CIS 3374 - 5274
Quality Assurance & Testing

LECTURE CLASS 1

Agenda

- Definition of Quality and Quality Assurance
- Challenges of Software Quality
- Verification and Validation
- Definition and Objectives of Testing
- V-Model
- Testing Methods and Types

Software Is Everywhere!

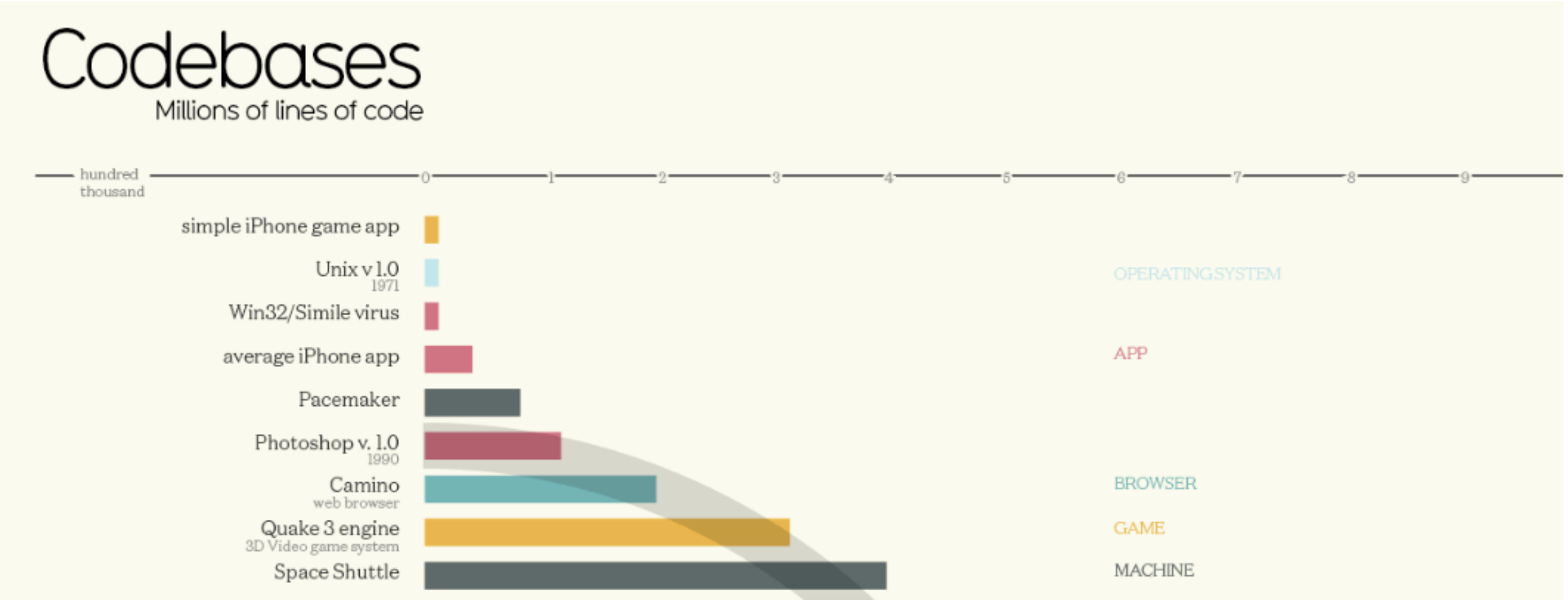
1977 General Motors Oldsmobile Toronado

The first production car to incorporate embedded software

2015 Tesla Model E

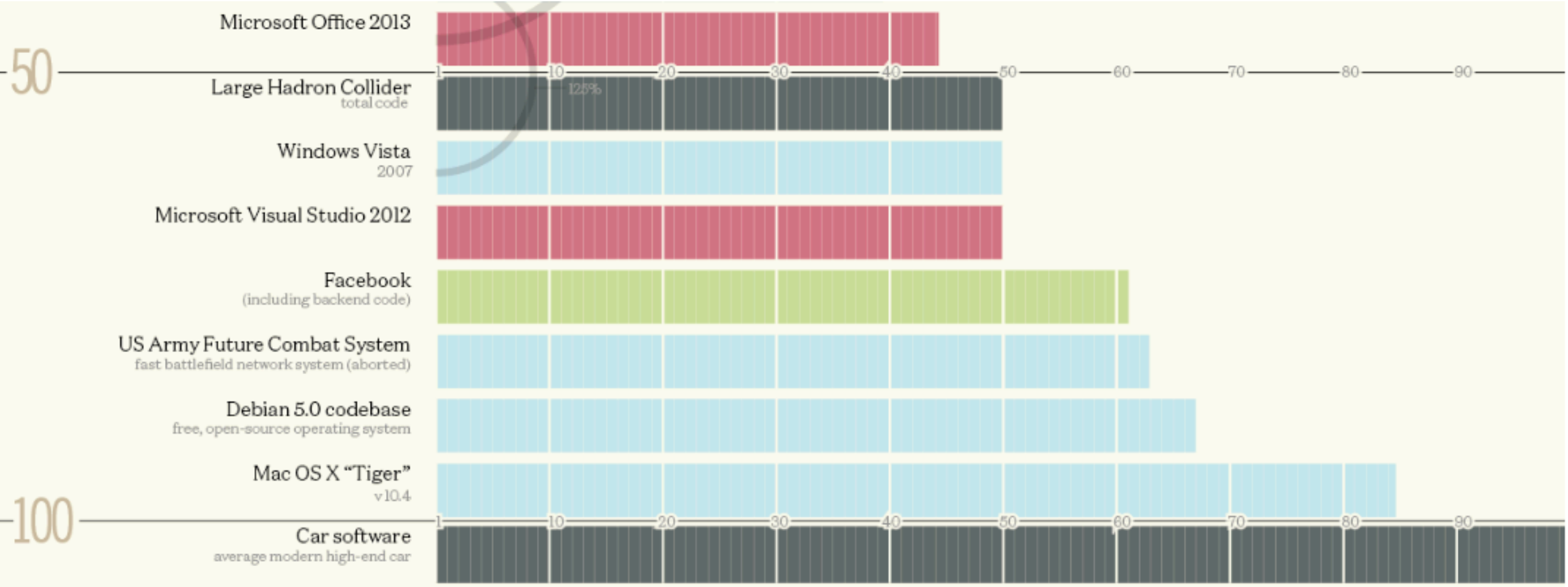
Tesla's Roadster has ~30M lines of code

Software Is Everywhere!



Source: Visual Capitalist, How Many Millions of Lines of Code Does It Take?

Software Is Everywhere!



Source: Visual Capitalist, [How Many Millions of Lines of Code Does It Take?]

Software Bugs Are Everywhere!

Video: Software Disasters

Course Description

Learn the techniques for creating quality systems.

This course discusses the crucial steps to assure that systems:

- do what they are intended to do;
- work reliably;
- satisfy the client's requirements;
- are completed on time and within budget.

