

ITP2200

Introduction to Software Testing

Bogdan Marculescu

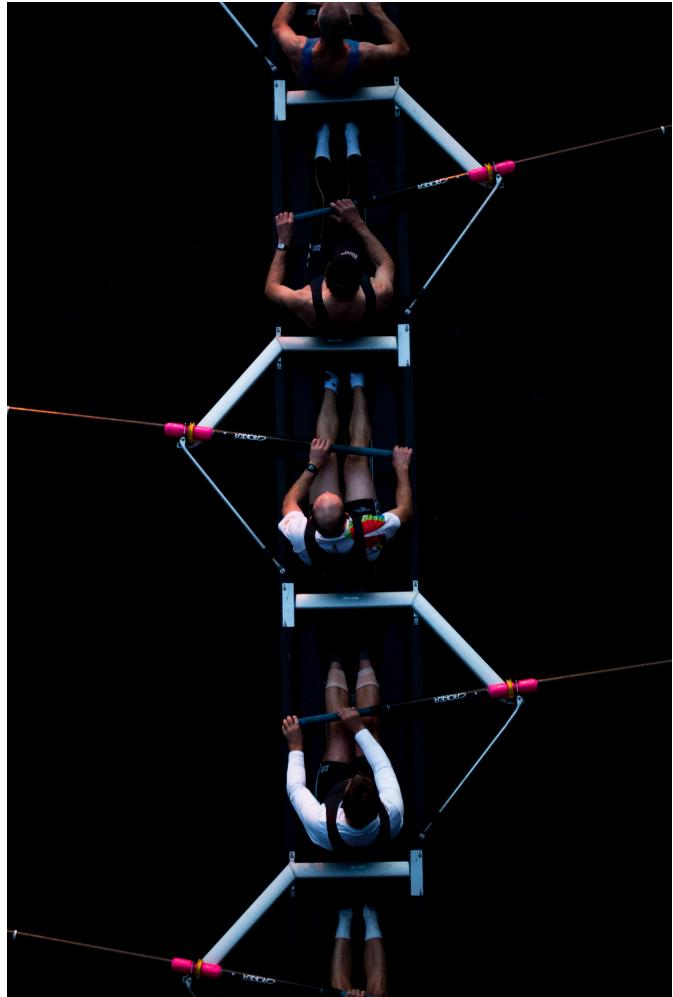
Quick bit of admin

This is my first time giving this course:

- So bear with me
 - Always feel free to **ask questions**
 - Collaboration is encouraged, but **contribute, understand, and be able to motivate**
 - Always read up more than lecture notes (books, stack overflow, blogs, anything interesting – use **initiative**)
 - **Reflect** on examples from other courses or your own experience
 - Exam: Group home exam, more info later



Quick bit of admin



The point of the course:

- You are in an environment where **quality** is increasingly important
- You **WILL** have to test your work
- You will have use the work of others, and make your work usable to them

So you should

- Write good code (and show it's good)
- Understand how to apply testing techniques
- Learn to assess software quality
- Reason about software quality and improve software quality.

Overview of Testing



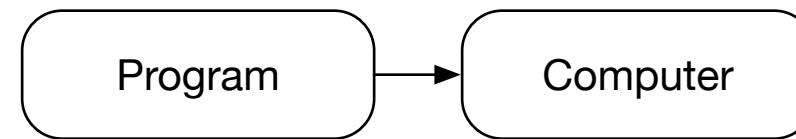
Boom slide:

- What happens when testing is not sufficient
- Financial loss
- Lives endangered
- Smaller bugs also have quite an impact

What does a Test Engineer do?

Test Engineer:

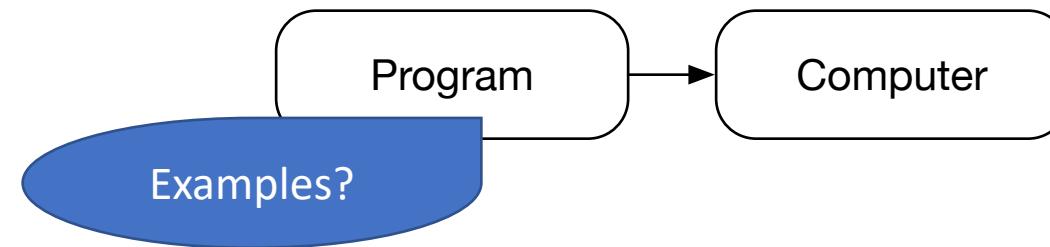
- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



What does a Test Engineer do?

