Software for a better world

A World Leading SFI Research Centre

Science Foundation Ireland sfi For what's next

CS4004
Software Testing & Inspection
Lecture 2 – Why Do We Test

Dr Faeq Alrimawi

HOST INSTITUTION

UNIVERSITY OF LIMERICK OLLSCOIL LUIMNIGH

PARTNER INSTITUTIONS

NUI Galway OÉ Gaillimh · Maynooth University · Trinity College Dublin, Coláiste na Tríonóide, Baile Átha Cliath, The University of Dublin · University College Dublin, An Coláiste Ollscoile, Baile Átha Cliath · UCC · MTU, Ollscoil Teicneolaíochta na Mumhan, Munster Technological University · DCU · Waterford Institute of Technology · Dundalk Institute of Technology, Dún Dealgan · LIT · GMIT

Are You on Brightspace?

If you still **CANNOT** see the module on Brightspace,
please **email me with your student ID**

My email: faeq.alrimawi@ul.ie

Today, we are covering...

- Why Do We Test?
- Terminology
- “Famous” bugs, gremlins and faults
- Verification & Validation

Why Do We Test? (1-2)

- It is **not surprising** you that we have software everywhere around us and **errors can happen** to them. Errors can:
  - Cause inconvenience
  - Reduce customer satisfaction
  - Be expensive
  - Be fatal
- For example, a drone used for military attacks
  - A small error **can kill innocent people**
- Consider if an error happens in **banking system**? Recent BoI incident!

Why Do We Test? (1-2)

- It i u t
  an n t
  - fac
- Fo u
  - nn

https://www.thejournal.ie/bank-of-ireland-8-6143756-Aug2023/

- Consider if an error happens in **banking system**? Recent BoI incident!

Why Do We Test? (2-2)

- 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

Designing good test cases is the best!

Today, we are covering…

- Why Do We Test?
- Terminology
- “Famous” bugs, gremlins and faults
- Verification & Validation

Terminology

- **Fault**: A **static defect** in the software

- **Error**: An **incorrect internal state** that is the manifestation of some **fault**

- **Failure**: An **incorrect external behaviour** with respect to the requirements or another description of the expected behaviour

Many definitions exist, but we'll use these

Terminology - Patient Analogy

I have fever, headache, sore throat

I'll run some tests to look for internal anomalous conditions, e.g., chest infection, high temp.