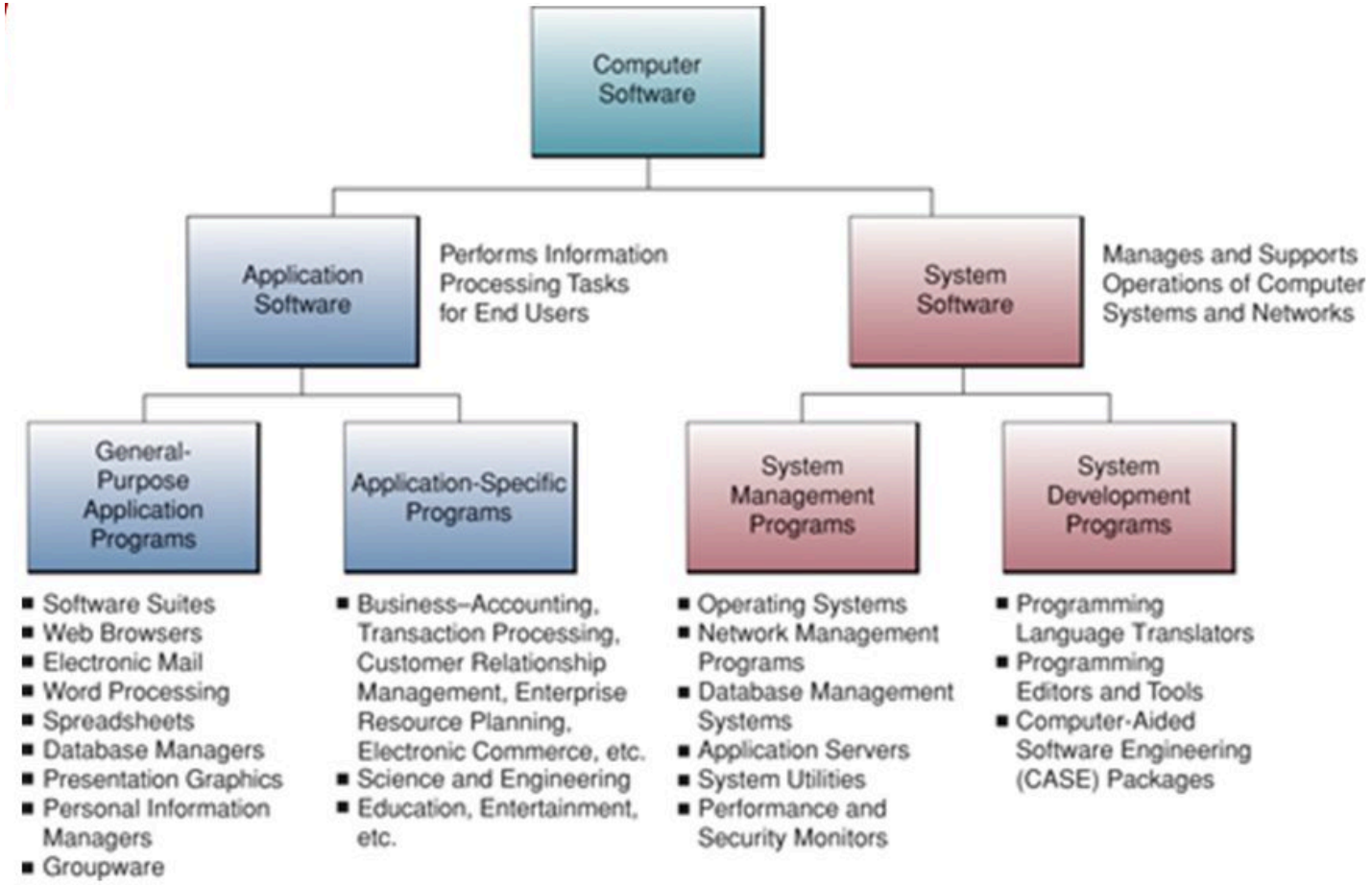
Quality practices will be introduced and reviewed to give you a perspective as to why some systems succeed and others fail.

Software Definition

Software is computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

Source: IEEE

Types of Software



Software Quality Definition

Software quality is:

(1) The degree to which a system, component, or process meets specified requirements.

(2) The degree to which a system, component, or process meets customer or user needs or expectations.

Source: IEEE

Software Quality Characteristics

- Good design – looks and style
- Good functionality – it does the job well
- Reliable – acceptable level of breakdowns or failure
- Consistency
- Durable – lasts as long as it should
- Good after sales service
- Value for money

Source: ISTQB

Software Quality Assurance Definition

Software quality assurance (SQA) is:

1. A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
2. A set of activities designed to evaluate the process by which the products are developed or manufactured.

Source: IEEE

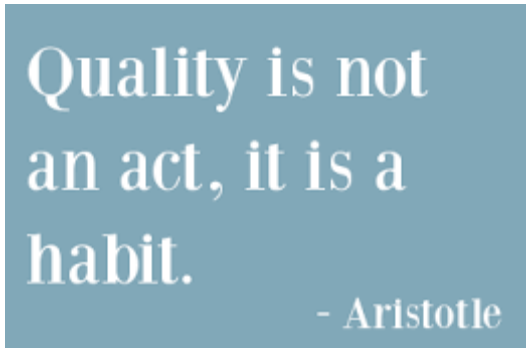
SQA Objectives in Software Development

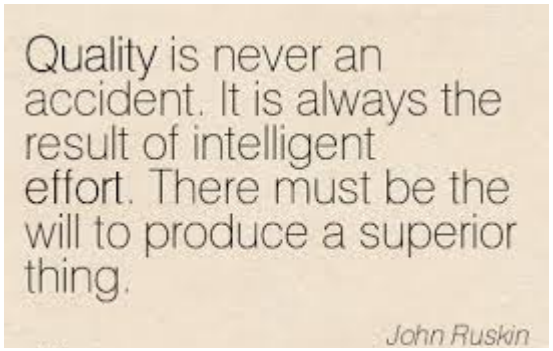
(1) Assuring an acceptable level of confidence that the software will conform to functional and technical requirements.
(2) Assuring an acceptable level of confidence that the software will conform to managerial scheduling and budgetary requirements.