Test Engineer:

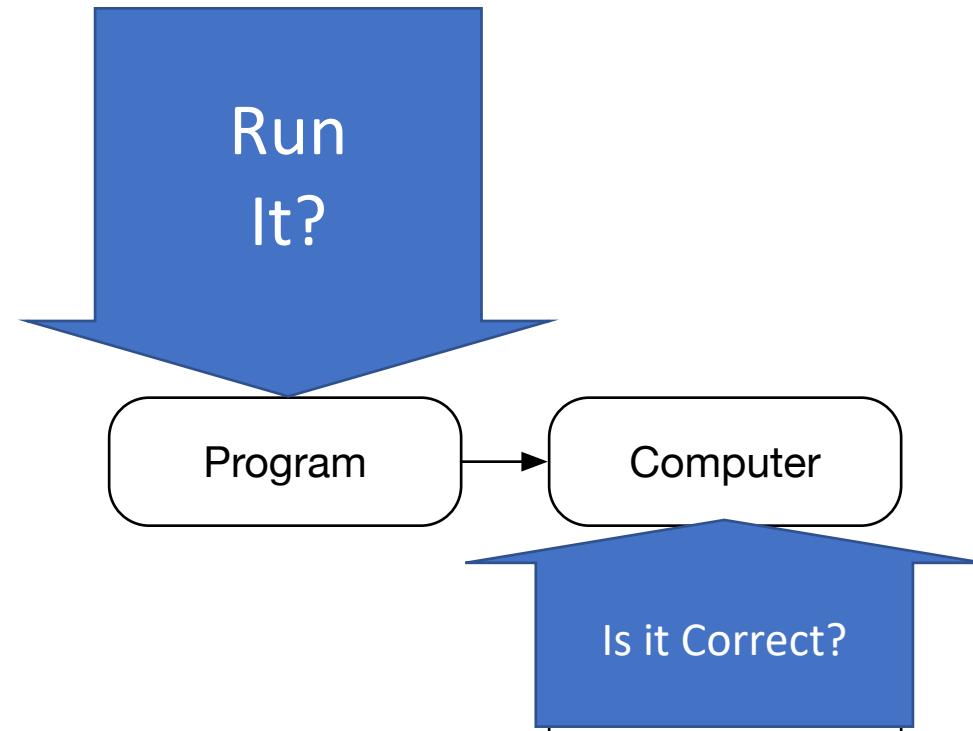
- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



What does a Test Engineer do?

Test Engineer:

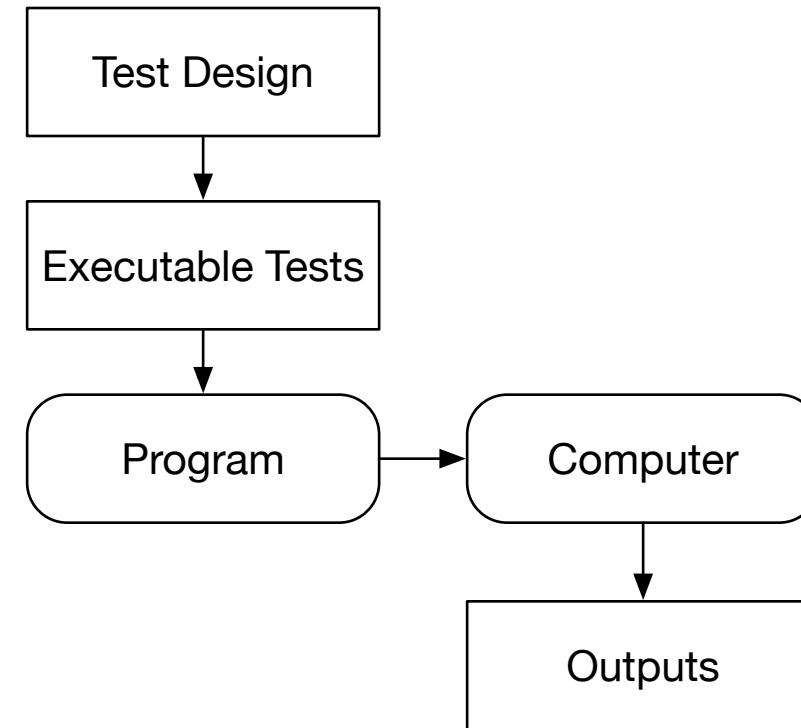
- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



What does a Test Engineer do?

Test Engineer:

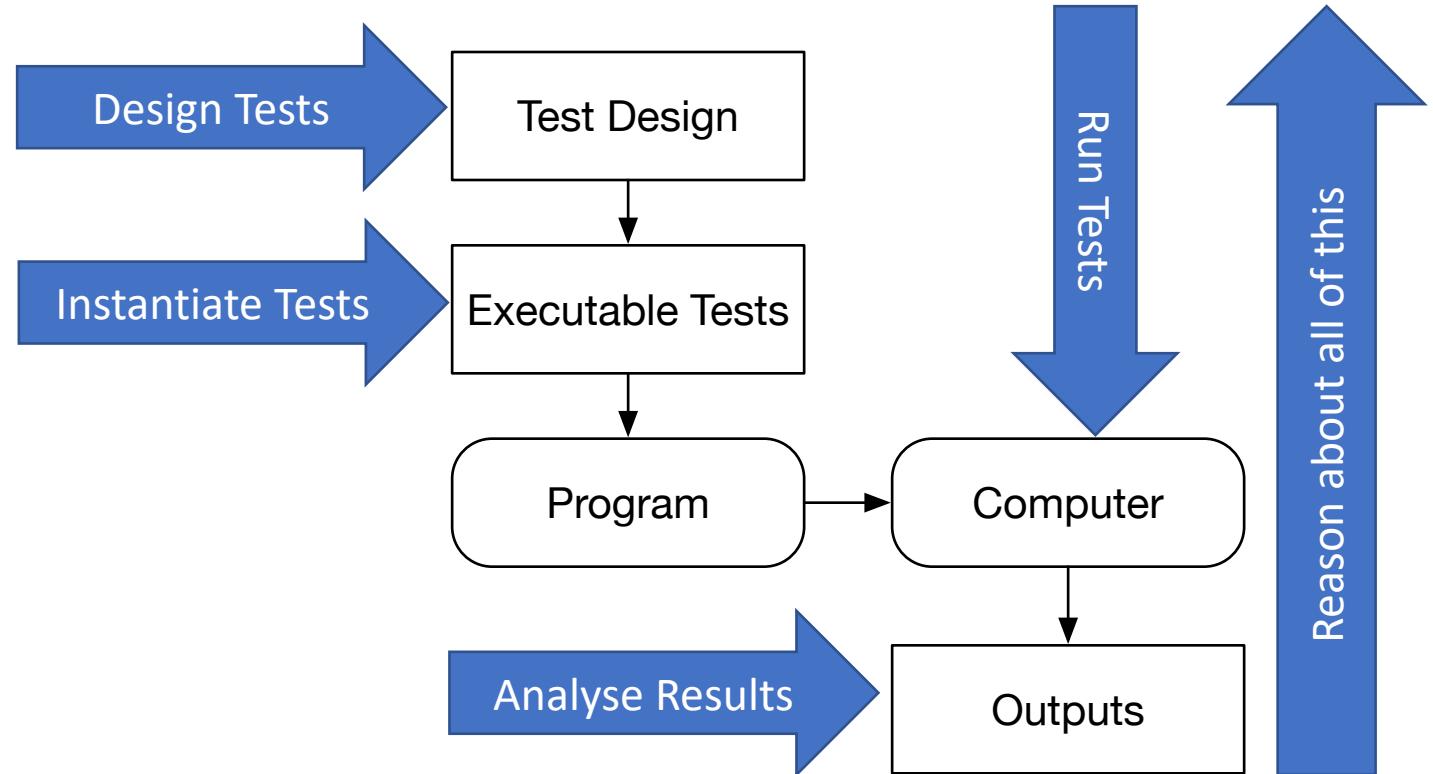
- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



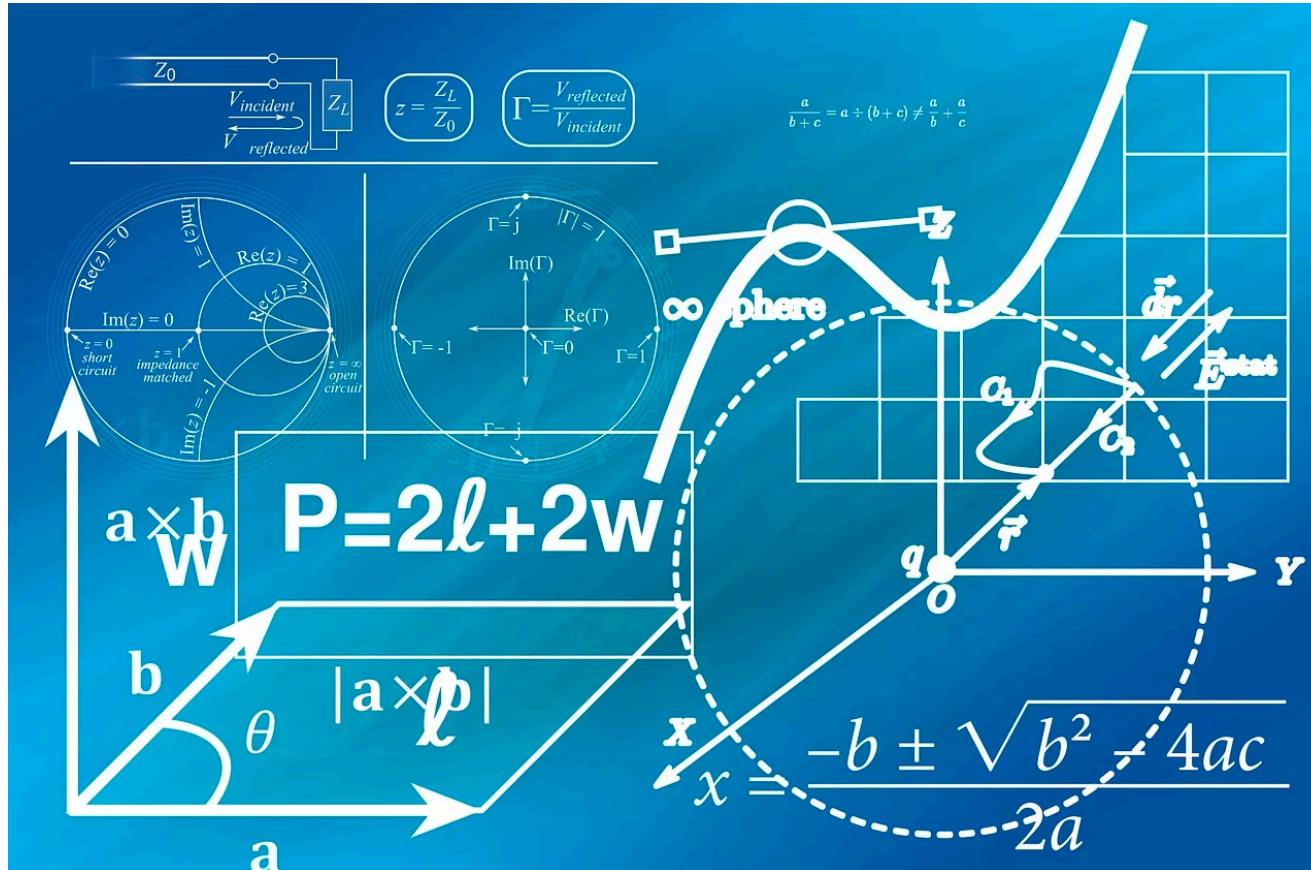
What does a Test Engineer do?