Hmm… let's discover the root cause for you symptoms

Failure

Incorrect **External** Behaviour

Error

Incorrect **Internal** State

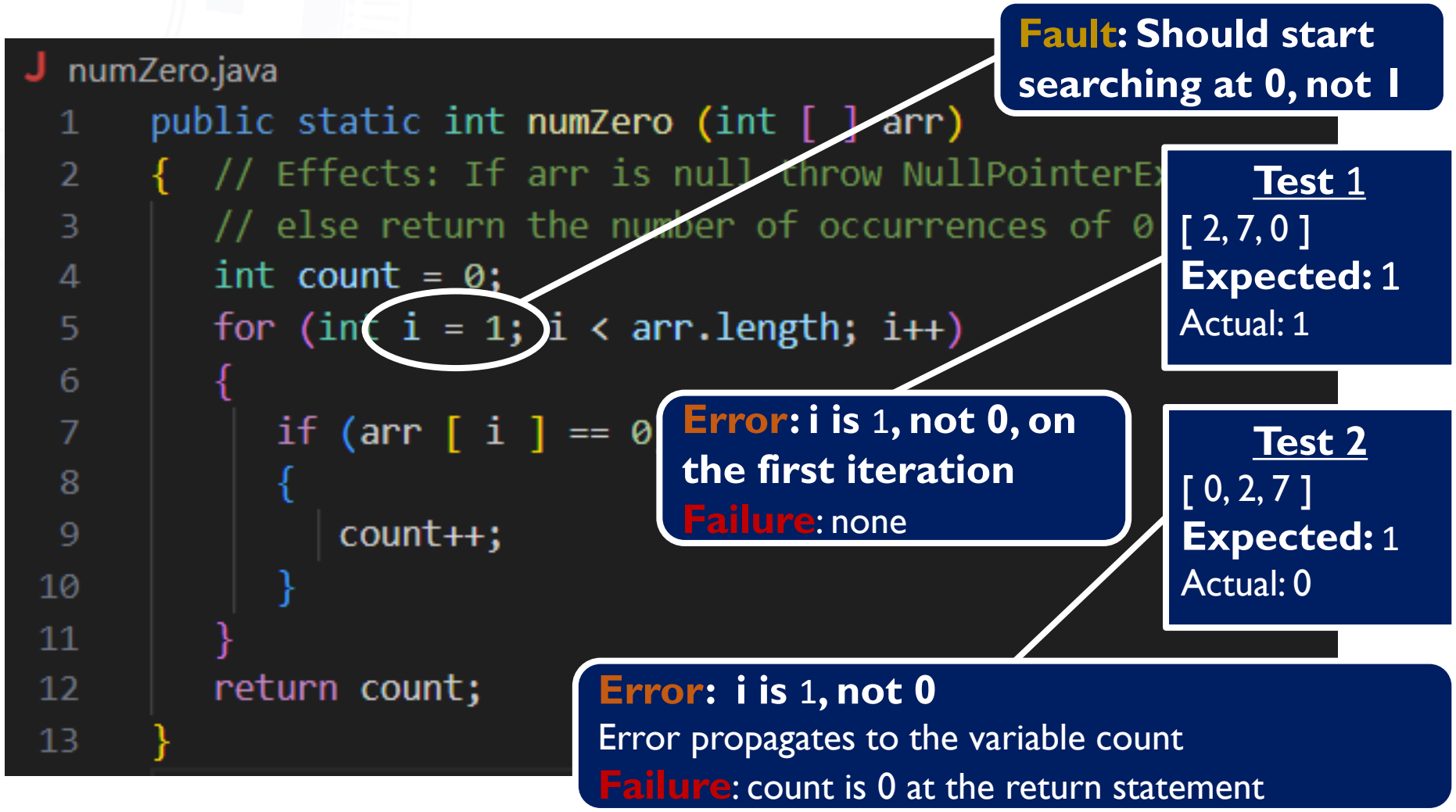
Fault

Static Defect

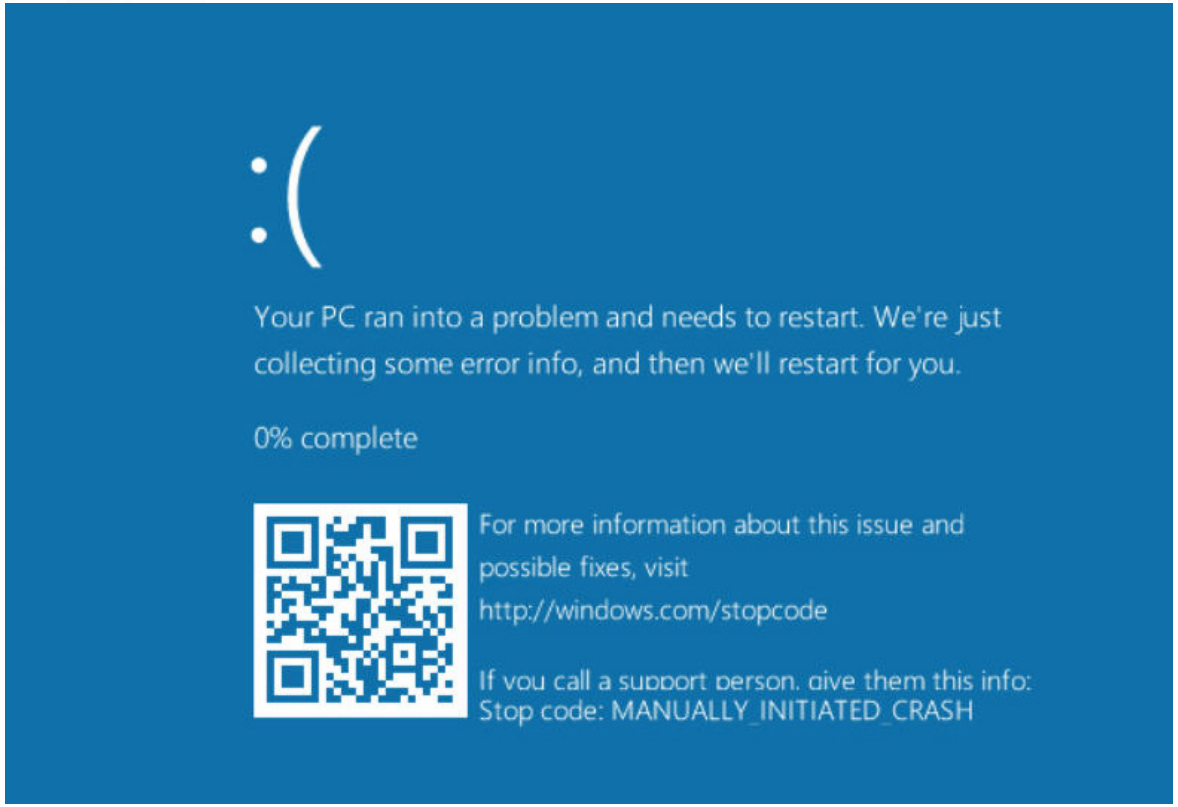
Dis-Analogy: Most medical problems result from external attacks (bacteria, viruses) or physical degradation as we age.

Software faults were there at the beginning and do not "appear" when a part wears out.

Terminology - An Example



More on Faults, Errors & Failure



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

Today, we are covering...

- Why Do We Test?
- Terminology
- “Famous” bugs, gremlins and faults
- Verification & Validation

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

- **Old term!** 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:

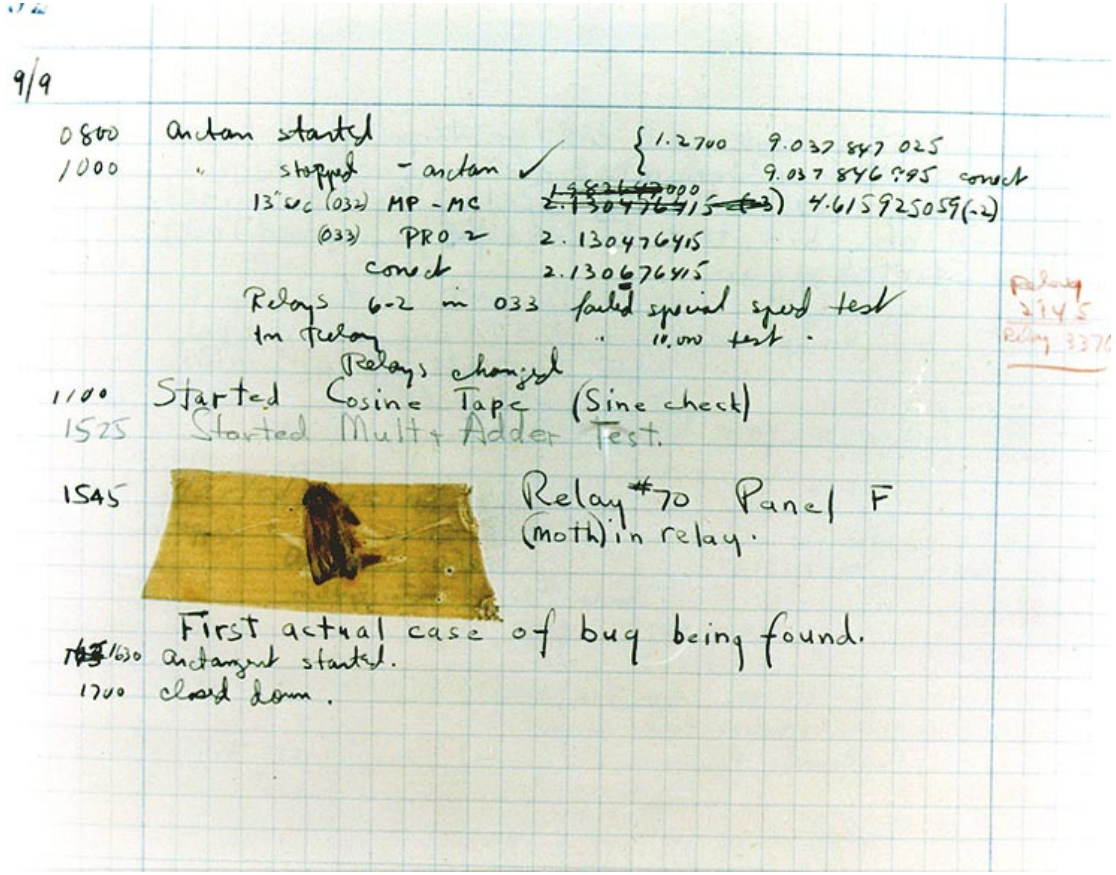
    "'Bugs' -- as such little faults and difficulties are called -- show themselves and months of intense watching, study and labour are requisite before commercial success or failure is certainly reached"

