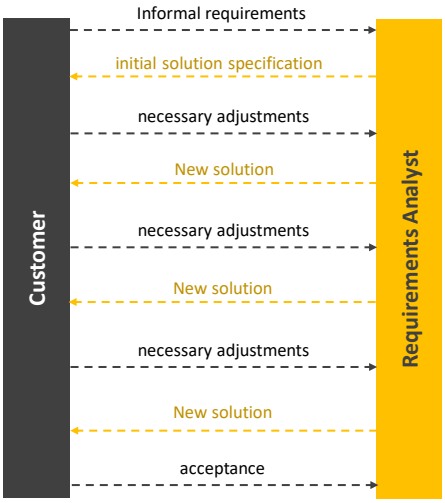
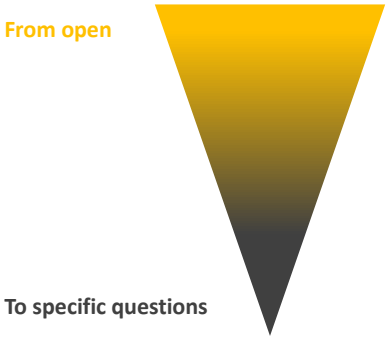
Last time

- Interviews are the most commonly used technique
- Must be carefully planned and prepared
  - Know who will be interviewed and why
  - Define the objective(s) of the interview
  - Prepare in advance the questions that will be asked (or part of them)
- **Some rework is expected at this stage!**