Test Engineer:

- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



Is that all? That's easy.

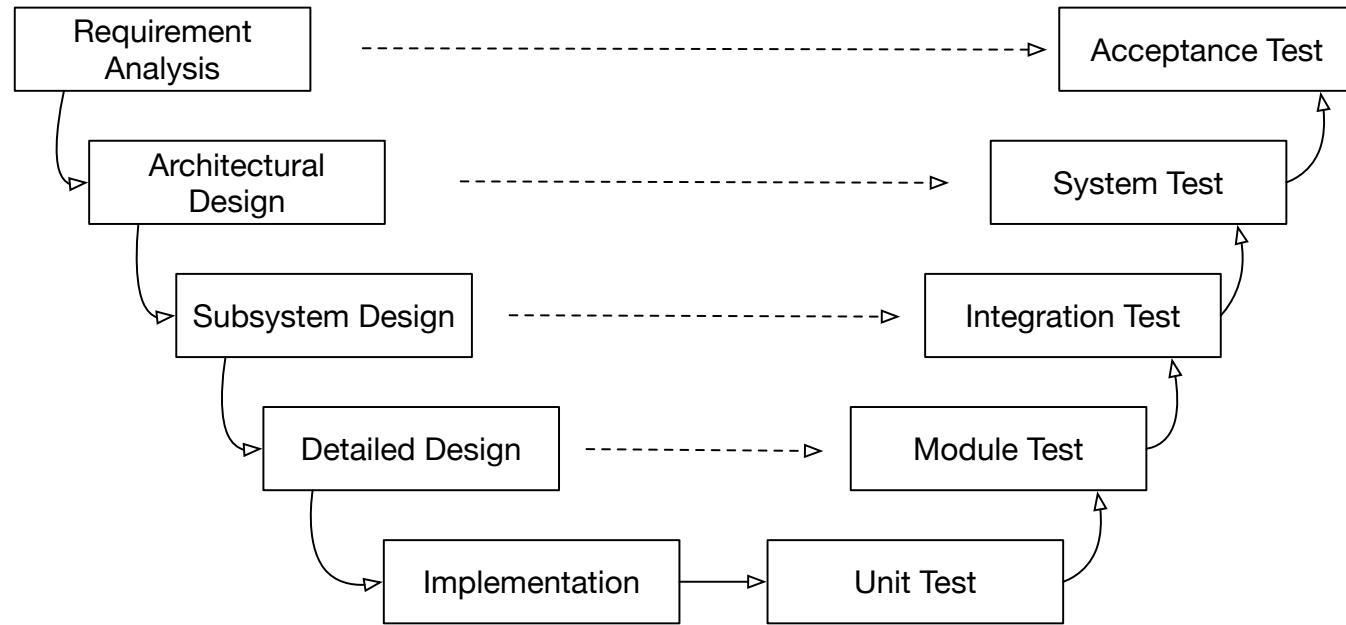


Audience participation question:

- What is the most complex* software you're worked with?