Bugs (2-2)

Grace Hopper Records First ‘Bug’

Grace Hopper: Computer Scientist

- 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.



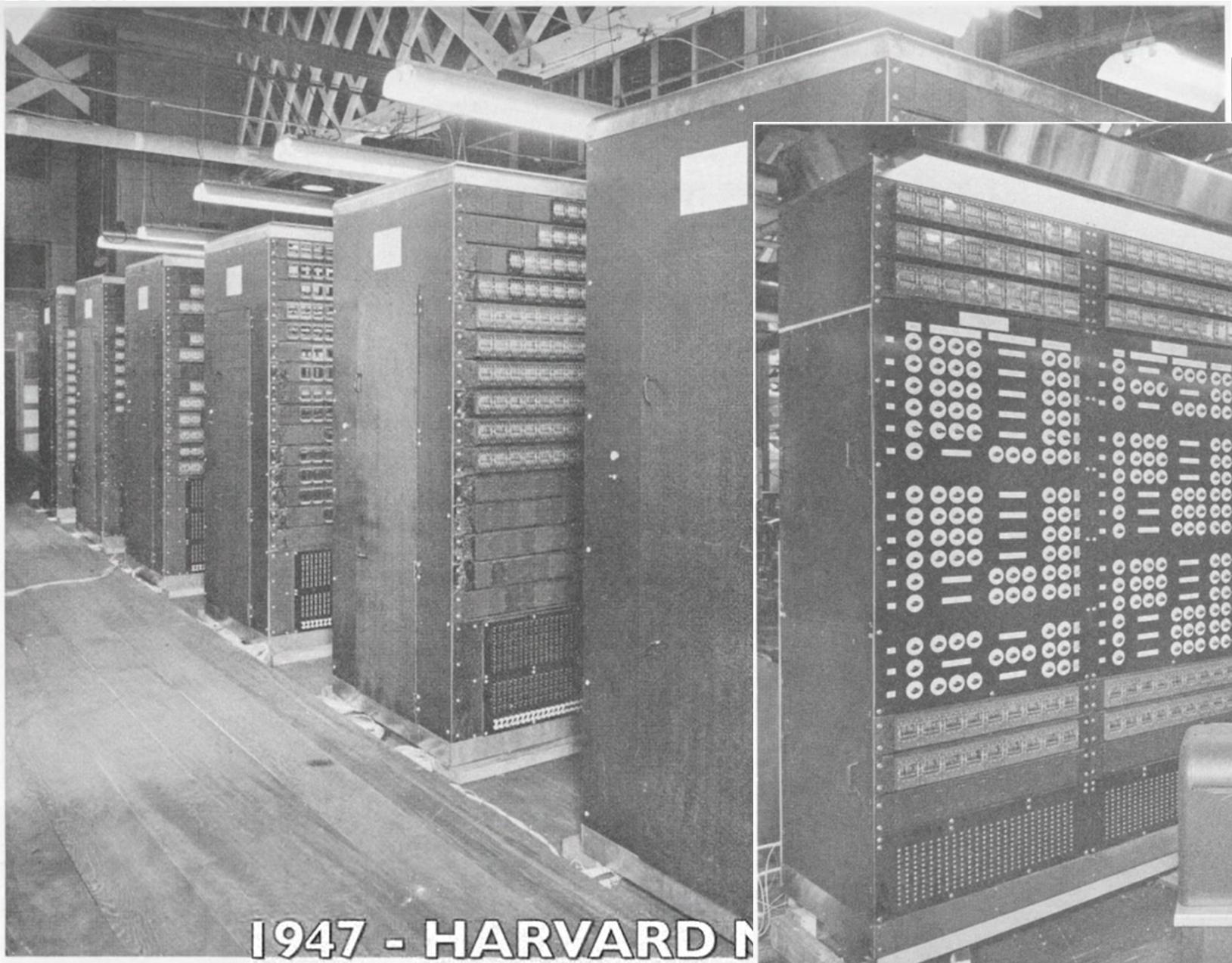


Figure 5 Mark II: Relay Cubicles

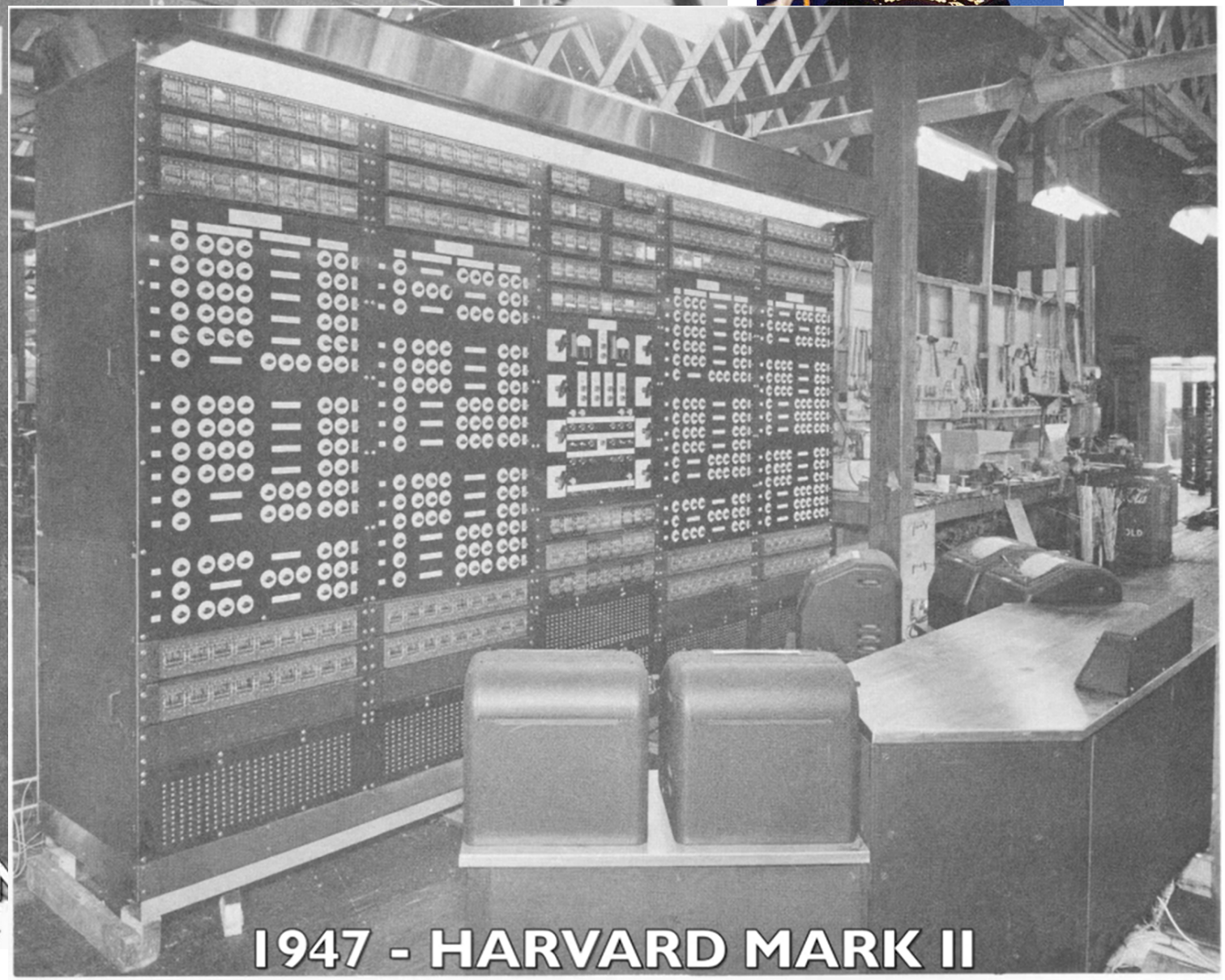


Figure 1 Mark II: Manual Control Panel and Printer Table

Tiny Fault, Catastrophic Failure

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

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