Last time



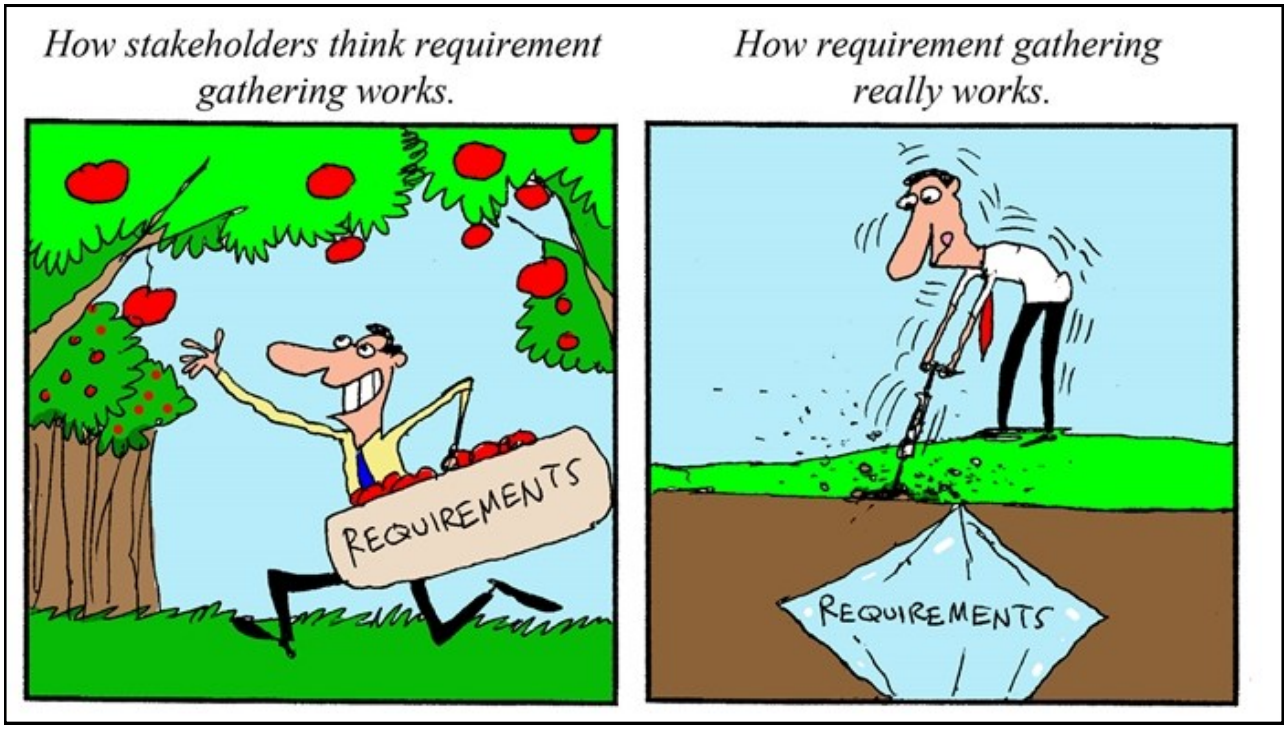
Get information about the **context**

Get information about the **process**

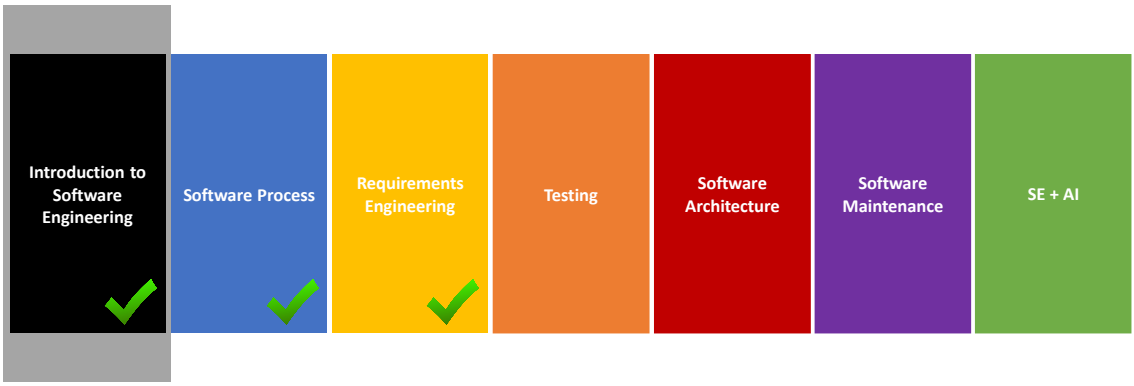
Focus on a specific **activity** of the process

Get information about the **steps** performed for that activity

Get information about the **data** and **business rules** of each step



We are making progress…



TDresearchteam
Technical Debt Research Team

VCU Computer Science
College of Engineering

Introduction to Software Testing

Dr. Rodrigo Spínola

Agenda

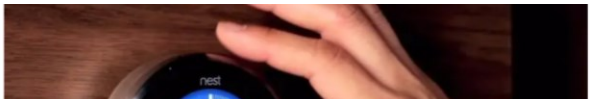
- Software fails
- Software quality
  - Internal vs external quality factors
  - Dynamic vs static analysis
  - Software validation vs software verification
- Software testing
  - Types: system, integration, functional, unit

> Lecture 14 – Introduction to Testing - Part 1

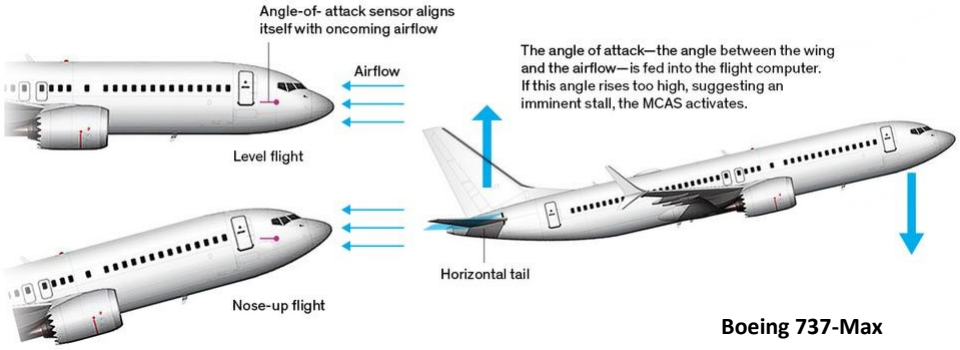
TDresearchteam
Technical Debt Research Team