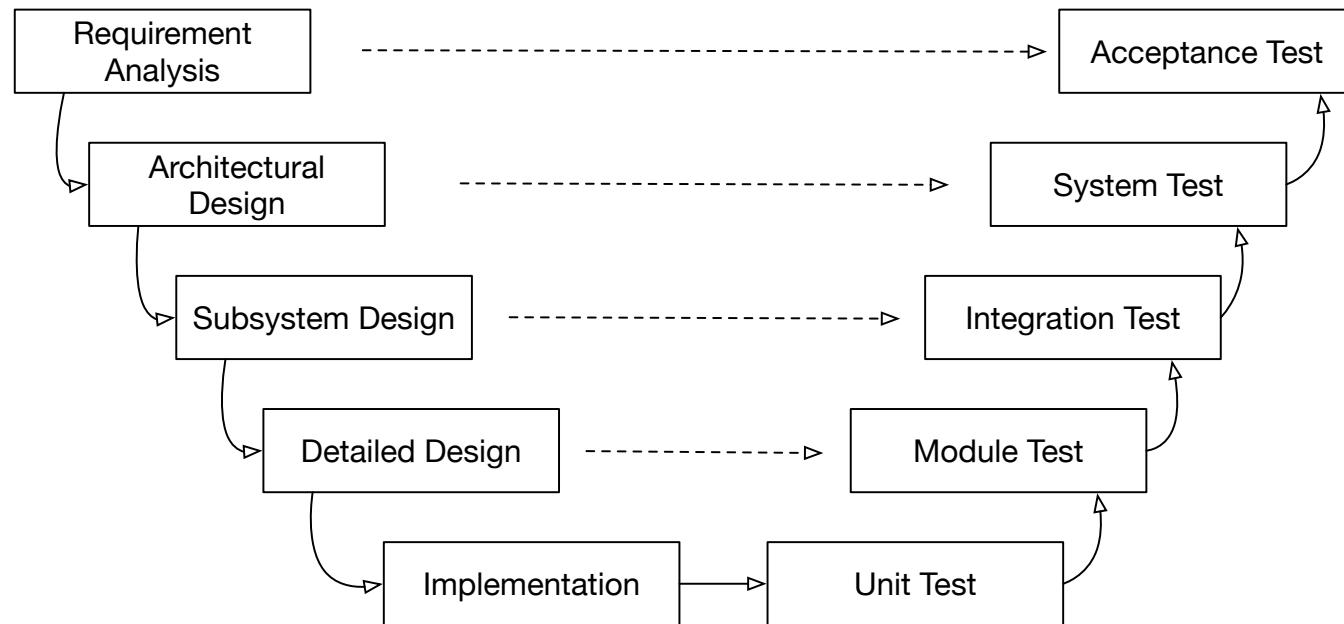
* Complex can be complicated, big, impressive, any definition you like.

Quick intro into the V-model



- Or the one-sentence guide to the entire field of software engineering

V-model: testing from the perspective of software activities

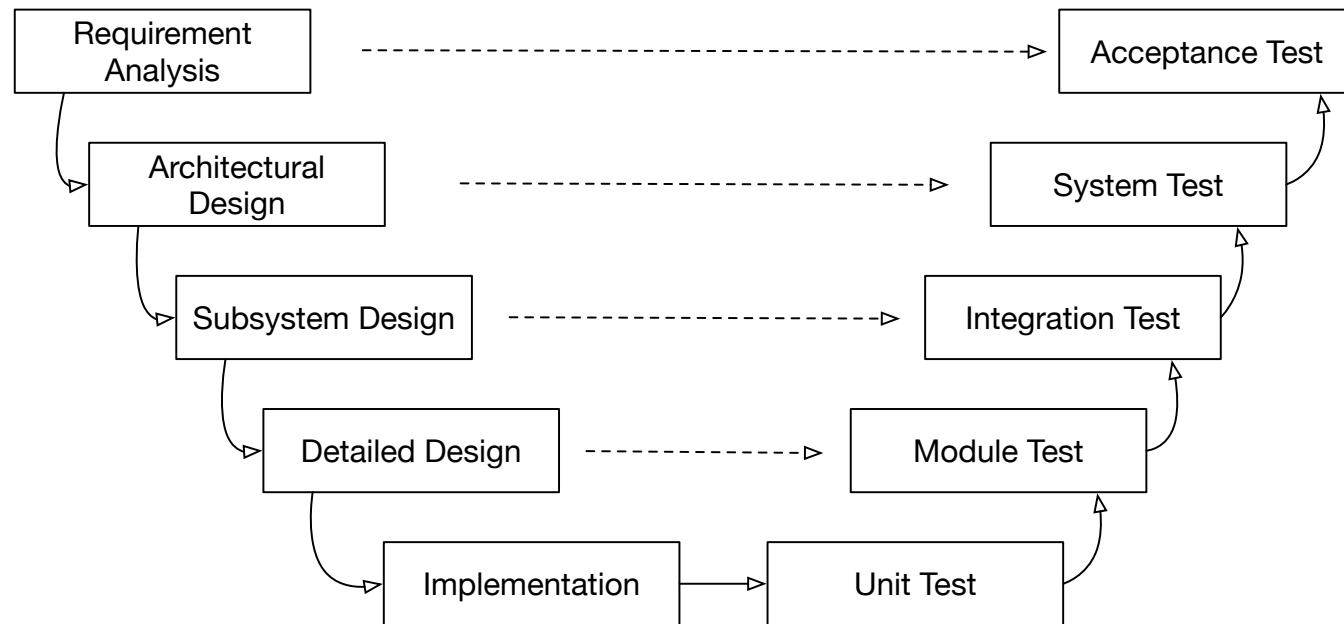


V-model :

Audience participation question:

- Where do the examples fit in this model?

