Software Quality Assurance; Software Testing

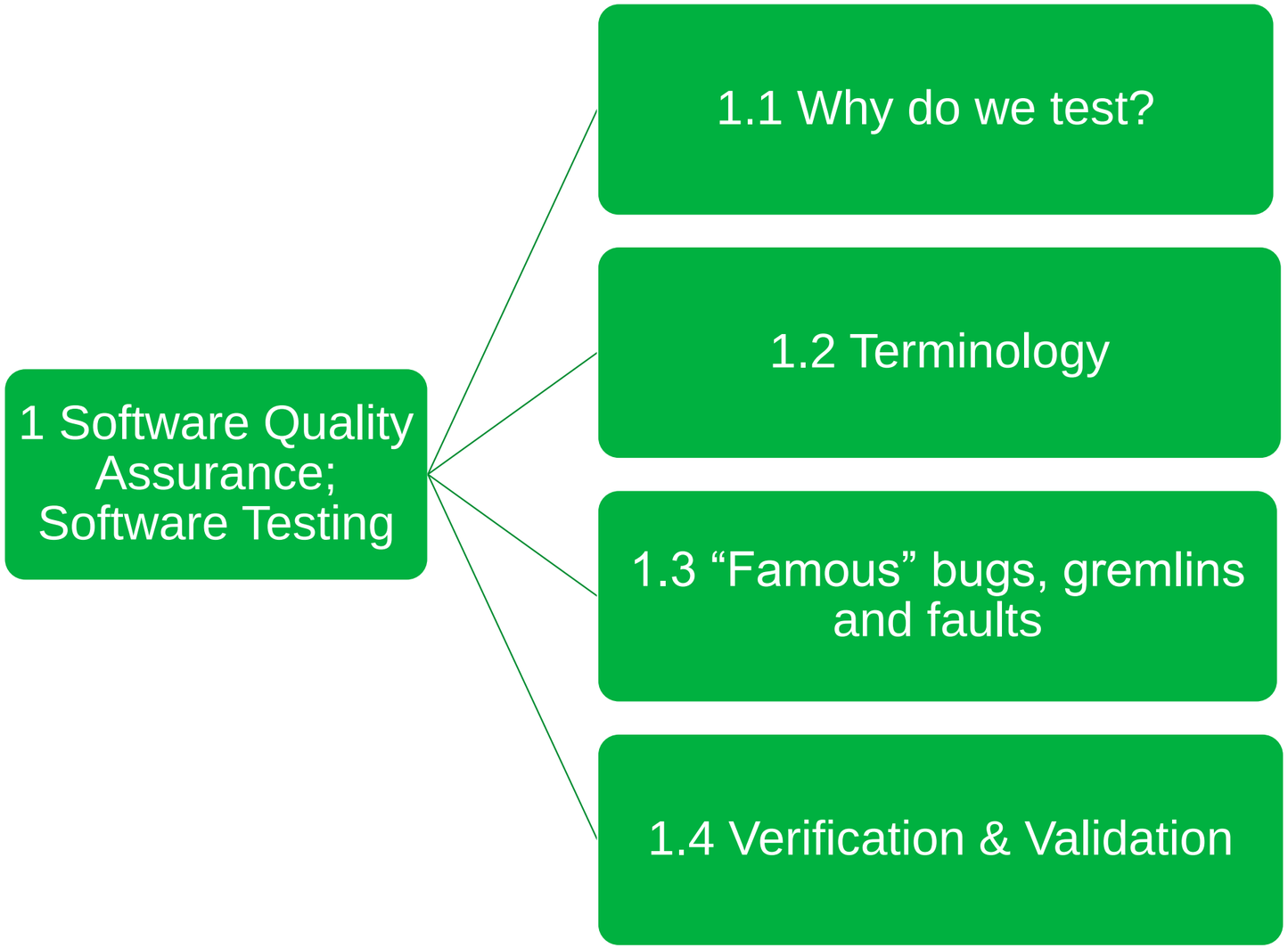
Dr. Salim Saay
Lecturer at the Department of Computer and Information System
Uuniversity of Limerick
Salim.saay@ul.ie

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Signalling Used in Slides

A rectangle like this one, highlights information or concepts of particular importance

Overview


1 Software Quality Assurance; Software Testing
1.1 Why do we test?
1.2 Terminology
1.3 “Famous” bugs, gremlins and faults
1.4 Verification & Validation

1.1 Why Do We Test?

Software Quality Assurance – Why do we test?