VCU Computer Science
College of Engineering

Software failure stories



How the new Max flight-control system (MCAS) operates to prevent a stall



Boeing 737-Max

TDresearchteam
Technical Debt Research Team

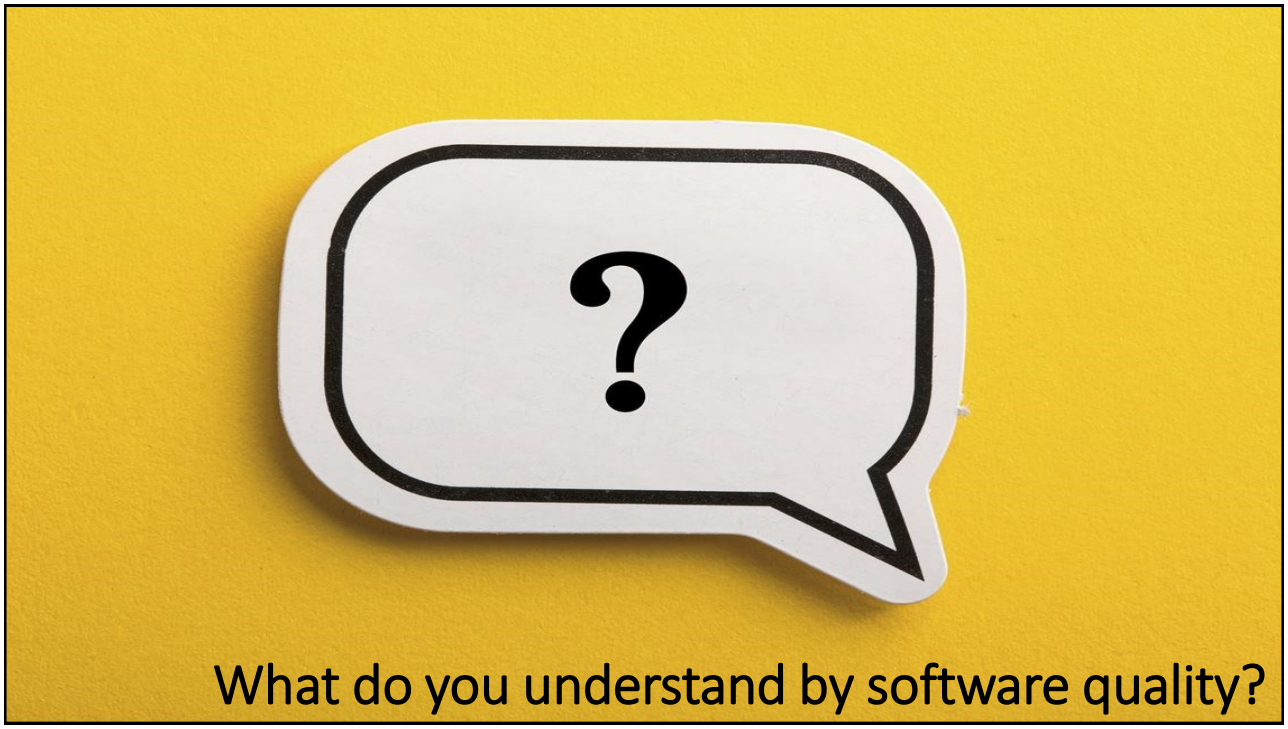
VCU Computer Science
College of Engineering

Software horror stories

- https://www.cs.tau.ac.il/~nachumd/horror.html

"Software is eating the world."

We need to assure its quality.



Software Quality

what software engineers **should** want

what the user wants

- Software meets requirements specification and well-known software development standards
- Software Quality Assurance
  - Set of activities applied during the **whole** software development process to assure that both the development process and the software product reach expected quality levels

It is not only about testing!!!

How???

Software Quality Factors

- **External factors** are perceived by both developers and users
  - Examples: usability, performance, reliability

What does it mean?

- **Internal factors** are perceived only by software development teams
  - Examples: modularity, maintainability, legibility
  - If the internal factors are attended, the external factors tend to be observed too.