Challenges of Software Quality Assurance

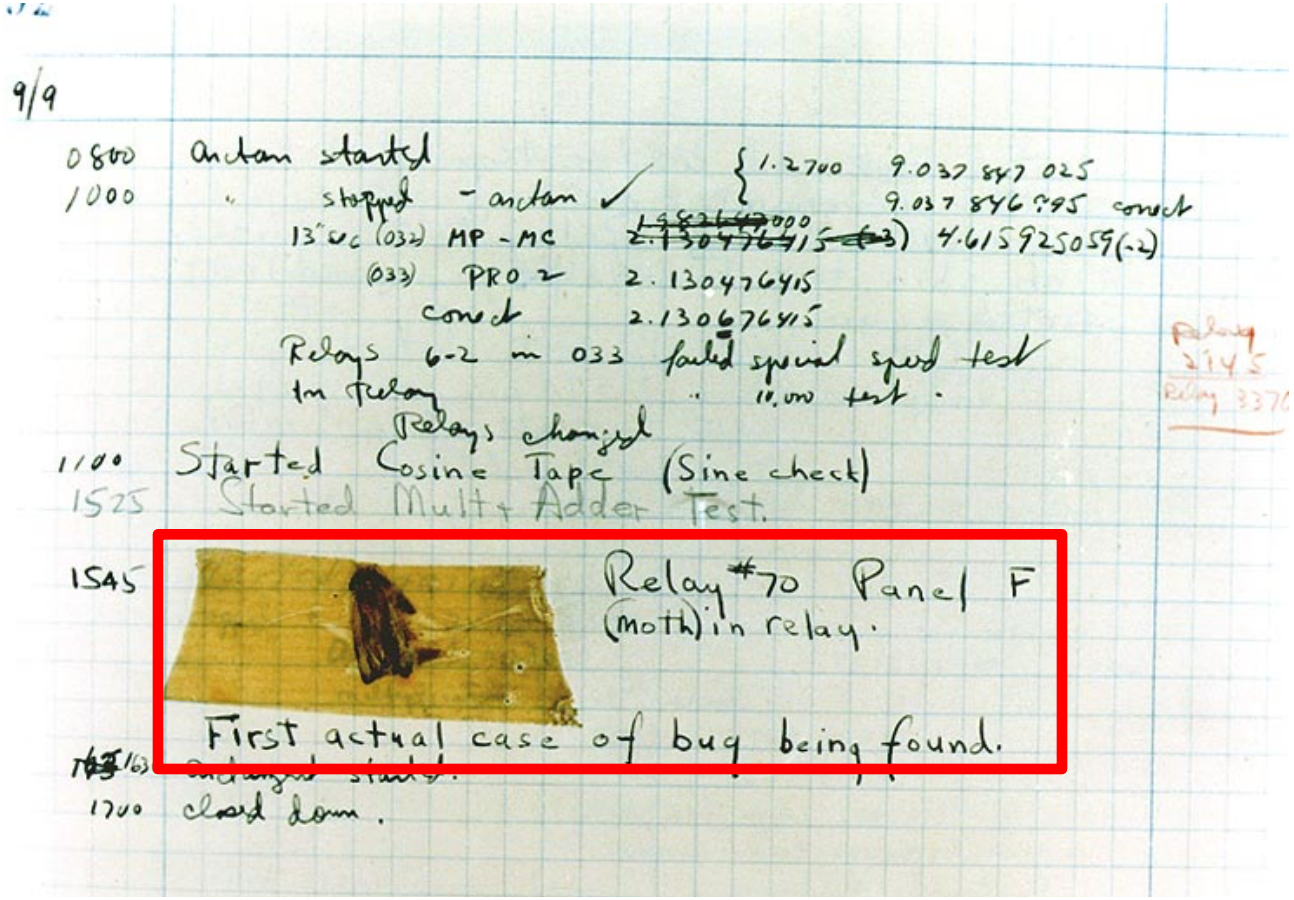
- High complexity
- Diverse and potentially very large number of users
- Invisibility of the product
- Limited opportunities to detect defects (“bugs”)

Quality Quotes





Software Bugs Are Everywhere!