- It is not surprising you that we have software everywhere around us and errors can happen to them.
  - Errors can cause inconvenience.
  - Errors can reduce customer satisfaction.
  - Errors can be expensive.
  - Errors can be fatal.
- Eg. a drone that using for the military attack and manage by a software, a small error can kill innocent people, consider if error happen in banking system …

Why Do We Test?

- Testing is an activity which is part of every engineering discipline
  - Testing in manufacturing, medicine, machinery building etc.
  - Before we use the product we test that
- **Humans → prone to mistakes**
  - We test to find our mistakes
  - the design a good test case is the most important

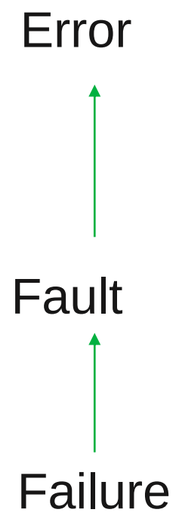
The true subject matter of the design of test cases is the most important.

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

1.2 Terminology

Terminology

- ***Software Fault:*** A static defect in the software.
- ***Software Error:*** An incorrect internal state that is the manifestation of some fault.
- ***Software Failure:*** External, **incorrect behavior** with respect to the requirements or another description of the expected behaviour.



There is many different definition but these definitions stand for this module.

I have a bad headache and a sore throat (the symptoms) -- the ***failures***

Hmm … lets discover the cause for the symptoms -- the ***fault***

I'll run some tests to look for anomalous internal conditions (high temperature, chest infection. …) -- the **errors**

Software Failure: External, incorrect behavior with respect to the requirements or another description of the expected behaviour.

Software Fault: A static defect in the software.

Software Error: An incorrect internal state that is the manifestation of some fault.

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

A More Precise Understanding of Terminology

```
/**
 * Counts zeroes in an array
 *
 * @param x array to count zeroes in
 * @return number of occurrences of 0 in x
 * @throws NullPointerException if x is null
 */
public static int numZero (int[] x)
{
   int count = 0;
   for (int i = 1; i < x.length; i++)
   {
      if (x[i] == 0) count++;
   }
   return count;
}
```

A program state is defined during the execution of a program as the current value of all live variables and the current location (Program Counter)

The Program Counter can be described as:

- **line number in a file: (PC=4), or**
- **(PC= “int count=0”)**

A More Precise Understanding of Terminology

```
/**
 * Counts zeroes in an array
 *
 * @param x array to count zeroes in
 * @return number of occurrences of 0 in x
 * @throws NullPointerException if x is null
 */
public static int numZero (int[] x)
{
  int count = 0;
  for (int i = 1; i < x.length; i++)
  {
    if (x[i] == 0) count++;
  }
  return count;
}
```

Test case 1: numZero([2, 7, 0]):

- Correctly evaluates to 1
- It's an error state
- Error state does not propagate to the output
- We observe no failure

Test case 2: numZero([0, 7, 2]):

- …
- …
- …
- …

Not all inputs will "trigger" a fault into causing a failure

- https://www.youtube.com/watch?v=zAty8Rpg92I
- https://youtu.be/yZwrOsnKypE

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Famous” **Bugs**, and Faults