Validation & Verification

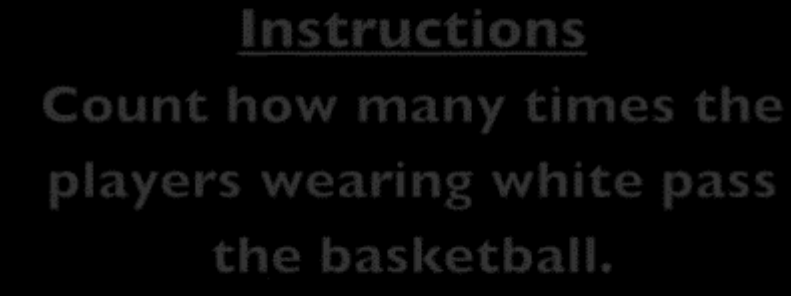
- **Validation:** evaluates a system to determine if it meets its requirements. **Did you build the right thing?**
  - Is this what the customer wants?
  - Is the specification correct?
- **Verification:** evaluates a system to determine if the product of a specific software development activity meets its expected results. **Did you build the thing right?**

Static vs Dynamic Analysis

- **Static analysis:** does not involve software execution. It can be applied to any software development artifact
  - E.g., requirements review, code inspection
- **Dynamic analysis:** requires software execution
  - E.g., testing

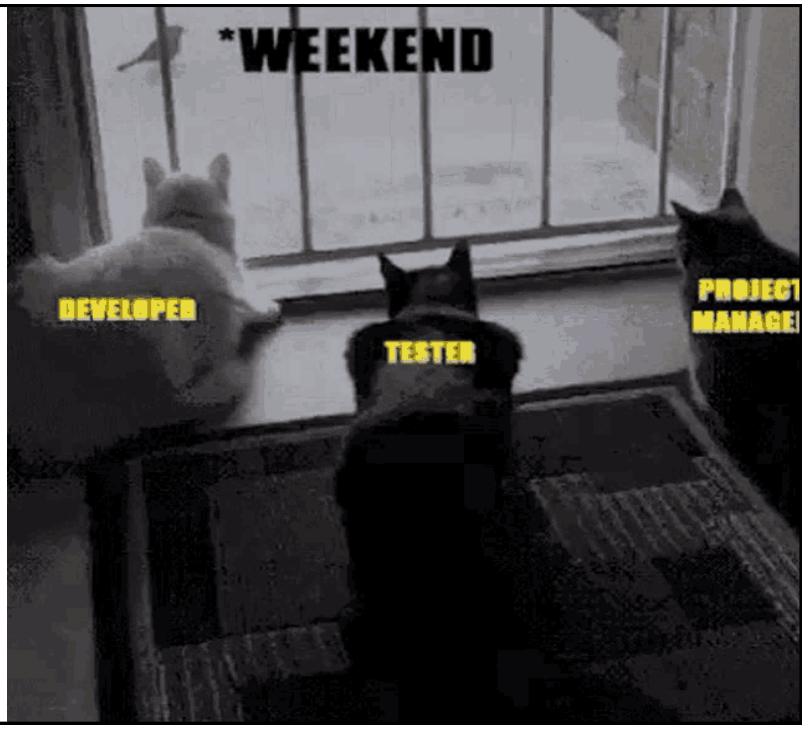
Would you be a good tester?

- Test planning
- Test coverage
- Test case
- Test procedure



Due to confirmation bias, it is common to see developers' tendency to confirm behavior rather than to break it.

Software Testing



Software Testing

Software **testing** is the process of **evaluating** and verifying that a software product **does what it is supposed to do**.

- During testing activities, we must use **representative data** WHY?
- To be effective, the test must be **carefully designed** WHY?
- Irreproducible or improvised tests **should be avoided** WHY?

