

ITP2200

Introduction to Software Testing

Bogdan Marculescu

Lecture 12

Quick recap and essentials

Motivation



TYPE: E	ENERGY (KeV):	10
L	PRESCRIBED	
00000	0.000000	
00000	200.000000	
70000	0.270000	
00000	0.000000	VERIFIED
00000	359.200000	VERIFIED
00000	14.200000	VERIFIED
00000	27.200000	VERIFIED
1.000000	1.000000	VERIFIED
0.000000	0.000000	VERIFIED
DATE: 2012-04-16 TIME: 11:48:58 OPR ID: 033-tfs3p		
SYSTEM: BEAM READY TREAT: TREAT PAUSE REASON: OPERATOR		
OP.MODE: TREAT X-RAY	AUTO	173777
COMMAND:		



Beizer levels



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

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

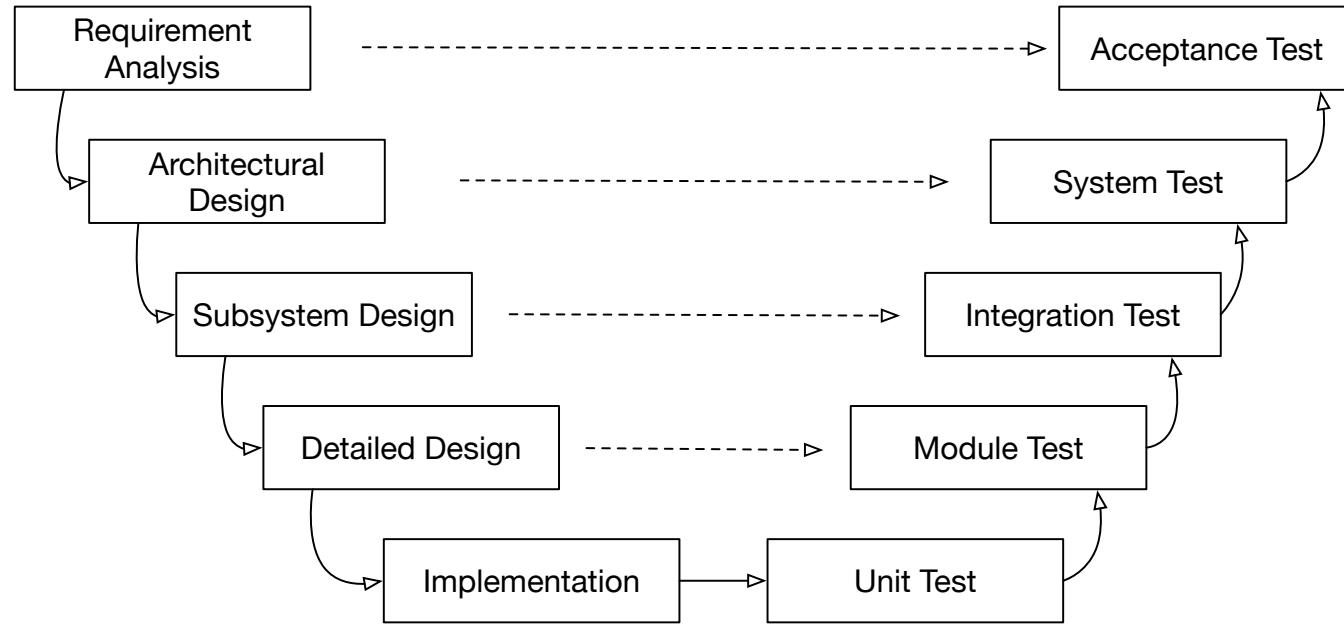
Level 1 - The purpose of testing is to show that the software works.

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**.

Quick intro into the V-model



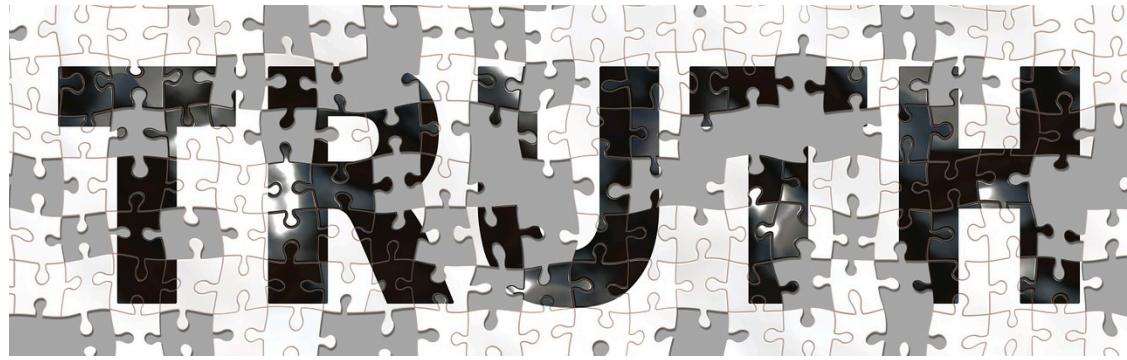
- Or the one-sentence guide to the entire field of software engineering

Requirements and criteria

- Requirement: a condition or capability of a system required by a user to achieve a goal
- How can we assess that a requirement is fulfilled by our system?
- We would like a system that only has acceptable behaviours



Coverage – have we looked everywhere



- Did we evaluate every state our system can be in? Does our system behave as expected in all situations?
- Did we run every method? With every combination of inputs?
- Can an input, or a combination of inputs, induce an undesirable behaviour?
- Are all components of the system of good enough quality?

Oracles – what should it behave like?



- Is the system's behaviour what it should be?
- How do we understand what behaviour the system **should** have?
- What behaviours have we **observed** and can we **replicate**

Tests in software development

- Components should be individually tested – **unit tests**
- Small modules and sub-assemblies need to be good quality – **integration tests**
- The overall product is tested before release – **acceptance tests**
- Re-testing to ensure changes are not introducing/exposing additional bugs – **regression tests**



Tests in software development



- Procedures detailing **what** to test, **when** to test, and **who** is responsible
- What **criteria** to assess against?
- **How** are tests designed? Conducted?
- What **values** to use?
- How to **interpret** results?

Tests in software development

- Changes can occur for a variety of reasons
- Changes usually cascade and cause other changes – modifications that need to be **regression tested**
- Long term software quality – regression tests, test case **selection** and **prioritization, maintainability**





Høyskolen
Kristiania



Questions so far?
