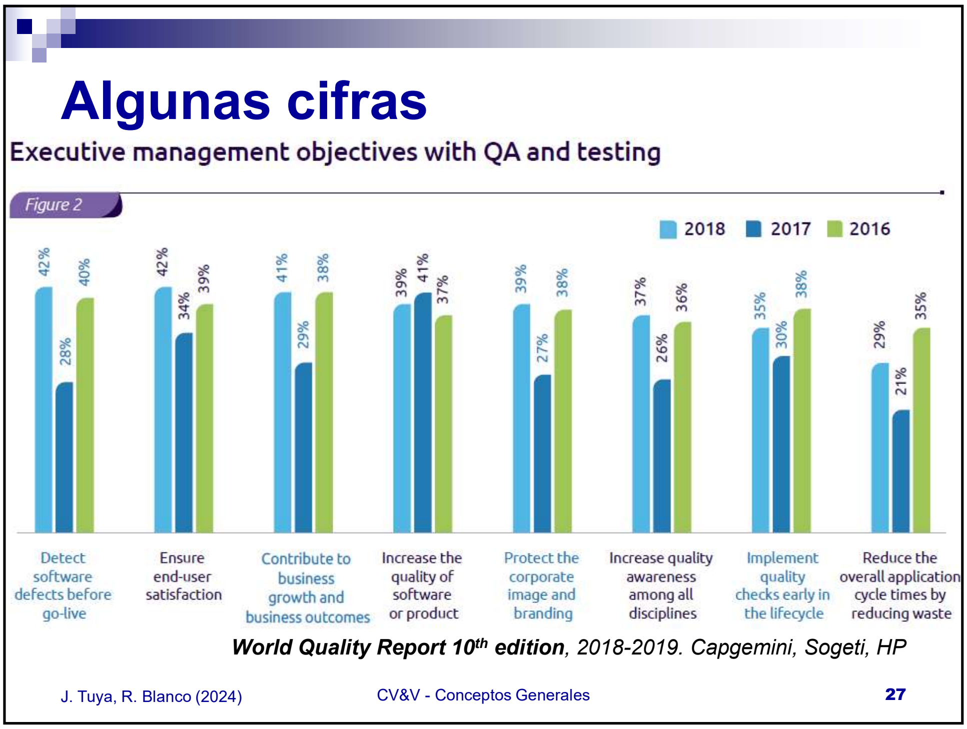
- Total de recursos empleados en pruebas (depende del tipo de sistema):
  - □ 30% a 50% [Hartman, 2002]
  - □ 26% [World Quality Report 2018]
- All in all, coders introduce bugs at the rate of 4.2 defects per hour of programming. If you crack the whip and force people to move more quickly, Humphreys notes, things get even worse. "[The industry] can't survive with this level of quality," he adds. (http://www.cs.usask.ca/grads/jpp960/490/BombSquad.html)
- 20 por Día / 100 por Semana / 400 por Mes / 5000 por Año. "5000 Defect Project (not atypical for IBM)"
  - □ Paul Gibson, Testing Challenges for IBM, UK Test 2005 Keynote, http://www.uktest.org.uk
- Hay que eliminar los defectos lo antes posible
- Y evitar que pasen al cliente



Algunas cifras

Fig 14 Percentage of IT budget allocated to QA activities

World Quality Report 12th edition, *2020-2021. Capgemini, Sogeti, Micro Focus*



Algunas cifras

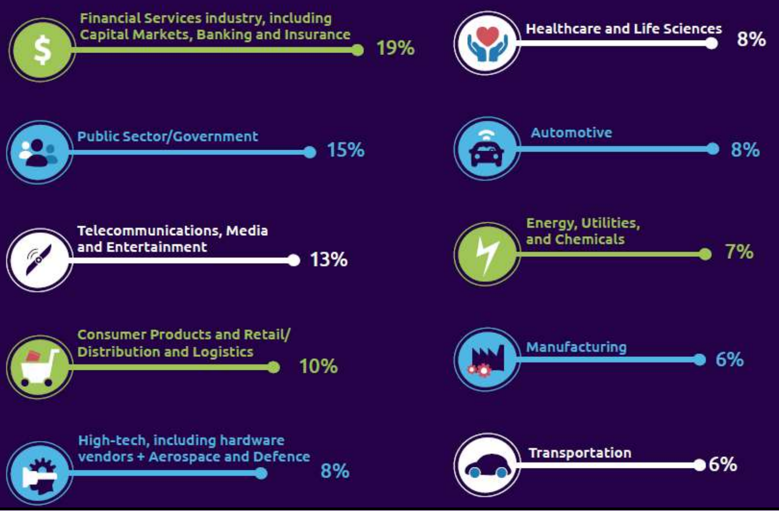
Executive management objectives with QA and testing