TDresearchteam
Technical Debt Research Team

VCU Computer Science
College of Engineering

19

Developers are not the best person to test their own code

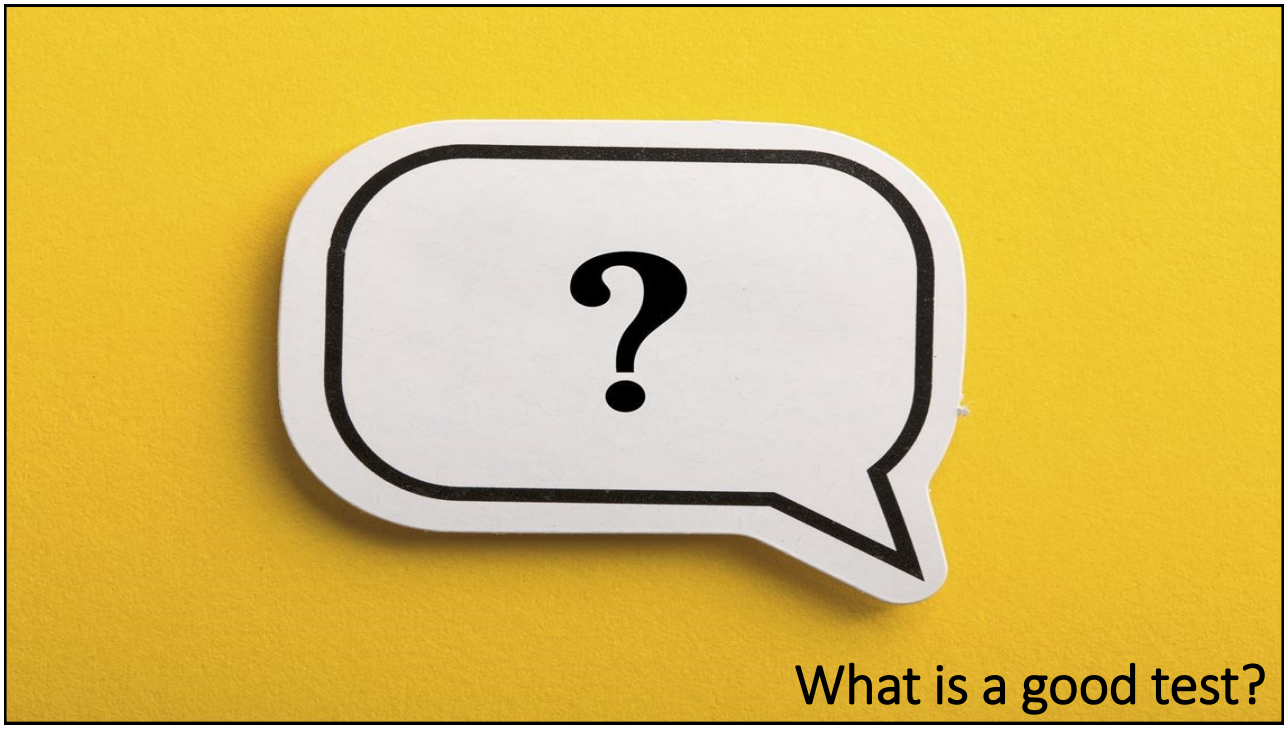
- They don't have time
  - Testing requires hard work, laser-sharp focus, and most of all, dedicated time!
- Unconscious bias
  - The person who develops the code generally sees only ***happy paths*** of the product and doesn't want to go into much detail…
- Slows down release times
  - Developer and QA tester are two separate jobs. When they are carried out by the same person, they cannot be done simultaneously. Coding will have to stop to allow for testing to begin and vice versa.
- QA testers are specially trained
  - Independent testers matter

20

Software Testing

- **Goals:**
  - maximize test coverage
  - detect as many errors as possible within given cost and time limits
- Dijkstra's words: "Testing can never show the **ABSENCE** of errors in software, only their **PRESENCE**"

Testing's goal runs counter to the goals of software development activities:
- The goal is to break it
- Hard for a developer to get in the proper mindset



What is a good test?

- High probability of revealing errors
- It is not redundant
- Appropriate coverage level
- Appropriate complexity level

Software Testing

- If you do **NOT** reveal errors during the software test execution, does it mean that:
  - you have a high-quality software

or

  - you have low-quality tests?

Testing Vs Debugging

- **Testing:** process of running a software with the aim of revealing the presence of faults, or, failing that objective, increasing confidence in the software
- **Debugging**: it is a consequence of the test. After the presence of the error is revealed, the defect must be found and corrected.

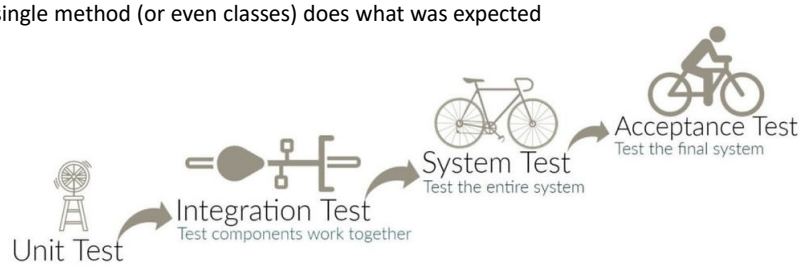
Debugging is not testing!