Video of the explosion: http://www.youtube.com/watch?v=gp_D8r-2hwk

Ariane 5 – A Software Failure

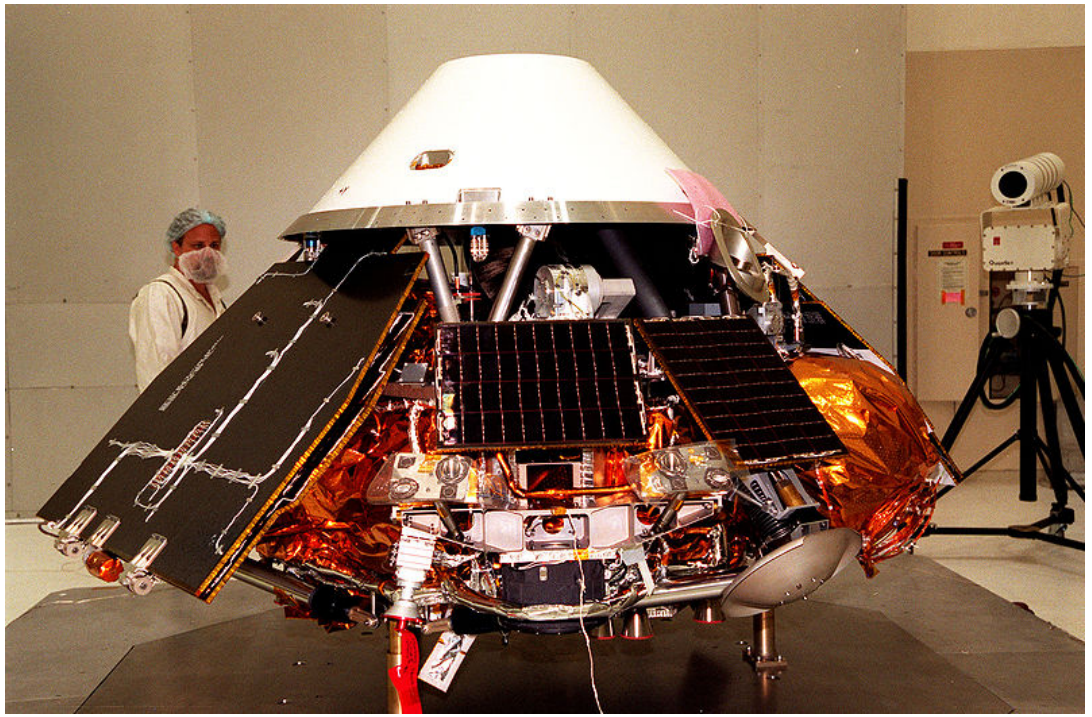
- On June 4th, **1996** and only 30 seconds after the launch, the **Ariane 5 rocket** began to disintegrate slowly until its final **explosion**
- Simulations with a similar flight system and the same conditions revealed that **in the rocket's software** (which came from Ariane 4), **a 64-bit floating number was transformed into a 16-bit integer (number became too big!)**