Bugs 1/2

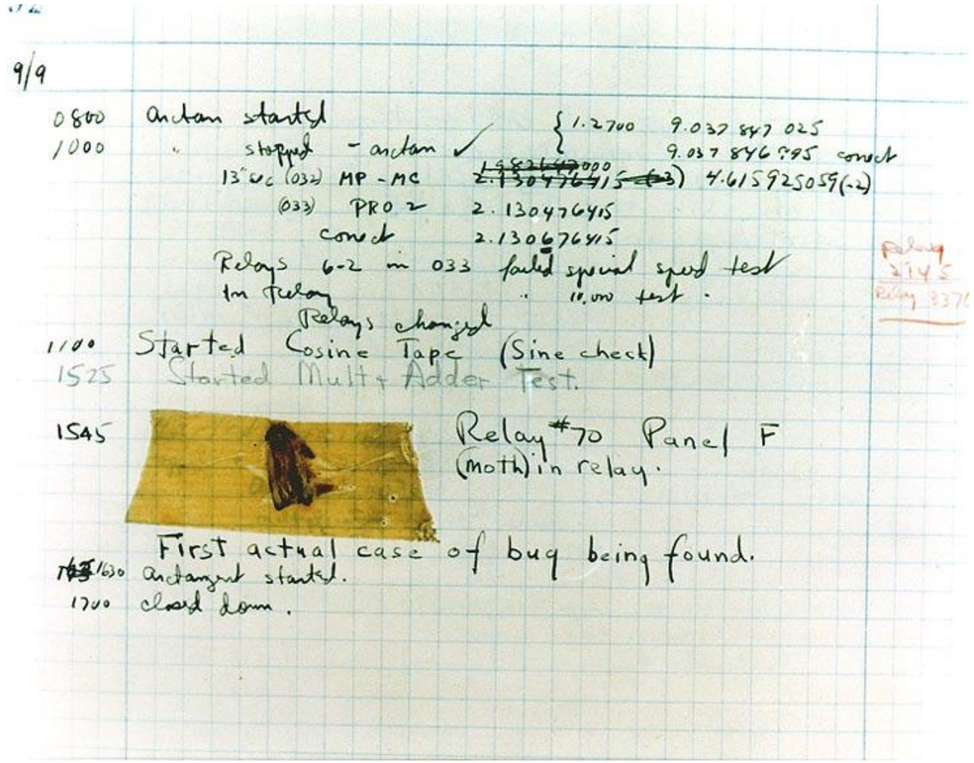
- The term **bug is often** used **informally** to refer to all three of **fault, error, and failure**.
  - We will usually use the specific term, and avoid using “**bug**.”
  - When using “bug” we will specify what we mean by it.
- **Engineers** have been talking about **bugs** for **100+ of years**
  - For **example Thomas Edison in** a letter he wrote to **Theodore Puscas** in 1878 **talks about bugs**:
    - '**Bugs**' -- as such **little faults** and difficulties are called -- show themselves and months of intense watching, study and labor are requisite before commercial success or failure is certainly reached
- Bug is an old term in machinery

Gremlins and Bugs 2/2

- **Grace Hoper**

  - On September 9, **1947** At 3:45 p.m., First Instance of Actual Computer Bug Being Found
  - **Grace Murray Hopper records** 'the first computer bug' in the Harvard Mark II **computer's log book**.
  - The problem was traced to **a moth** stuck between relay contacts in the computer, which Hopper duly taped into the Mark II's log book with the explanation: “First actual case of bug being found.”
  - The bug was actually found by others but Hopper made the logbook entry.

Grace Hopper



https://www.computerhistory.org/tdih/september/9/

Small Faults and Big Crashes.

370 million dollars worth of fireworks because of a software bug. (Source: ESA)

http://www.youtube.com/watch?v=gp_D8r-2hwk

Ariane 5 is a European rocket developed and operated by Arianespace for the European Space Agency (ESA).

Arianne 5

- **On June 4th, 1996** and only 30 seconds after the launch, the Ariane 5 rocket began to disintegrate slowly until its final explosion.
- Simulations with a similiar flight system and the same conditions revealed that in the rocket's **software** (which came from Ariane 4), a **64-bit variable** with decimals was transformed into a 16-bit variable without decimals.

- Video explaining what happened:https://www.youtube.com/watch?v=W3YJeoYgozw
- Slides for the above video: https://www.slideshare.net/sommerville-videos/ariane-5-launcher-failure-30036896

Mars Polar Lander

- NASA study project in land of Mars.
- Part of the software assumed the values were on the imperial system (inches/feet) another part in metric
- Loss of more than $100 million. The total cost was US$165 million.