Testing

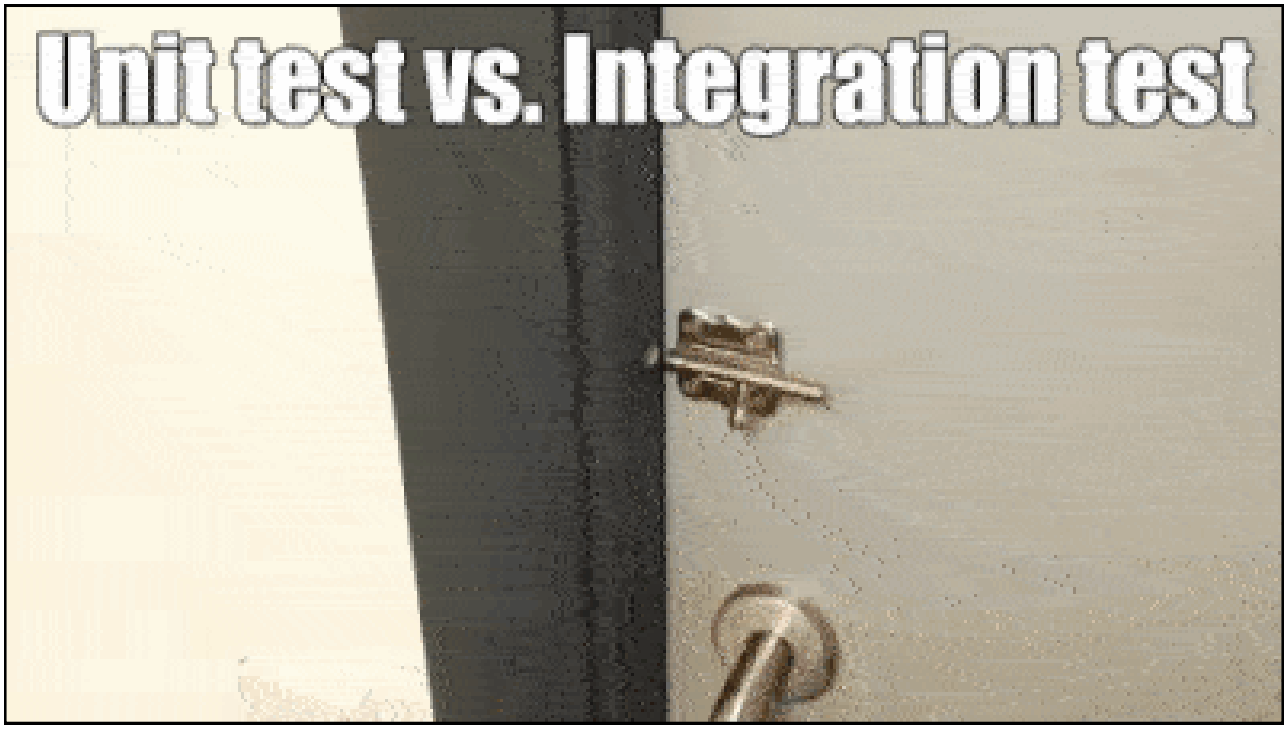
- Exhaustive testing infeasible
  - Remember Dijkstra's words: "Testing can never show the **ABSENCE** of errors in software, only their **PRESENCE**"
- Divide and conquer: perform different tests at different levels of the software
  - Upper level doesn't redo tests of lower level

Types of Tests

- **Acceptance test:** integrated system meets its specification
- **Functional test:** is a type of testing that seeks to establish whether each application feature works as per the software requirements
- **Integration test:** interfaces between units have consistent assumptions, communicate correctly
- **Unit test:** single method (or even classes) does what was expected




?
Why do we need different levels of software testing?


Unit test vs. Integration test

30

Unit tests vs Integration Tests

Why do we need different levels of software testing?

Summary

- Software quality
  - Internal vs external quality factors
  - Dynamic vs static analysis
  - Software validation vs software verification
- Software testing
  - Types: system, integration, functional, unit


Class is over, questions?