

Software Systems Verification and Validation

Lecture 01 - Introduction

Lect. dr. Andreea Vescan

Babeş-Bolyai University
Cluj-Napoca

2015-2016

1 Famous Software bugs

- First bug
- Famous Software bugs

2 What is a bug?

- Terms for software failures
- When?Why?Cost?

3 Quality

- Software quality
- Quality control vs. Quality Assurance
- Verification and Validation

4 Software development life cycle Model

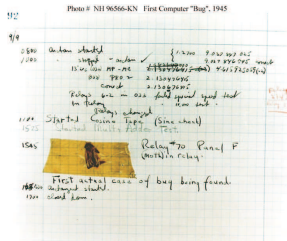
- V-Model
- Extended V-Model [?]

5 Next lecture

- Next lecture

First bug

- Grace Hopper - About first software bug:
- <https://www.youtube.com/watch?v=1QS0hDqpVLE>
- 1947 - Harvard University - Mark II



Mariner

- 1962



- Disaster: The Mariner 1 rocket diverted from its intended flight path shortly after launch.
- Cause: A programmer incorrectly transcribed a handwritten formula into computer code.
- Cost: \$ 18.5 million

World War III - almost

- 1983



- Disaster: The Soviet early warning system falsely indicated the United States had launched five ballistic missiles.
- Cause: A bug in the Soviet software failed to filter out false missile detections caused by sunlight reflecting off cloud-tops.
- Cost: Nearly all of humanity

Therac-25 radiation therapy machine

- 1985



- Disaster: Canadas Therac-25 radiation therapy machine malfunctioned and delivered lethal radiation doses to patients.
- Cause: Because of a subtle bug called a race condition, a technician could accidentally configure Therac-25 so the electron beam would fire in high-power mode without the proper patient shielding.
- Cost: Three people dead, three people critically injured

Pentium Fails Long Division

- 1993



- Disaster: Intels highly-promoted Pentium chip occasionally made mistakes when dividing floating-point numbers within a specific range.
- Cause: The divider in the Pentium floating point unit had a flawed division table, missing about five of a thousand entries and resulting in these rounding errors.
- Cost: \$475 million, corporate credibility.

Disney's Lion King

- 1995



- Disaster: The Disney company released its first multimedia cd-rom game for children, *The Lion King Animated Storybook*. Several parents couldn't get the software to work.
- Cause: Disney failed to test the software on a broad representation of the many different PC models available on the market.
- Cost: cd-rom replacements, corporate credibility.

Ariane Rocket

- 1996



- Disaster: Ariane 5, was intentionally destroyed seconds after launch on its maiden flight. Also its cargo of four scientific satellites to study how the Earth's magnetic field interacts with solar winds.
- Cause: Shutdown occurred when the guidance computer tried to convert the sideways rocket velocity from 64-bits to a 16-bit format.
- Cost: \$500 million.

Mars Climate Crasher

- 1998



- Disaster: After a 286-day journey from Earth, the Mars Climate Orbiter fired its engines to push into orbit around Mars. The engines fired, but the spacecraft fell too far into the planets atmosphere, likely causing it to crash on Mars.
- Cause: The software that controlled the Orbiter thrusters used imperial units (pounds of force), rather than metric units (Newtons) as specified by NASA.
- Cost: \$125 million.

Cancer Treatment

- 2000



- Disaster: Radiation therapy software by Multidata Systems International miscalculated the proper dosage, exposing patients to harmful and in some cases fatal levels of radiation.
- Cause: The software calculated radiation dosage based on the order in which data was entered, sometimes delivering a double dose of radiation.
- Cost: Eight people dead, 20 critically injured.

Terms for software failures [Pat05]

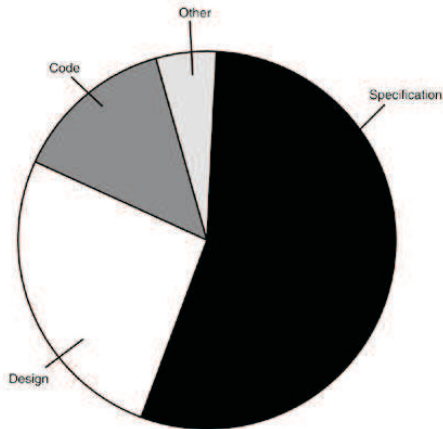
- **Failure:** A failure is said to occur whenever the external behavior of a system does not conform to that prescribed in the system specification.
- **Error:** An error is a state of the system. In the absence of any corrective action by the system, an error state could lead to a failure which would not be attributed to any event subsequent to the error.
- **Fault:** A fault is the adjudged cause of an error.
- Process of failure manifestation - represented as a behavior chain:
 - fault \rightarrow error \rightarrow failure.
- In this course, all software problems will be called bugs.

When a software bug occurs?