In 1946, when Grace Hopper was released from active duty (from the Navy), she joined the Harvard Faculty at the Computation Laboratory where she continued her work on the Mark II and Mark III (electromechanical computers). Operators traced an error in the Mark II to a moth trapped in a relay, coining the term bug. This bug was carefully removed and taped to the log book. Stemming from the first bug, today we call errors or glitches in a program bugs.

Source: Wikipedia

Causes of Software Bugs

1. Faulty requirements definition
2. Client-developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Non-compliance with documentation and coding instructions
7. Shortcomings of the testing process
8. User interface and procedure errors
9. Documentation errors

Software Quality Assurance Components

- Reviews
- Software testing
- Procedures and work instructions
- Templates and checklists
- Staff training, retraining and certification
- Configuration management
- Documentation control

Reviews

Design Review – a systematic, comprehensive, and documented analysis of a design to determine its capability and adequacy to meet its requirements. A design review also serves to identify present and potential problems.

Source: IEEE

Reviews

Peer Reviews

- **Inspections** - a means of verifying products by manually examining the developing product, a piece at a time, by small groups of peers to ensure that it is correct and conforms to product specifications and requirements.
- **Walkthroughs** - a form of software peer review in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems.

Source: IEEE

Review Objectives

- To detect analysis and design errors as well as subjects where corrections and changes are required.
- To identify new risks likely to affect the project.
- To locate deviations from templates, style procedures and conventions.
- To approve the analysis or design of a product.
- To provide an informal meeting place for exchange of professional knowledge about methods, tools and techniques.

Verification and Validation

Verification – The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase

Validation - The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements

Verification – Did we build the software product right?
Validation – Did we build the right software product?

IEEE Std 610.12-1990 (IEEE 1990)

Definition and Objectives of Testing (Part 1)

Testing - An empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test. [Cem Kaner]

Objectives of Testing

- Finding defects which may get created while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To gain the confidence of the customers by providing them a quality product.