More details

- **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 - Another Failure

- NASA's project to land on 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**.
- https://youtu.be/xyMKBHnc1U8



Mars Polar Lander being Tested

Other Spectacular Soft. Failures

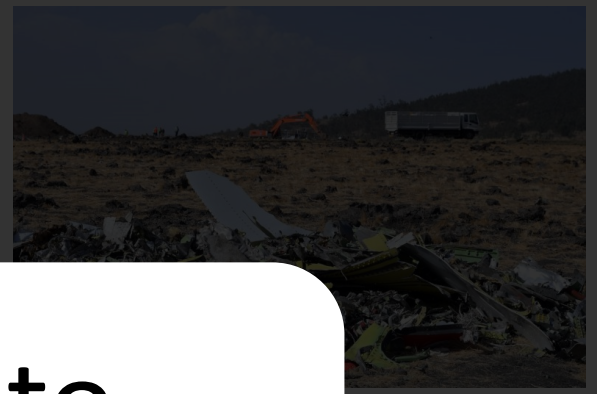
Boeing A220: Engines failed after software update allowed excessive vibrations

Toyota brakes: Dozens dead, thousands of crashes

Northeast blackout: 50 million people, $6 billion USD lost … alarm system failed

Other Spectacular Soft. Failures

Boeing A220: Engines failed after software update allowed excessive vibrations



ashes

Software Testers mission is to
find Faults before Faults find Users

Northeast blackout: 50 million people, $6 billion USD lost … alarm system failed

Today, we are covering…

- Why Do We Test?
- Terminology
- “Famous” bugs, gremlins and faults
- Verification & Validation

Testing Goals

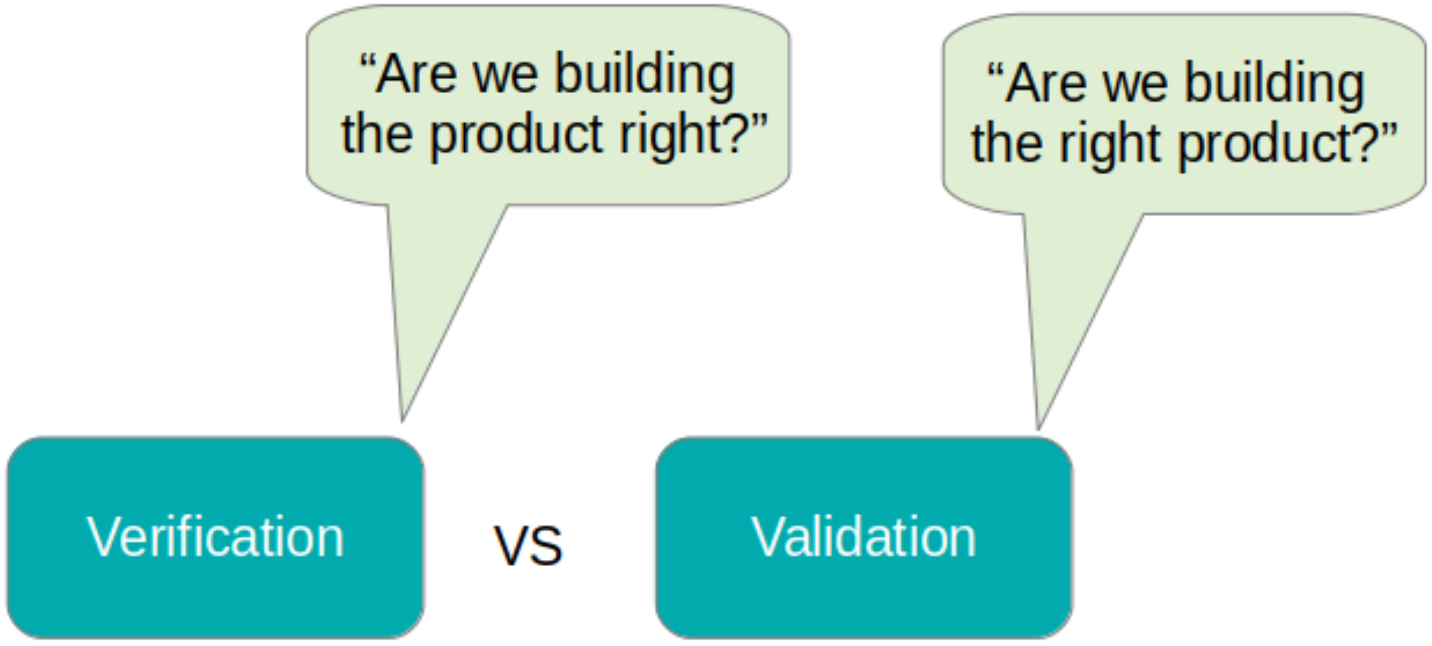
- The true subject matter of the tester is not testing, but **the design of test cases**

- To design a good test case:
  - What are we trying to do when we test?
  - What are our goals?