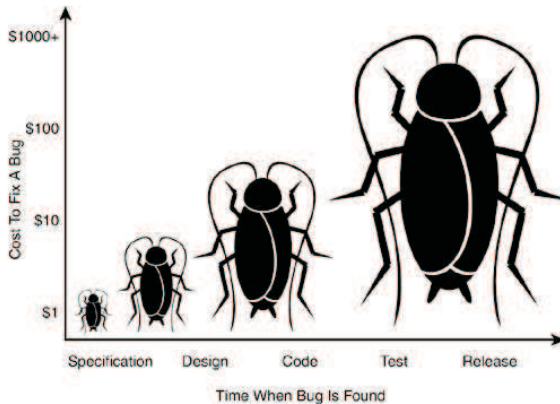
A software bug occurs when one or more of the following rules is true [Pat05]:

- The software doesn't do something that the product specification says it should do.
- The software does something that the product specification says it shouldn't do.
- The software does something that the product specification doesn't mention.
- The software doesn't do something that the product specification doesn't mention but should.
- The software is difficult to understand, hard to use, slow, or in the software tester's eyes will be viewed by the end user as just plain not right.

Why do bugs occur? [Pat05]



The cost of bugs [Pat05]



Software quality assessment

- "Software quality: Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software." [Pre00]
- Activities for software quality assessment
 - Static analysis
 - Based on the examination of a number of documents. It does not involve actual execution of the code under development.
 - Includes: code review, inspection, algorithm analysis, proof of correctness.
 - Dynamic analysis.
 - Involved actual program execution in order to expose possible program failures.
 - Includes: testing.

Quality control vs. Quality Assurance

- Quality control
 - QC = Quality of products
 - How do you control the quality of the work you have done?
 - Goal: detect problems in the work products
- Quality Assurance
 - QA = Quality of processes
 - How do you assure the quality of the work you are going to do?
 - Goal: Ensure adherence to processes, standards and plans

Verification and Validation (SEI and NASA) [NT05], [PY08]

Software Engineering Institute

• Verification

- assures the product is developed according to requirements, specifications and standards.
- building the product correctly.
- Are we building the product right?

• Validation

- assures that the product will be usable on the market.
- building the correct product.
- Are we building the right product?

NASA - Software Assurance Guidebook and Standard [NAS]

• Verification and Validation

- the process that assures that the software product:
 - will satisfy the requirement (functional and others) = validation.
 - every step in the product development is resulting in a correct (sub)product=verification.

Verification and Validation - comparison

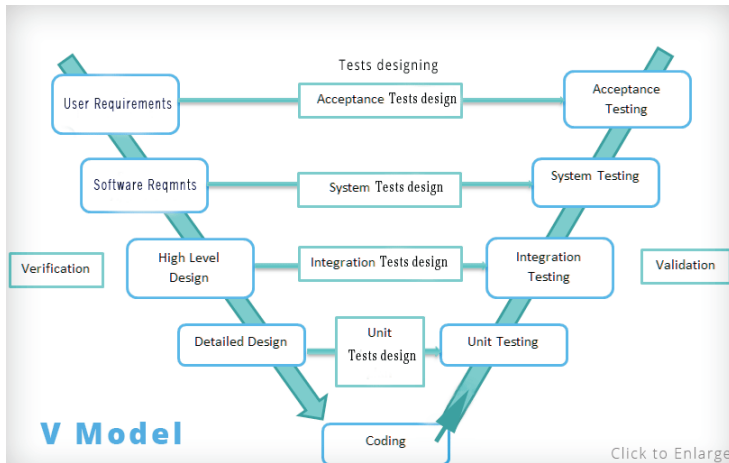
Verification

- evaluates if the product of a given development phase satisfies the requirements of that phase;
- reviews products to ensure their quality (consistency, completeness, correctness);
- static and dynamic analysis techniques.

Validation

- helps us at confirming that a product meets its intended use.
- is performed toward the end of the system development to determine if the entire system meets the customers's needs and expectations;
- is performed on the entire system by actually running the system in its real environment and using a variety of tests.

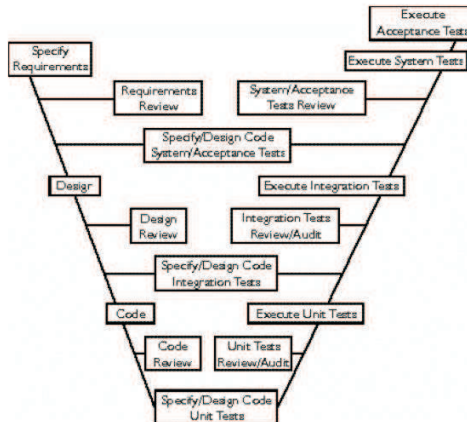
V-Model



Click to Enlarge



Extended V-Model



Next Lecture (Still today!)

- Inspection

Questions

- Thank You For Your Attention!

References



Jean-Francois Collard and Ilene Burnstein.

Practical Software Testing.

Springer-Verlag New York, Inc., 2003.



Nasa - standard for software assurance.

<http://www.hq.nasa.gov/office/codeq/doctree/87398.htm>.



K. Naik and P. Tripathy.

Software Testing and Quality Assurance.

Wiley Publishing, 2005.



R. Patton.

Software Testing.

Sams Publishing, 2005.