Real-time Software Systems Engineering (2IN70)
Automotive Software Engineering
Autumn 2022 - WEEK3

1 - Software development
2 - Requirements

Mathematics and Computer Science – SET

Contact: dr.ir. Ion Barosan

Location: Metaforum -MF6.091

E-mail: i.barosan@tue.nl

Some slides - Thanks to David Manrique

Content

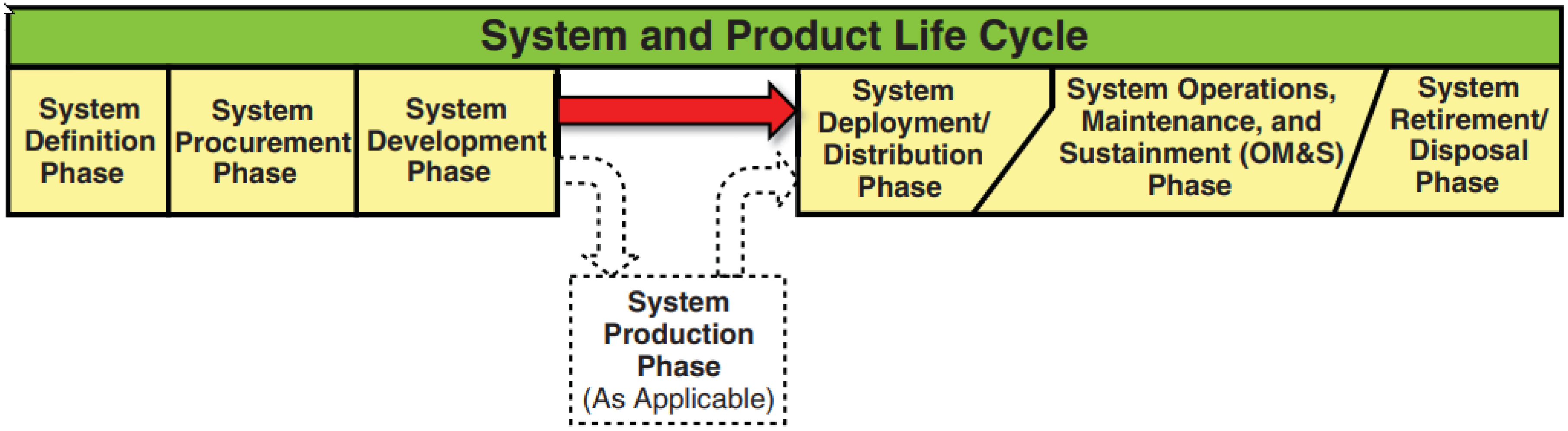
Software Development Process

Requirement gathering

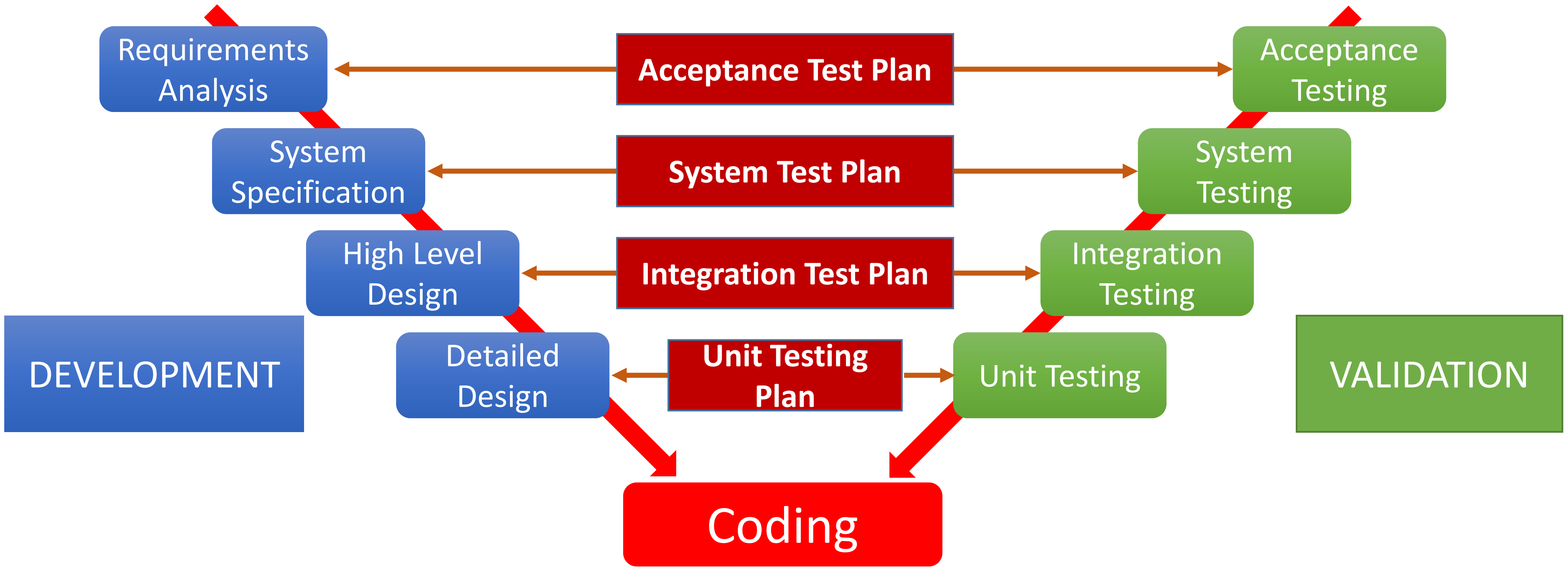