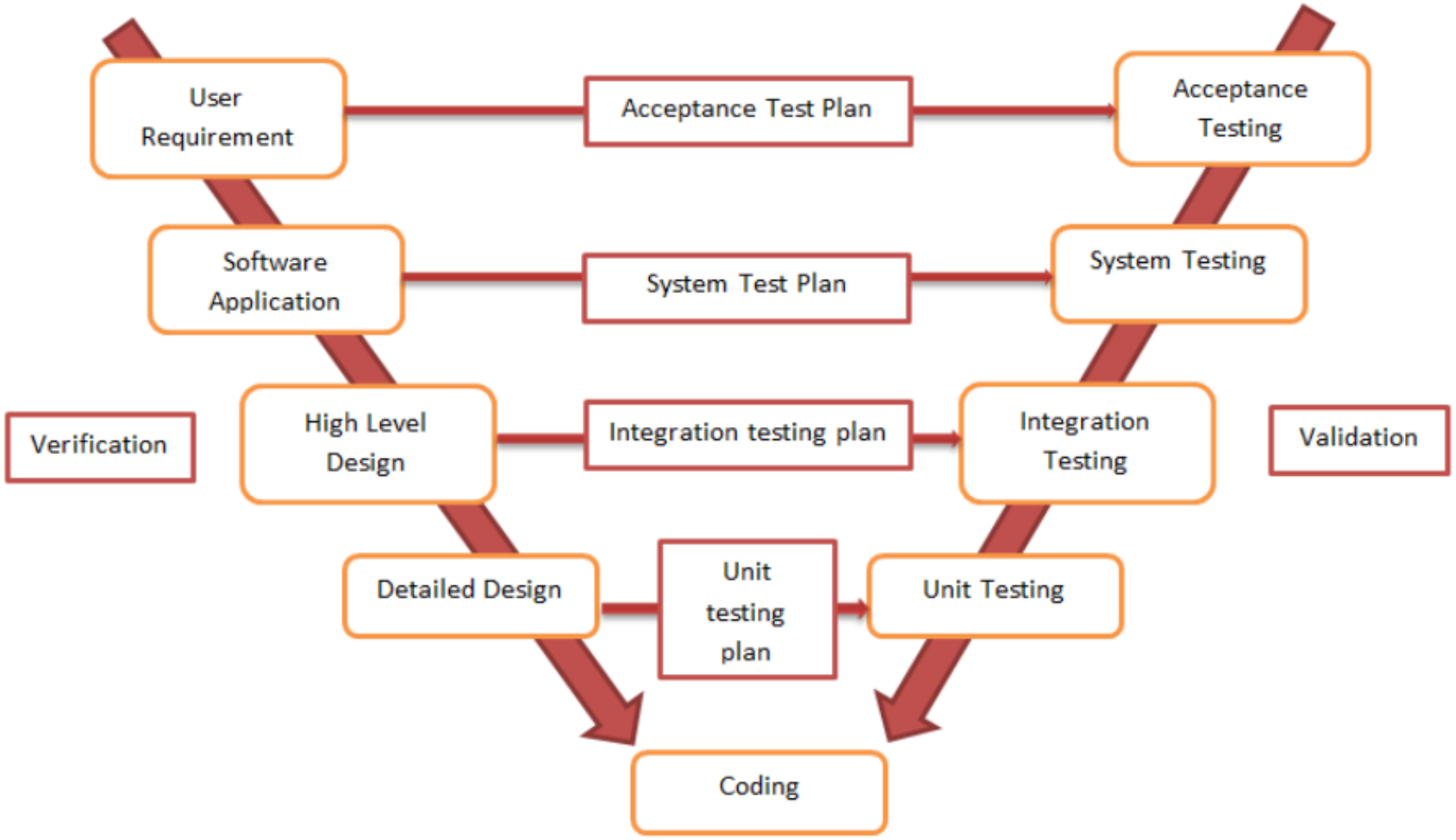
Definition of Testing (Part 2)

Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc. [James Bach]

- Testing is an open-ended investigation–think "Sherlock Holmes".
- "Evaluating" means making a value judgment; is it good? is it bad? pass? fail? how good? how bad?
- "Exploration" implies that testing is inherently exploratory. All testing is exploratory to some degree, but may also be structured by scripted elements.
- "Experimentation" implies interaction with a subject and observation of it as it is operating.

Source: James Bach and Michael Bolton, *Testing and Checking Refined*

V-Model


User Requirement
Acceptance Test Plan
Acceptance Testing
Software Application
System Test Plan
System Testing
Verification
High Level Design
Integration testing plan
Integration Testing
Validation
Detailed Design
Unit testing plan
Unit Testing
Coding

Testing Levels

Unit Testing refers to tests that verify the functionality of a specific section of code.

- These types of tests are usually written by developers as they work on code, to ensure that the specific function is working as expected.
- Unit testing might include static code analysis, peer code reviews, code coverage analysis and other software verification practices.