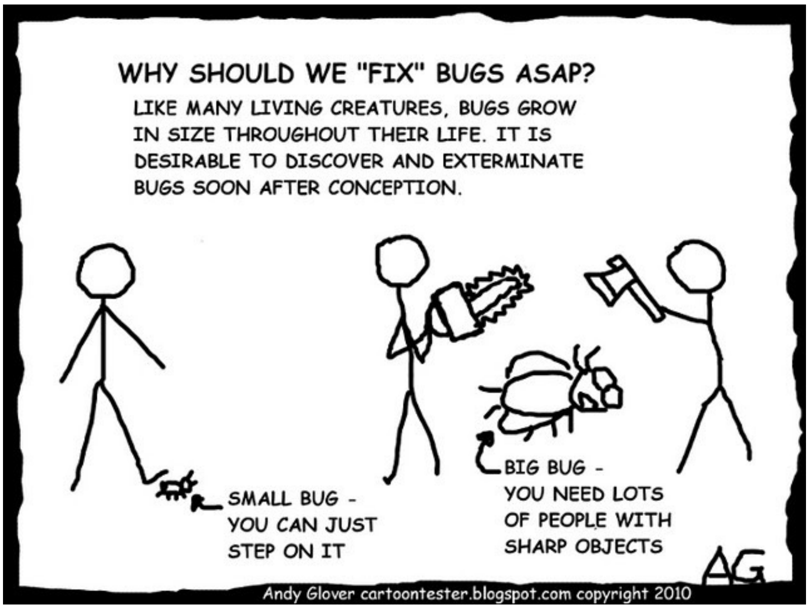
Figure 2

World Quality Report 10th edition, *2018-2019. Capgemini, Sogeti, HP*

Interviews by sectors

Figure 31




WHY SHOULD WE "FIX" BUGS ASAP?
LIKE MANY LIVING CREATURES, BUGS GROW IN SIZE THROUGHOUT THEIR LIFE. IT IS DESIRABLE TO DISCOVER AND EXTERMINATE BUGS SOON AFTER CONCEPTION.
SMALL BUG - YOU CAN JUST STEP ON IT
BIG BUG - YOU NEED LOTS OF PEOPLE WITH SHARP OBJECTS
AG
Andy Glover cartoontester.blogspot.com copyright 2010

THE CARTOON TESTER ON HIS WAY TO EXPOQA
A CONVERSATION WITH A FELLOW PASSANGER TURNS TO TESTING EVEN BEFORE THE PLANE HAS SET OFF!
ARE YOU ON A BUSINESS TRIP?
SORT OF. I'M GOING TO A S/W TESTING CONFERENCE. I'M REALLY LOOKING FORWARD TO IT.
AH! SO YOU CAN TELL ME HOW FULLY TESTED THIS PLANE WAS. YOU SEE, I'M A BIT SCARED OF FLYING.
NOTHING IS EVER FULLY TESTED.
OH, REALLY? WHAT DO YOU MEAN?
WELL, IT'S IMPOSSIBLE TO IMAGINE EVERY TEST SCENARIO, NEVER MIND TESTING THEM ALL.
BUT THAT CAN'T BE TRUE.
OK, DO YOU REALLY THINK THIS PLANE WAS TESTED WITH TODAY'S EXACT WEATHER CONDITIONS WITH THE WEIGHT OF PASSENGERS AND LUGGAGE? I DOUBT IT. AND THAT'S NOT ALL...
STOP THIS PLANE! I WANT TO GET OUT!!
DID I SAY SOMETHING WRONG?
AG
Andy Glover cartoontester.blogspot.com Copyright 2010

Verificación y Validación

- Verification: Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled *(ISO/IEC 12207:2008 Systems and software engineering--Software life cycle processes) (ISO/IEC 15288:2008 Systems and software engineering--System life cycle processes)*
- Validation: Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled *(ISO/IEC 15288:2008 Systems and software engineering--System life cycle processes, 4.37)*. In a life cycle context, the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives *(ISO/IEC 12207:2008 Systems and software engineering--Software life cycle processes, 4.54)*

Múltiples formas

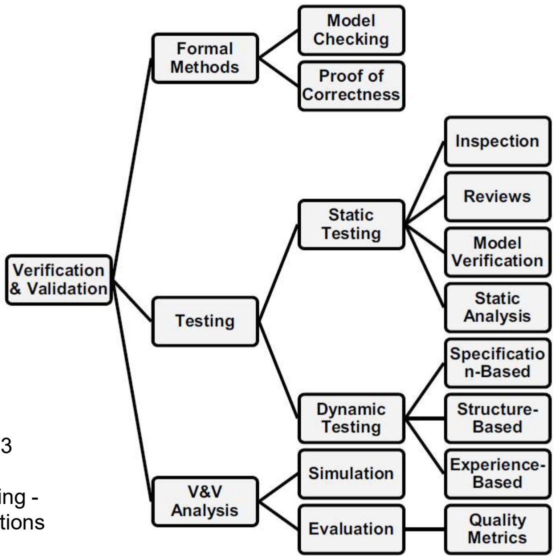


Figure A.1 — Hierarchy of Verification and Validation activities

ISO/IEC/IEEE 29119-1:2013
Software and systems engineering - Software testing - Part 1: Concepts and definitions