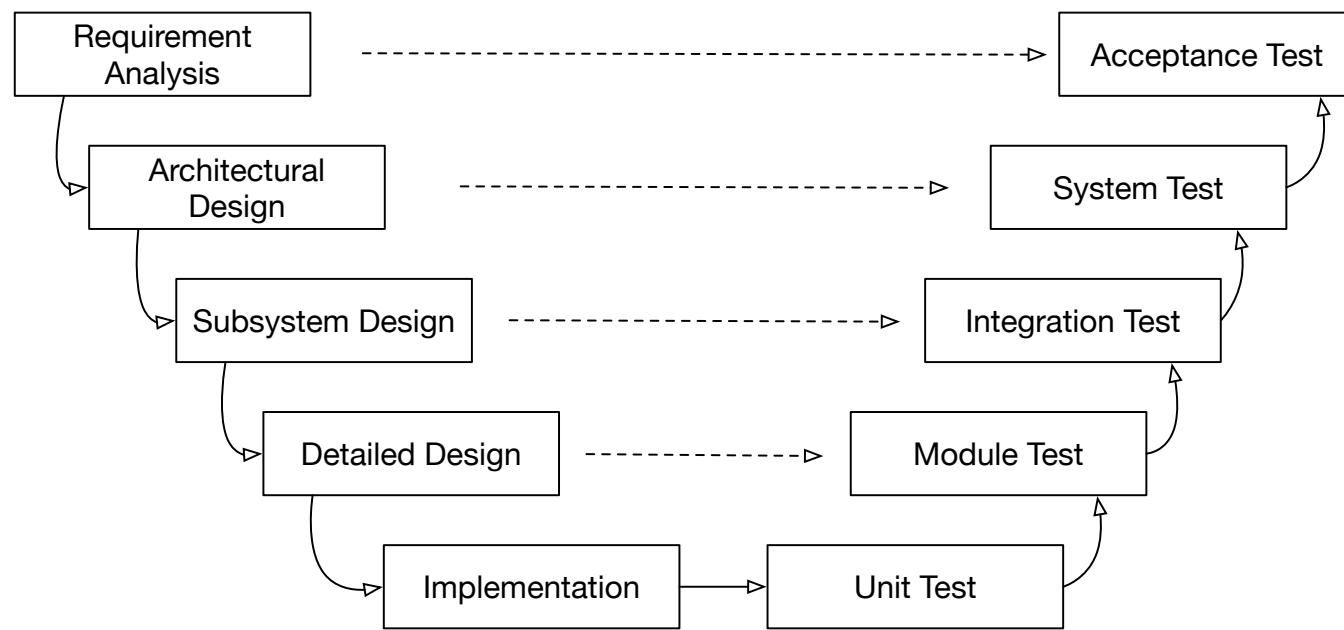
Is that all? That's easy.



V-model:

- Vector sort?
- A particular type of vector and its operations?
- A library of vector tools?
- A set of data structures that also includes vectors?
- A system that also vectors?