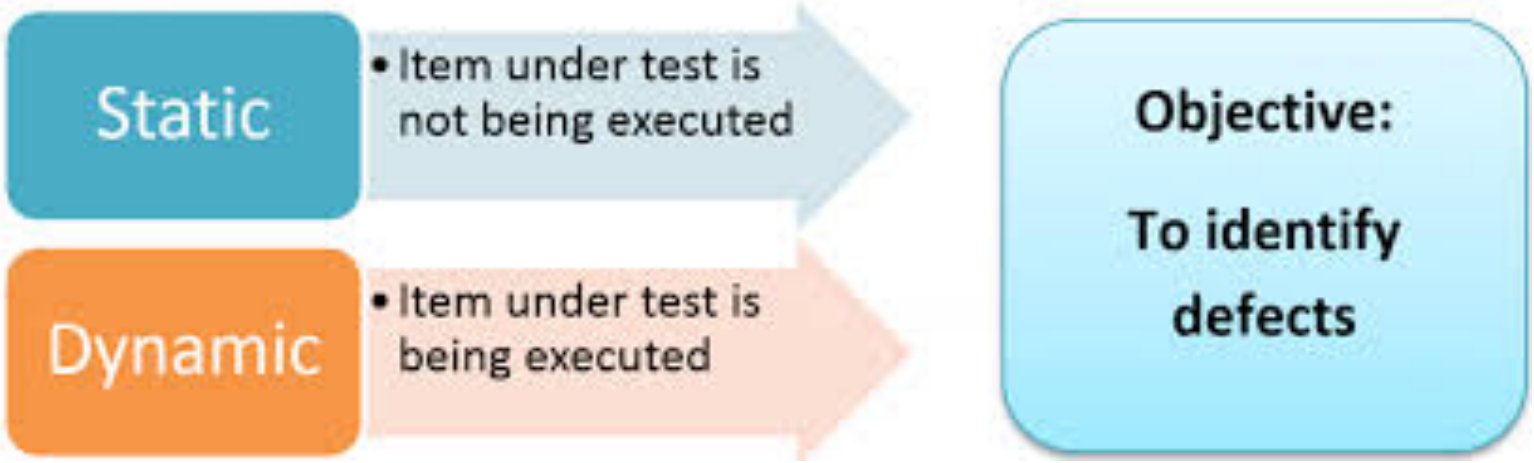
Integration Testing is any type of software testing that seeks to verify the interfaces between components against a software design.

- Integration testing works to expose defects in the interfaces and interaction between integrated components (modules).

System Testing tests a completely integrated system to verify that it meets its requirements.

(User) Acceptance Testing refers to tests that evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

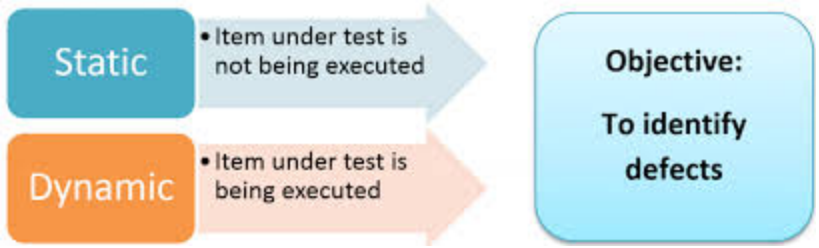
Testing Methods



Static testing involves **verification**, whereas dynamic testing involves **validation**.

Verification – Did we build the software product right?
Validation – Did we build the right software product?

Testing Methods



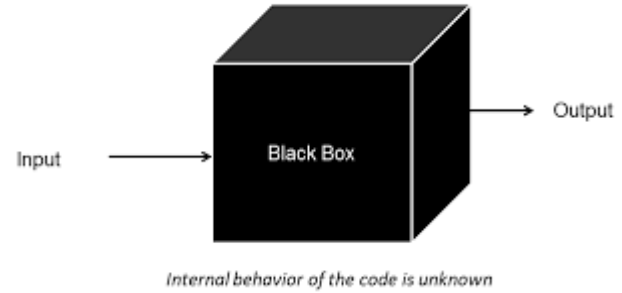
Static Testing - a form of software testing where the actual program is not used.

- Examples: Reviews, walkthroughs, inspections
- Examples: When programming tools/text editors check source code structure or compilers check syntax and data flow

Dynamic testing takes place when the program itself is run by actually executing test cases.

- Dynamic testing may begin before the program is 100% complete in order to test particular sections of code.

Testing Methods



White-box Testing tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user.

- An internal perspective of the system, as well as programming skills, are used to design test cases.
- The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation.

- The testers are only aware of what the software is supposed to do, not how it does it.

Testing Types

Automation Testing is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes.

- Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place.

Exploratory Testing is about exploring, finding out about the software, what it does, what it doesn't do, what works and what doesn't work.

- The tester is constantly making decisions about what to test next and where to spend time.
- It's a hands-on approach in which testers are involved in minimum planning and maximum test execution.

Regression Testing – Regression testing focuses on finding defects after a code change has occurred.

- Regressions occur whenever software functionality that was previously working correctly stops working as intended. Common methods of regression testing include re-running previous sets of test cases and checking whether previously fixed faults have re-emerged.

Students Ask What?