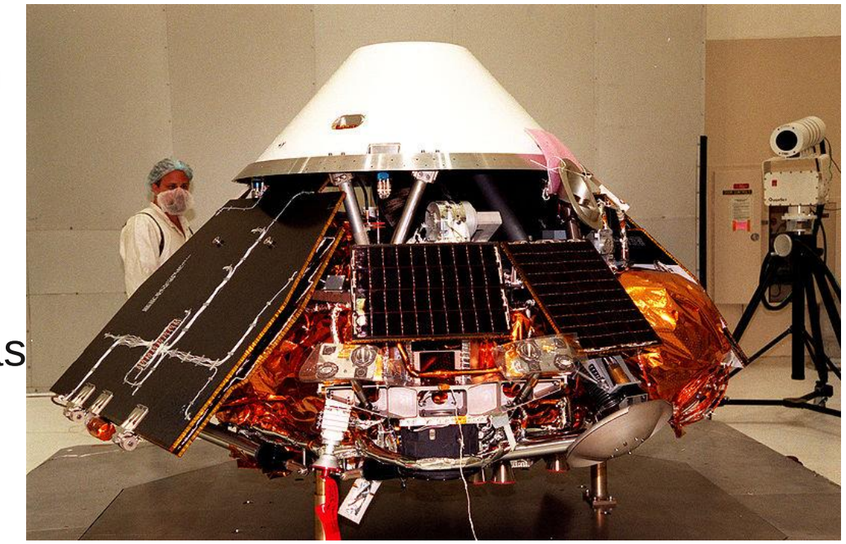


Fig 1: Mars Polar Lander being Tested

https://youtu.be/xyMKBHnc1U8

Verification & Validation

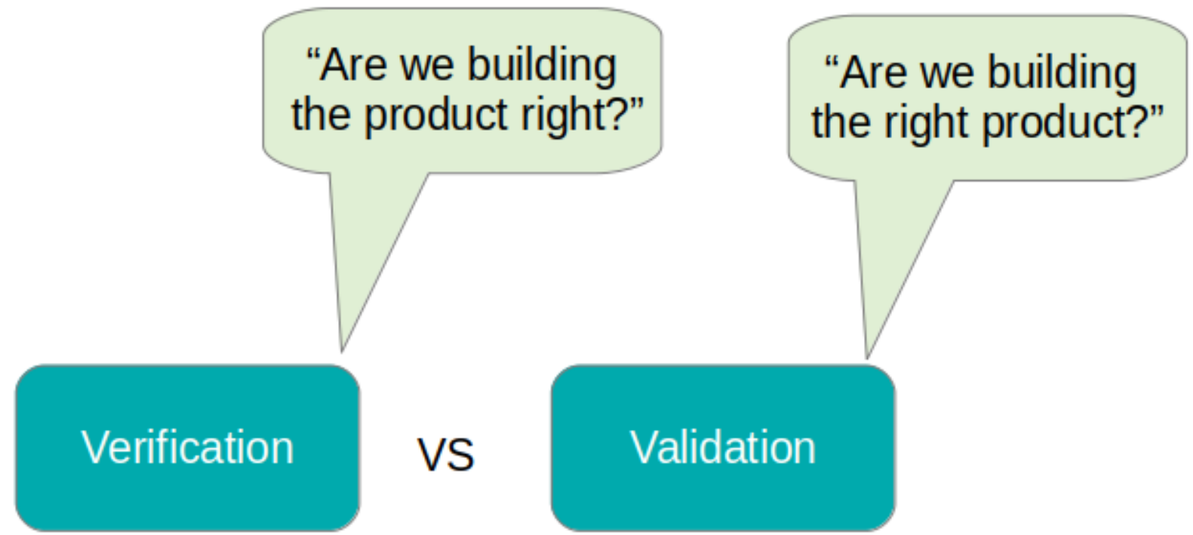
UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Testing Goals

- The true subject matter of the tester is not testing, **but the design of test cases**.
- To design a good test case:
  - What are we trying to do when we test?
  - What are our goals?

Verification & Validation

- **Verification:** To ensures that the product is built **according to the requirements** and design specifications.
- **Validation:** To ensure that the **product actually meets** the **user's needs** and that the specifications were correct in the first place.