Revisiones

- Review: A process or meeting during which a work product, or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval *(ISO/IEC 24765:2009 Systems and software engineering vocabulary)*
- Peer Review: Review of work products performed by others qualified to do the same work (ISO/IEC 24765:2009 Systems and software engineering vocabulary) Note: often performed during development of the work products to identify defects for removal.

Pruebas – Definiciones literatura

- Definición clásica: La prueba (testing) es el proceso de ejecutar un programa con la intención de encontrar fallos [Glenford J. Myers]
  - Un buen caso de prueba es el que tiene alta probabilidad de detectar un nuevo fallo
  - Un caso de prueba con éxito es el que detecta un fallo nuevo
- Activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an **evaluation** is made of some aspect of the system or component *(IEEE 829-2008 IEEE Standard for Software and System Test Documentation)*
- Concepto extendido de Prueba: A technical investigation of the product under test conducted to provide stakeholders with quality-related information (Cem Kaner, BBST)

Pruebas – Definiciones estándar

- **Set of activities** conducted to facilitate discovery and/or evaluation of properties of one or more test items. Note: Testing activities could include planning, preparation, execution, reporting, and management activities, insofar as they are directed towards testing. *(ISO/IEC/IEEE 29119-1:2013 Systems and software engineering – Software Testing - Part 1: Concepts and definitions)*
  - Dynamic testing: testing that requires the execution of the test item
  - Static testing: testing in which a test item is examined against a set of quality or other criteria without code being executed
- The **process** consisting of **all lifecycle activities**, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects *(ISTQB Standard glossary of terms used in Software Testing, Version 2.1)*

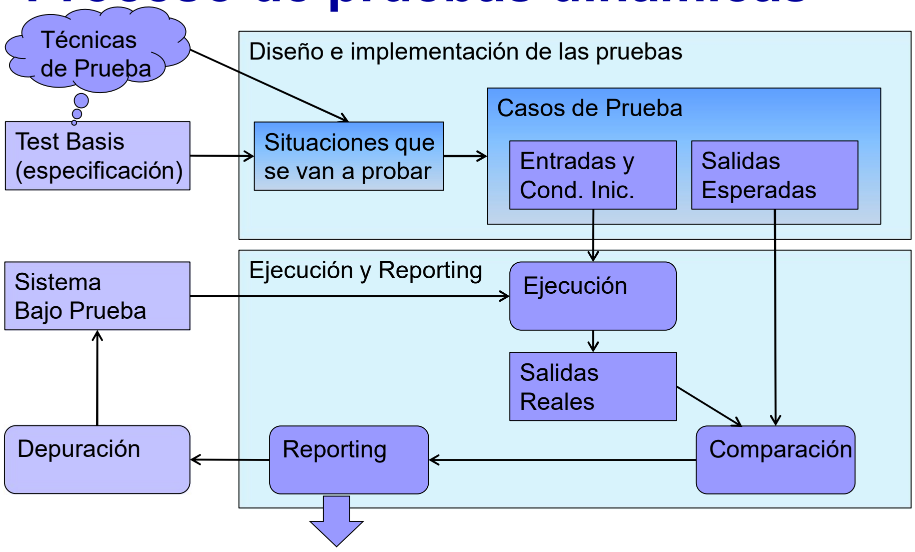
Pruebas - Discusión

- ¿Las pruebas son verificación o validación?
- ¿El objetivo es detectar errores?
- ¿Es el único objetivo?
- ¿Si a corto plazo voy a ser desarrollador, me interesa esto?

Pruebas - Clasificación

- Ejecución
  - □ **Dinámicas**: se ejecuta
  - □ **Estáticas**: no se ejecuta
- Visibilidad de la estructura interna
  - □ **Caja negra**: no es visible
  - □ **Caja blanca**: es visible