Verification & Validation (V&V)

Verification: To ensure that the product is meets its **requirements**

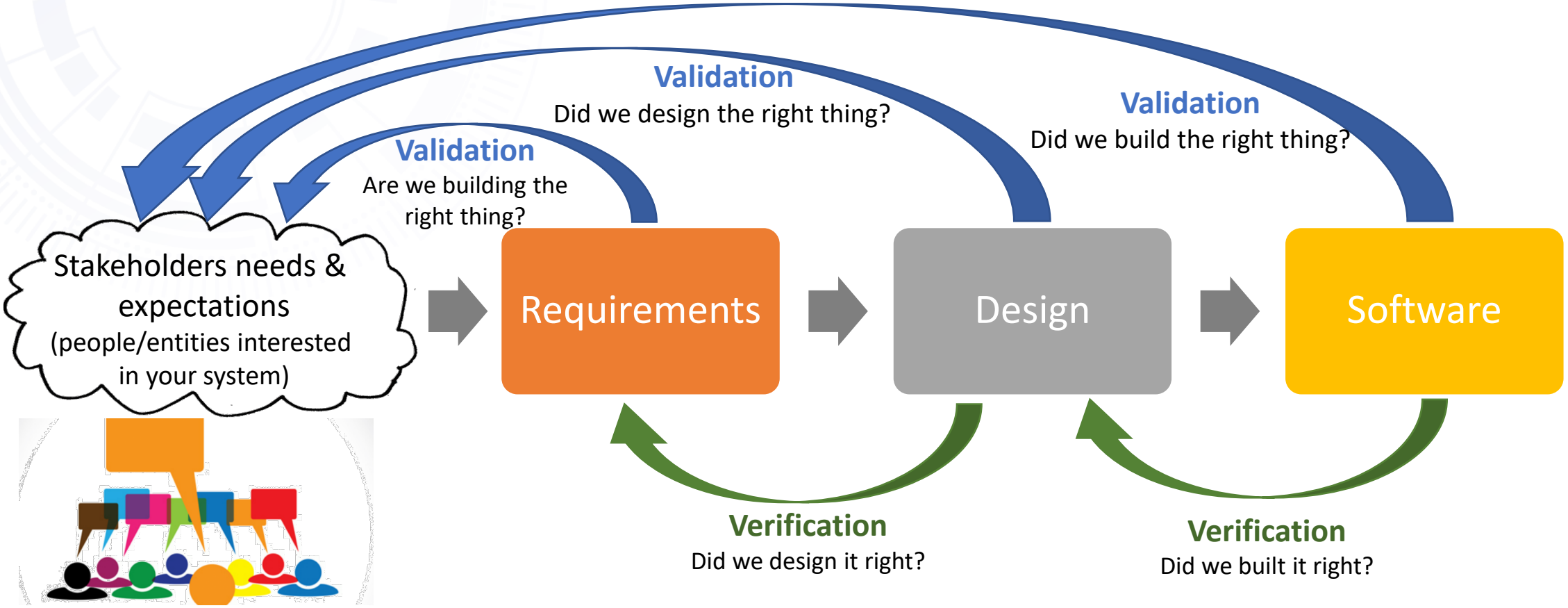
Validation: To ensure that the product **meets the user's needs**