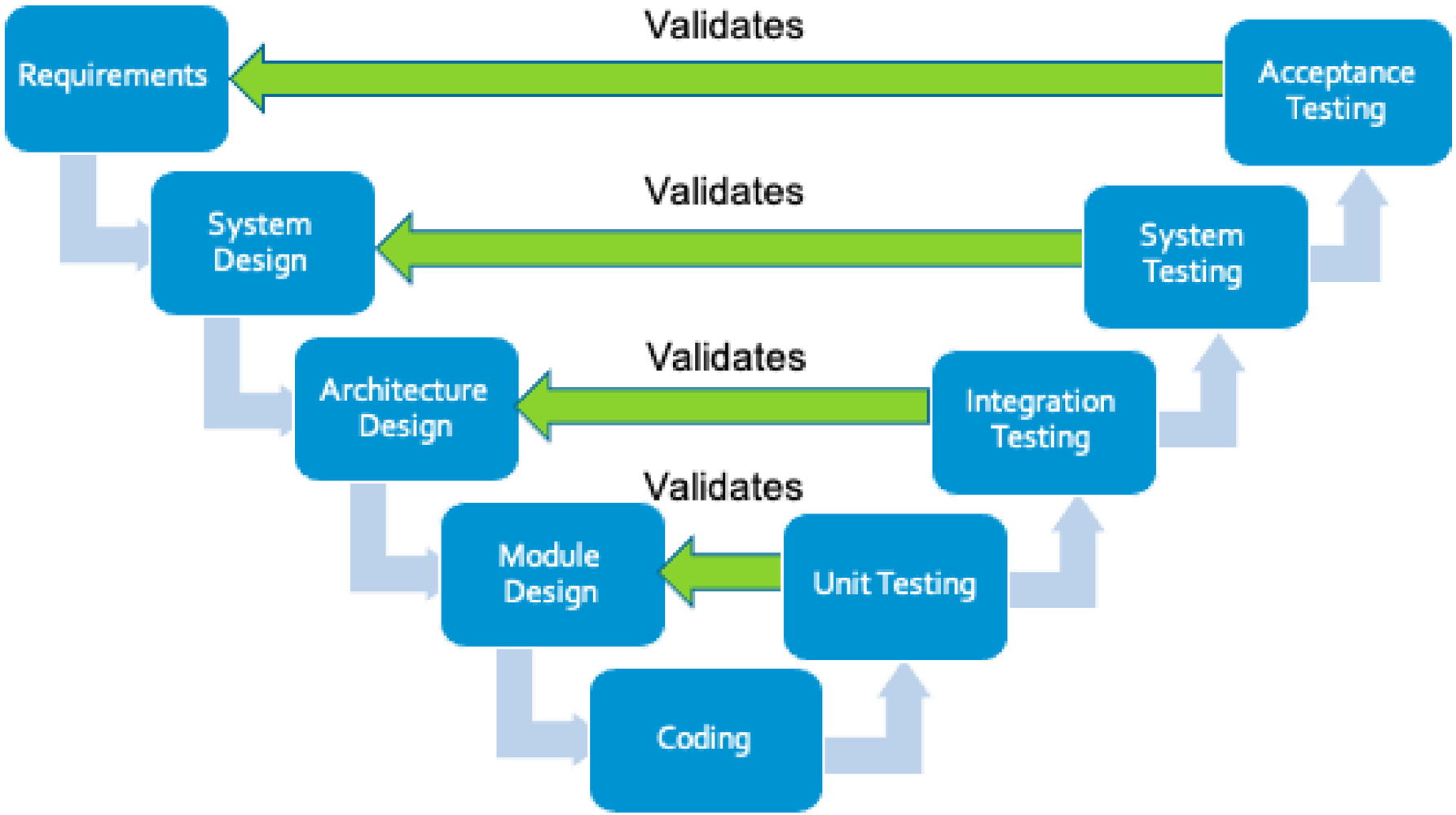
Verification & Validation

https://www.youtube.com/watch?v=ckMqOXQCb1U

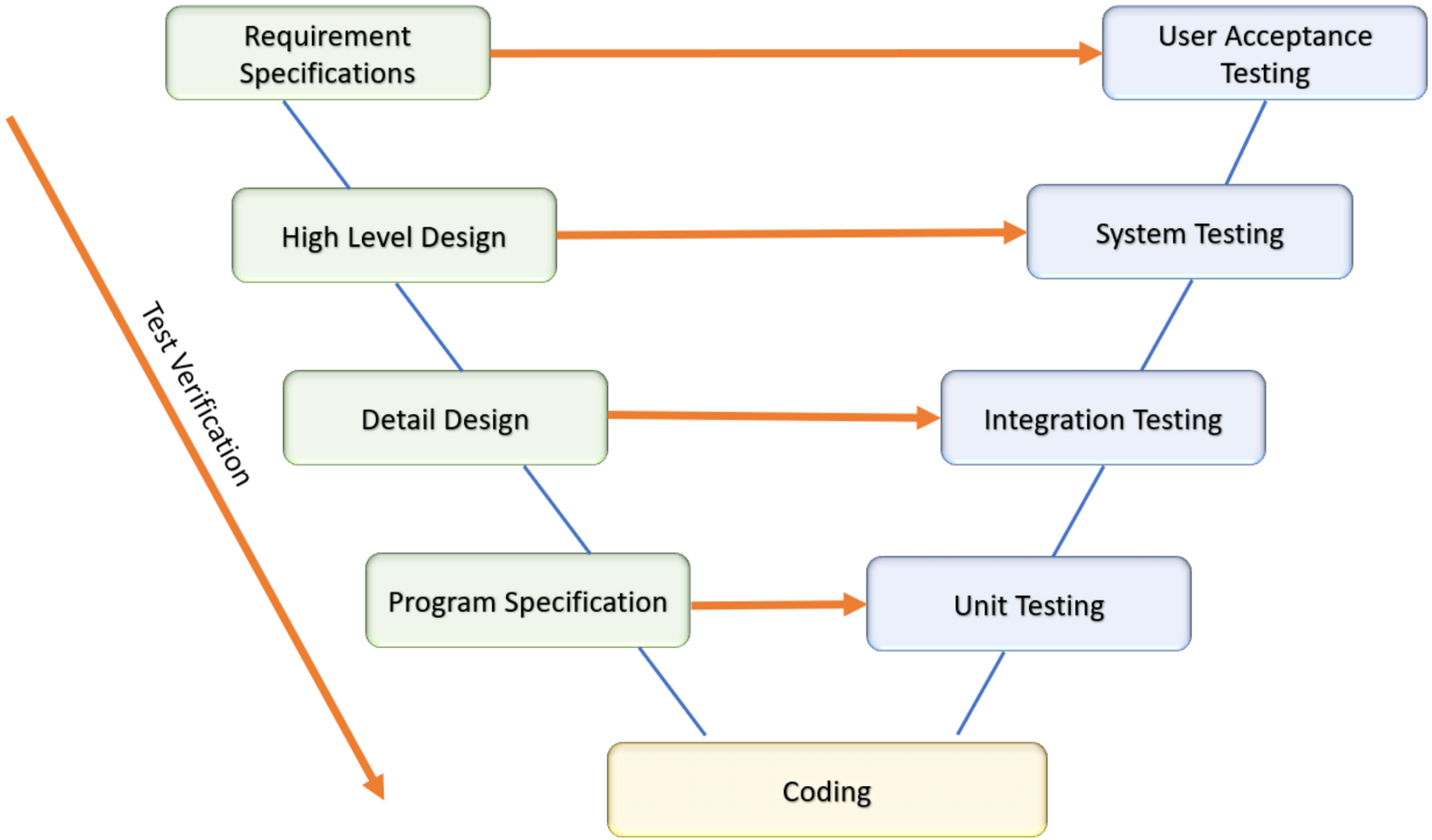
Verification & Validation

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
Question	Are we building the product *right*?	Are we building the *right* product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.

Verification & Validation



Verification & Validation



Verification & Validation

- **Verification** tends to be a more **technical activity**.
- It uses **knowledge** about the **individual software artifacts**, requirements and specifications.

- **Validation** usually depends on **domain knowledge (environment)**
- For e.g. validation asks whether the users are satisfied**.**

Independent Verification and Validation (IV&V)