Proceso de pruebas dinámicas



Pruebas Continuas

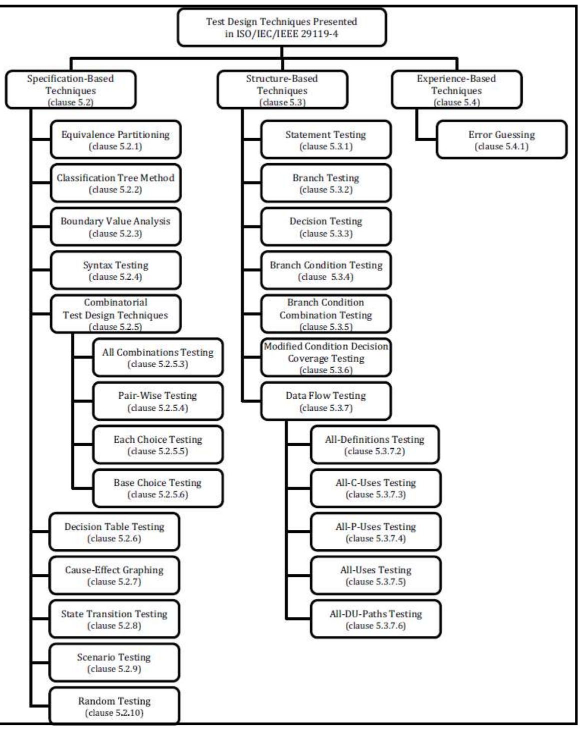
- Evolución hacia Agile y DevOps:
  - □ Desarrollo Continuo
  - □ Pruebas Continuas
- Pero
  - □ No basta con que las pruebas se ejecuten de forma automática
  - □ Deben ser **BUENAS** pruebas (Importancia del uso de técnicas)

Conocimientos básicos

- Definir (Técnicas)
  - □ Pruebas funcionales
  - □ Pruebas no funcionales, revisiones, análisis estático
- Automatizar (Herramientas)
- En el contexto de un proyecto
  - □ Niveles de Prueba, Ejecución, reporting, Gestión…
  - □ Consideraciones en proyectos ágiles
- Aspectos Metodológicos y Estándares
  - □ Metodologías (METESPA, TMAP Next^{TM})
  - □ Estándares (ISO/IEC/IEEE 29119)

Técnicas Estándar

ISO/IEC/IEEE 29119-4:2015 Software Testing - Part 4: Test techniques



- Test Design Techniques Presented in ISO/IEC/IEEE 29119-4
  - Specification-Based Techniques (clause 5.2)
    - Equivalence Partitioning (clause 5.2.1)
    - Classification Tree Method (clause 5.2.2)
    - Boundary Value Analysis (clause 5.2.3)
    - Syntax Testing (clause 5.2.4)
    - Combinatorial Test Design Techniques (clause 5.2.5)
      - All Combinations Testing (clause 5.2.5.3)
      - Pair-Wise Testing (clause 5.2.5.4)
      - Each Choice Testing (clause 5.2.5.5)
      - Base Choice Testing (clause 5.2.5.6)
    - Decision Table Testing (clause 5.2.6)
    - Cause-Effect Graphing (clause 5.2.7)
    - State Transition Testing (clause 5.2.8)
    - Scenario Testing (clause 5.2.9)
    - Random Testing (clause 5.2.10)
  - Structure-Based Techniques (clause 5.3)
    - Statement Testing (clause 5.3.1)
    - Branch Testing (clause 5.3.2)
    - Decision Testing (clause 5.3.3)
    - Branch Condition Testing (clause 5.3.4)
    - Branch Condition Combination Testing (clause 5.3.5)
    - Modified Condition Decision Coverage Testing (clause 5.3.6)
    - Data Flow Testing (clause 5.3.7)
      - All-Definitions Testing (clause 5.3.7.2)
      - All-C-Uses Testing (clause 5.3.7.3)
      - All-P-Uses Testing (clause 5.3.7.4)
      - All-Uses Testing (clause 5.3.7.5)
      - All-DU-Paths Testing (clause 5.3.7.6)
  - Experience-Based Techniques (clause 5.4)
    - Error Guessing (clause 5.4.1)

Técnicas

- Las técnicas no son recetas
  - □ No podemos hacer pruebas basándonos solamente en rellenar una plantilla
  - □ Pero sí son una herramienta que ayuda a realizar pruebas más efectivas, y de forma eficiente
- Se requiere aplicar conocimiento
  - □ Técnico
  - □ Funcional (dependencia del contexto)
  - □ Además de la experiencia y la creatividad

Bibliografía

- Básica:
  - □ Glenford J. Myers. The Art of Software Testing. Wiley (1979, 2004, 2011)
  - □ Cem Kaner, Jack Falk, Hung Quoc Nguyen. Testing Computer Software. Wiley (1999)
- Complementaria:
  - □ Dorothy Graham, Erik Van Veenendaal, Isabel Evans, Rex Black. Foundations on Software Testing. Cengage Learning (2008)
  - □ TMap® Next for result-driven testing, Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon, Uitgeverij Tutein Nolthenius, Den Bosch. UTN Publishers (2006)
  - □ ISO/IEC/IEEE 29119 Software and Systems Engineering – Software Testing