V-model: testing from the perspective of software activities



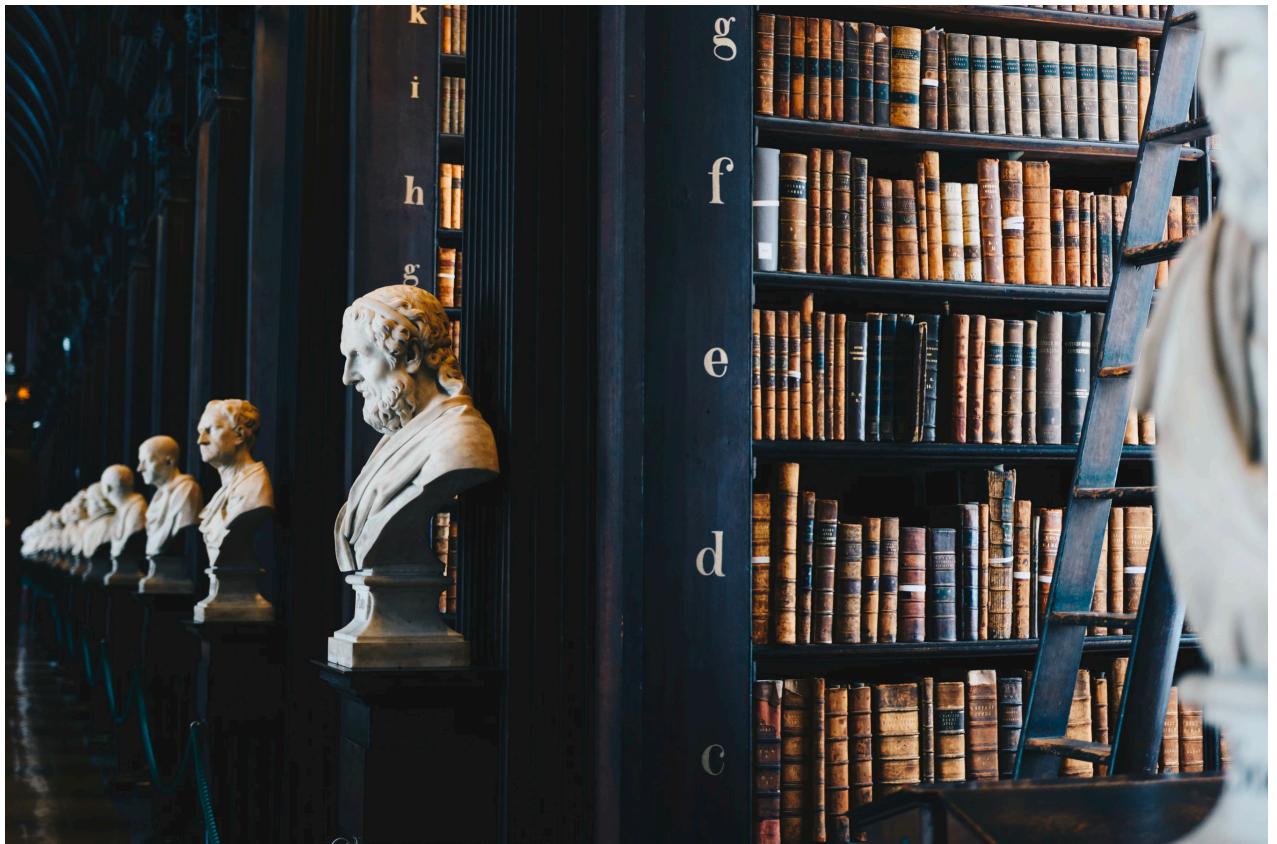
You will revisit this later, **but**

- Remember that **quality** is relevant at all steps
- Different **bugs** are found at each level
- Testing looks at different characteristics at each level

So, How do we think about testing?

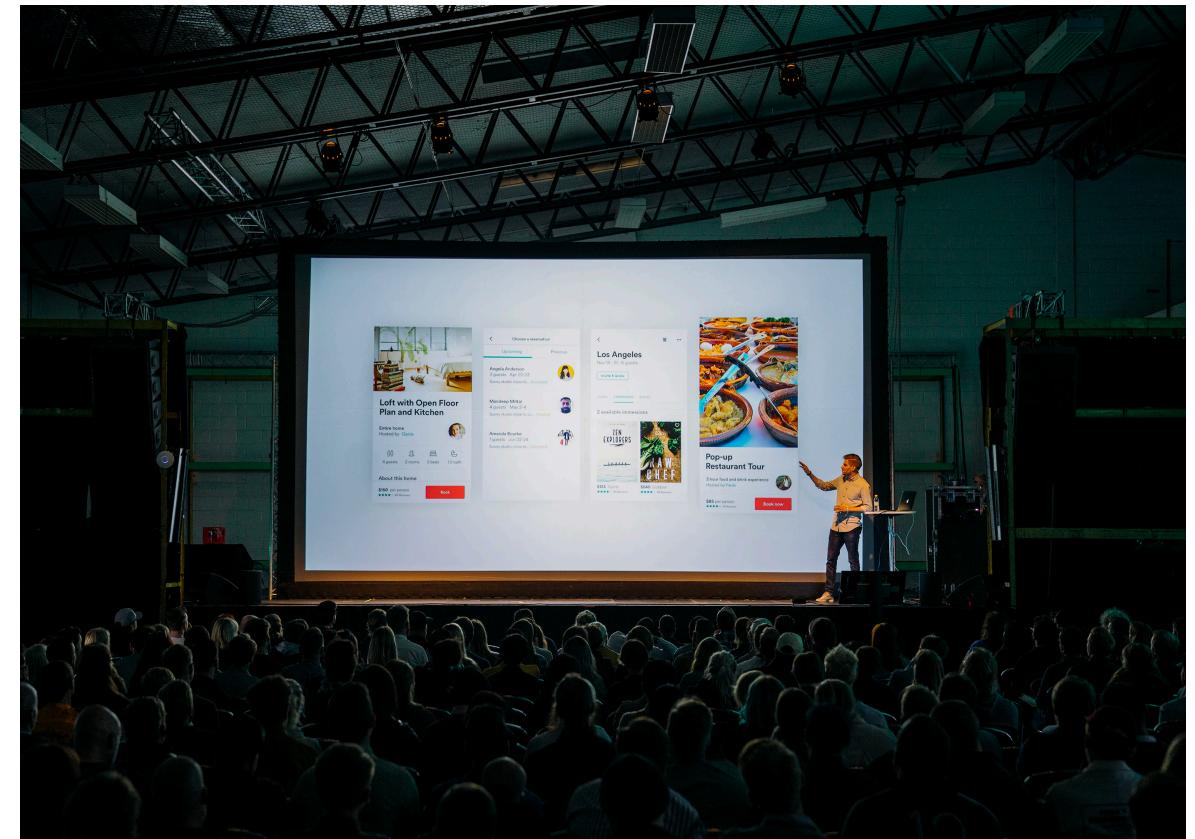
Audience participation
question(s):

- Did you test your code so far?
- How did you do it?
- What would you improve next time?