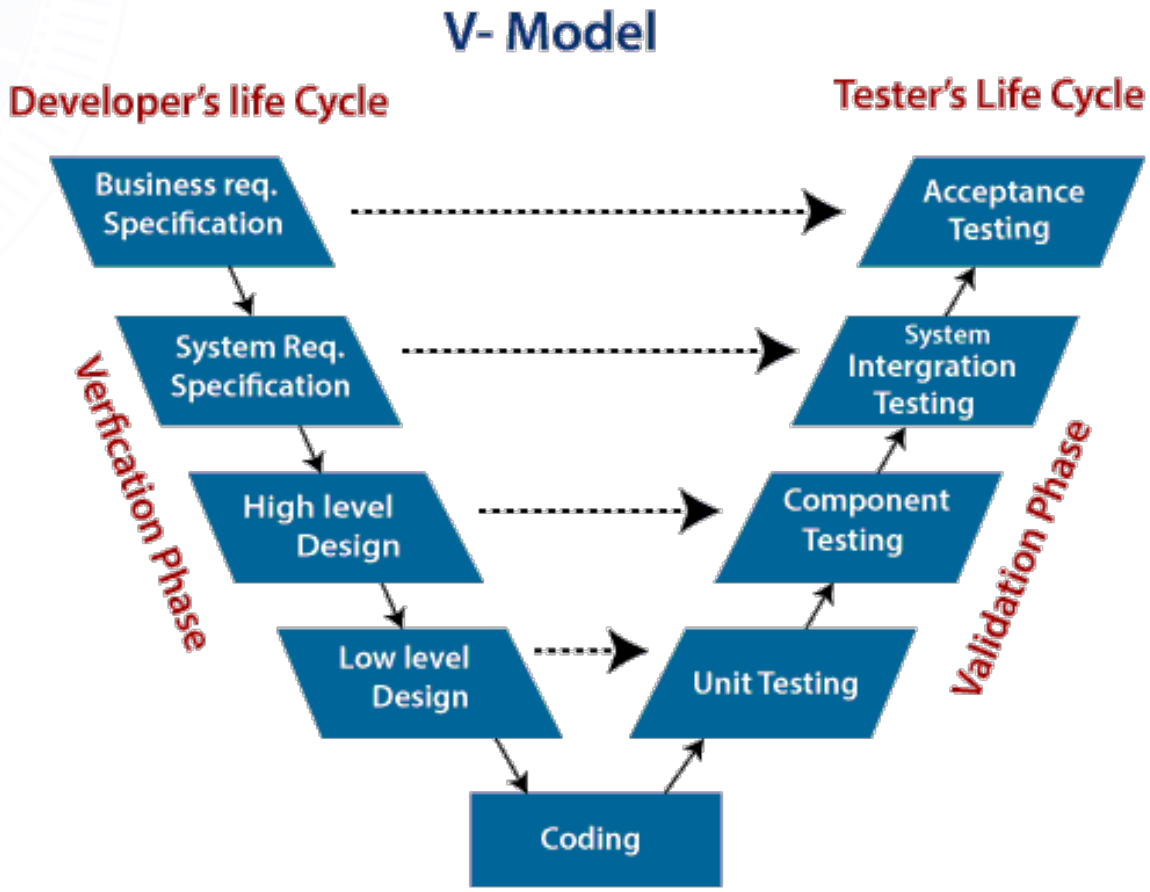
V&V – In More Details

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.

V&V – In Requirements Eng.

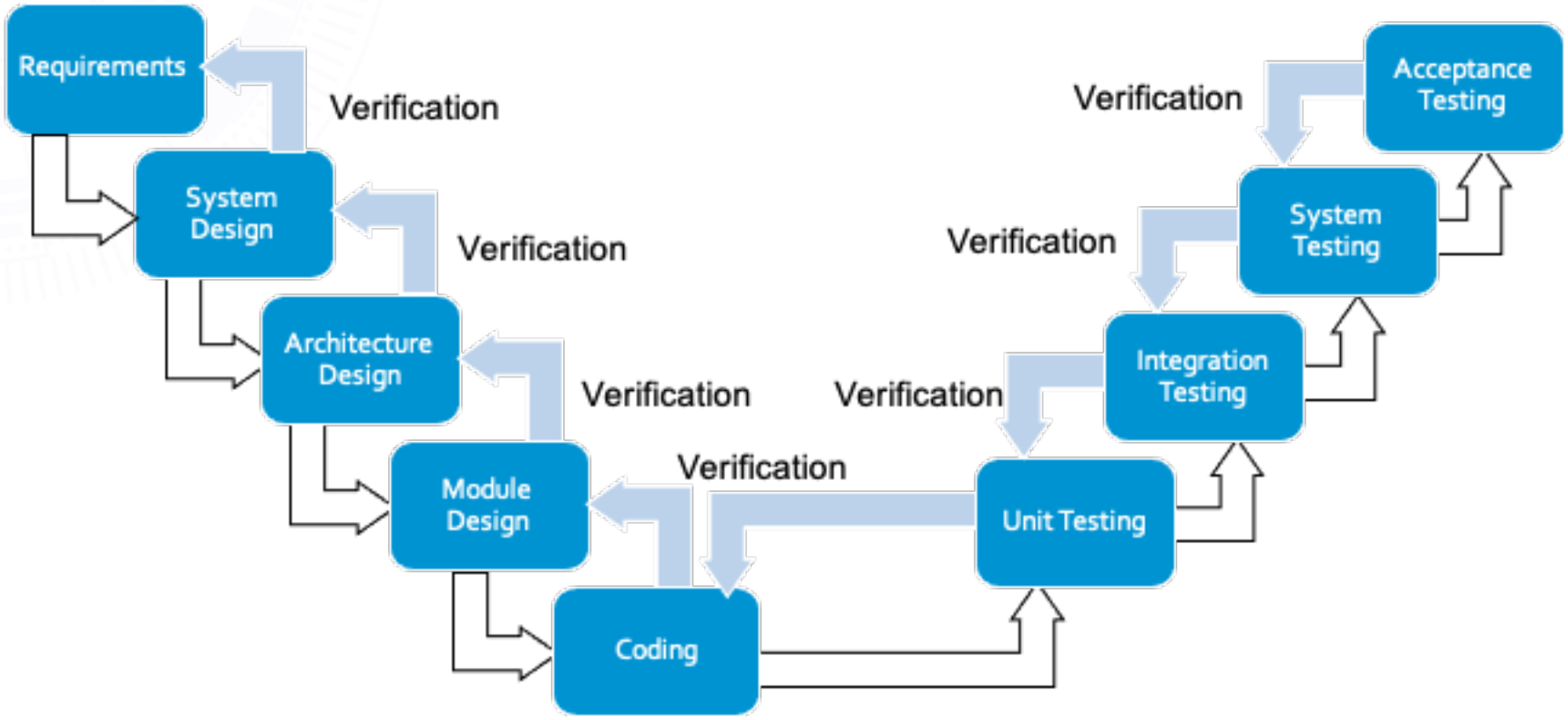


V&V – In Testing (1-3)