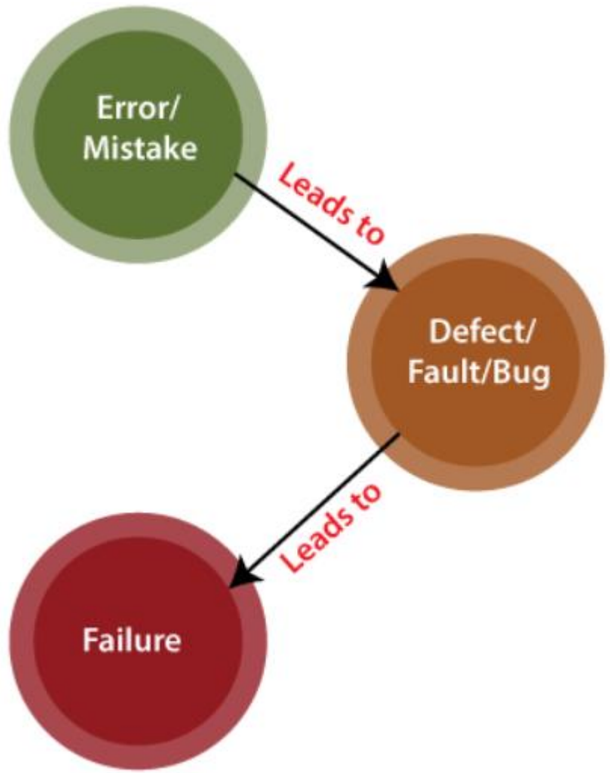
- Independent means that the evaluation was done by someone else than the developers of the software
- The IV&V team can be
  - within the same project,
  - within the same company,
  - or a completely different company

What is software testing?

We put a question in the forum and you can write your answer

Conclusion

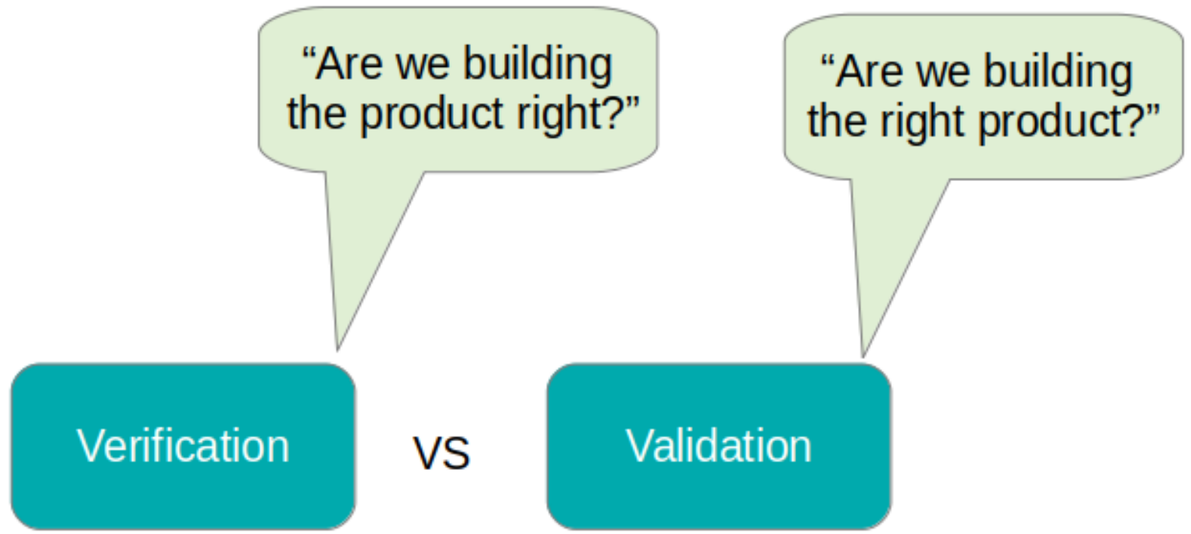
- Errors : internal mistake, including human mistake in the code, errors can result fault.
- Fault: fault is the cause of failure, for example incorrect power plugin cause blue screen
- Failure: incorrect behavior of computer for example blue screen of a computer is failure
- Bugs: It is an informal name specified to the **defect**.
- Defect: The Defect is the difference between the **actual outcomes** and **expected outputs**.



https://www.javatpoint.com/bug-vs-defect-vs-error-vs-fault-vs-failure

Conclusion

Evaluation and Validation



References

- (Alexander and Beus-Dukic, 2009)
- Ch. 4 “Context, Interfaces, Scope”
- (Wiegers and Beatty 2013)
- Ch. 5 “Establishing the business requirements”
- Ch. 28 “Change happens”

Thank you

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

University of Limerick,
Limerick, V94 T9PX,
Ireland.

Ollscoil Luimnigh,
Luimneach,
V94 T9PX, Éire.

+353 (0) 61 202020

ul.ie