So, How do we think about testing?

- Fix **bugs** as we find them?
- What if we have a presentation?
We want to show that our **code works**
- What other approaches can you think of?
(Yes, this part **IS** supposed to be interactive)



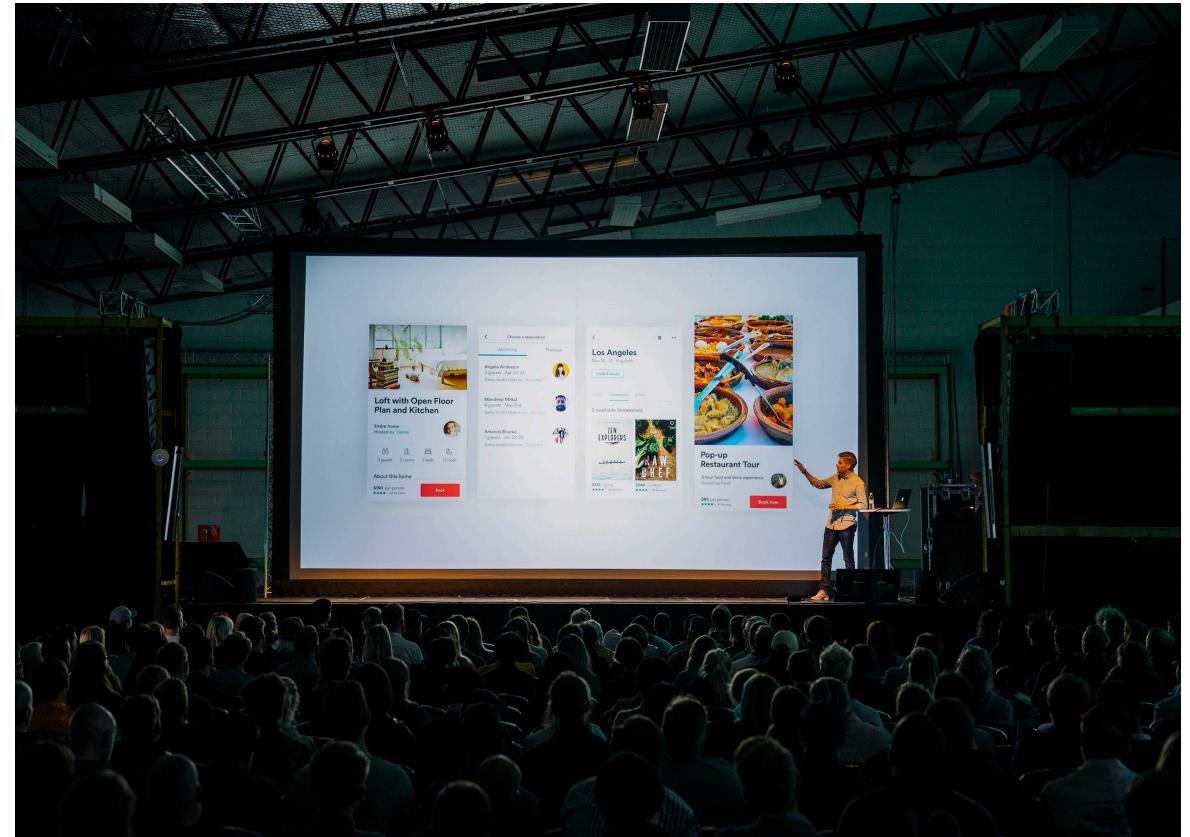
Think back to the original examples (boom slide)

So, How do we think about testing?

- Who wants to take a risk that their project presentation crashes?

(Yes, this part **IS STILL** supposed to be interactive)

Think back to the original examples (boom slide)



Beizer levels



Level 0 - There's no difference between testing and debugging.

Level 1 - The purpose of testing is to show that the software works.

Level 2 - The purpose of testing is to show that the software doesn't work.

Level 3 - The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.

Level 4 - Testing is a mental discipline that helps all IT professionals develop higher quality software.

Beizer levels



More thought-out,
More deliberate,
More long-term

Level 0 - There's no difference between testing and debugging.

Level 1 - The purpose of testing is to show that the software works.

Level 2 - The purpose of testing is to show that the software doesn't work.

Level 3 - The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.

Level 4 - Testing is a mental discipline that helps all IT professionals develop **higher quality software**.



Høyskolen
Kristiania



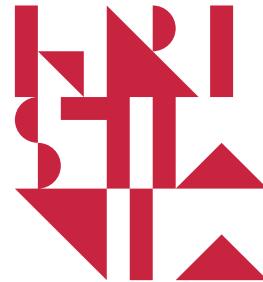
Questions so far?

Quick assignment until next week

Homework:

- Have a quick look at your projects from last semester.
- Try to map all the (very many, I know) definitions to those projects
- Write a quick description and let's discuss them
- Prepare questions about how they can be improved in terms of quality





Høyskolen
Kristiania



Last chance (for now) for questions!