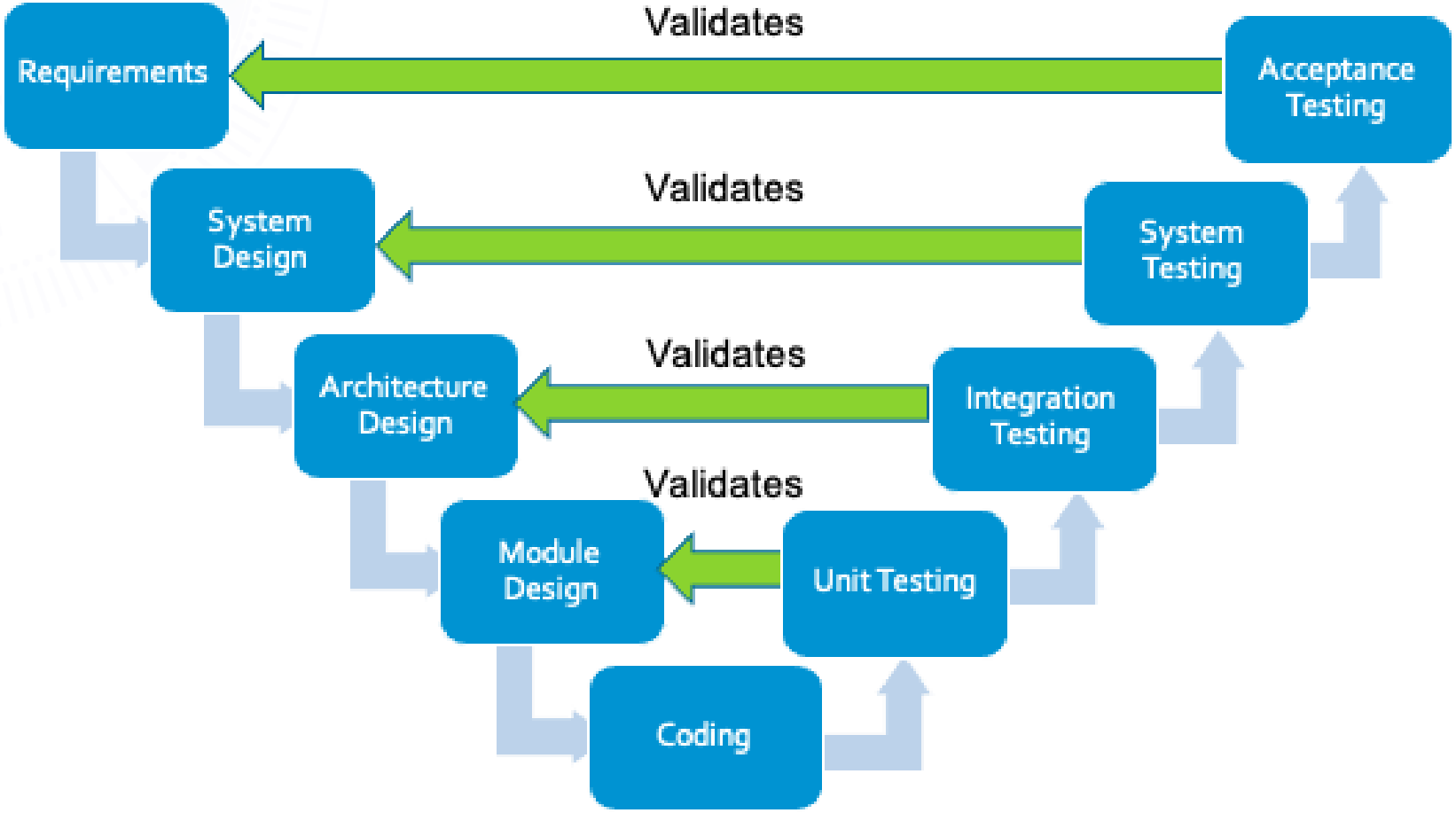
src: https://www.javatpoint.com/software-engineering-v-model

V&V – In Testing (2-3)



src: https://www.parasoft.com/blog/verification-vs-validation-in-embedded-software/

V&V – In Testing (3-3)



src: https://www.parasoft.com/blog/verification-vs-validation-in-embedded-software/

Independent V&V – 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

Final Thoughts...
Why Are you Testing!?

Why Are you Testing!? (1-2)

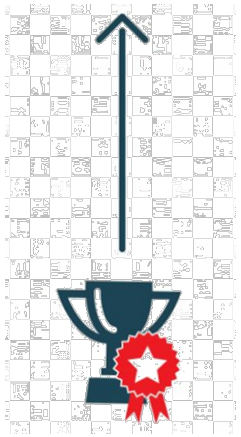
If you ***don't know why*** you're conducting each test, it ***won't be very helpful***

- **Written test objectives** and requirements must be **documented**
- What are your planned **coverage**?
- How much testing is **enough**?
- Common objective – spend the budget & test until the ship-date
  - Sometimes called the "date criterion"

Why Are you Testing!? (2-2)

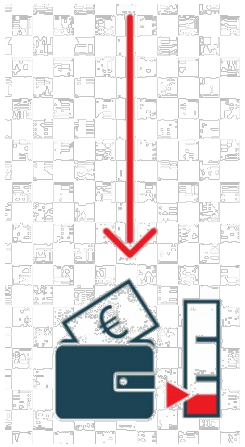
A tester’s goal is to
eliminate faults as early as possible

Improve Quality



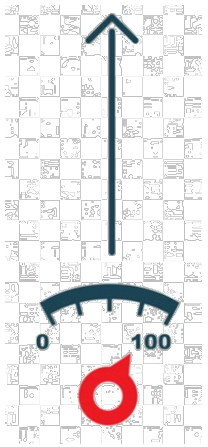
Quality

Reduce Cost



Cost

Preserve Customer Satisfaction



Satisfaction

LERO Software for a better world

A World Leading SFI Research Centre

Science Foundation Ireland sfi For what's next

What Are Your Questions?

HOST INSTITUTION

UNIVERSITY OF LIMERICK OLLSCOIL LUIMNIGH

PARTNER INSTITUTIONS

NUI Galway OÉ Gaillimh · Maynooth University · Trinity College Dublin, Coláiste na Tríonóide, Baile Átha Cliath, The University of Dublin · University College Dublin, An Coláiste Ollscoile, Baile Átha Cliath · UCC · MTU, Ollscoil Teicneolaíochta na Mumhan, Munster Technological University · DCU · Waterford Institute of Technology · Dundalk Institute of Technology, Dún Dealgan · LIT · GMIT