Automotive Software Architecture

Software Development Process

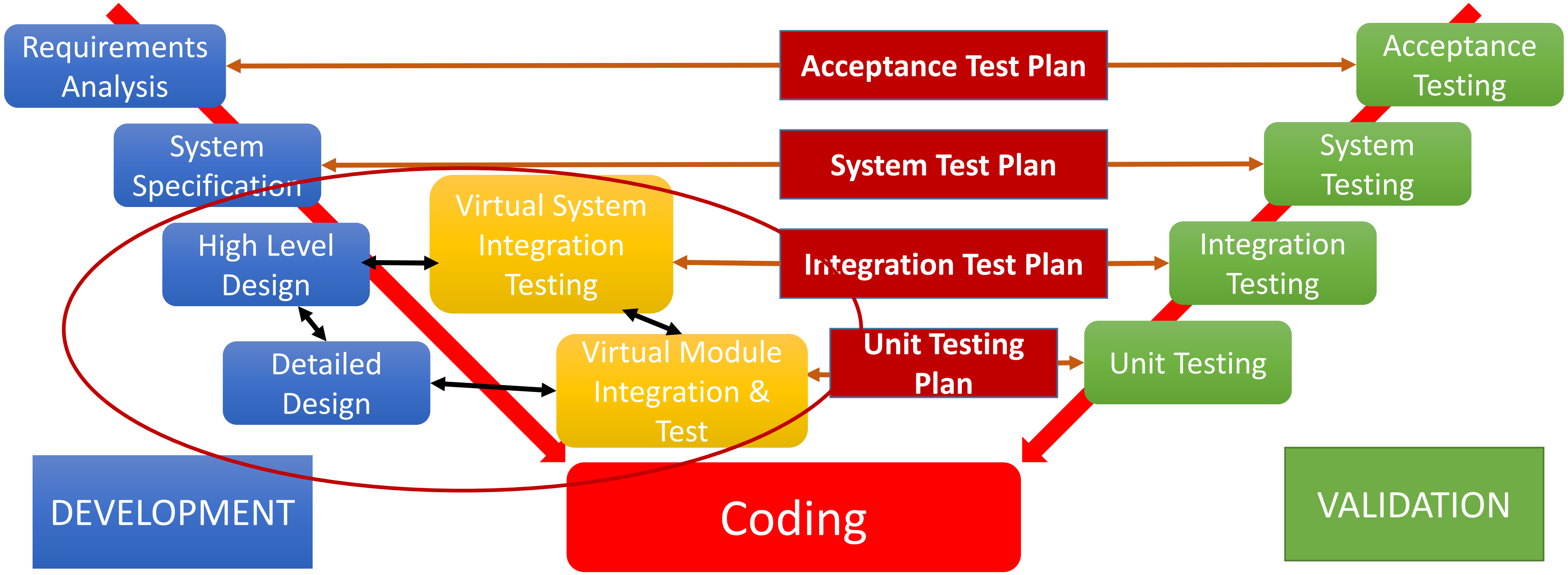
System and Product Life Cycle


System and Product Life Cycle
System Definition Phase
System Procurement Phase
System Development Phase
System Deployment/ Distribution Phase
System Operations, Maintenance, and Sustainment (OM&S) Phase
System Retirement/ Disposal Phase
System Production Phase (As Applicable)

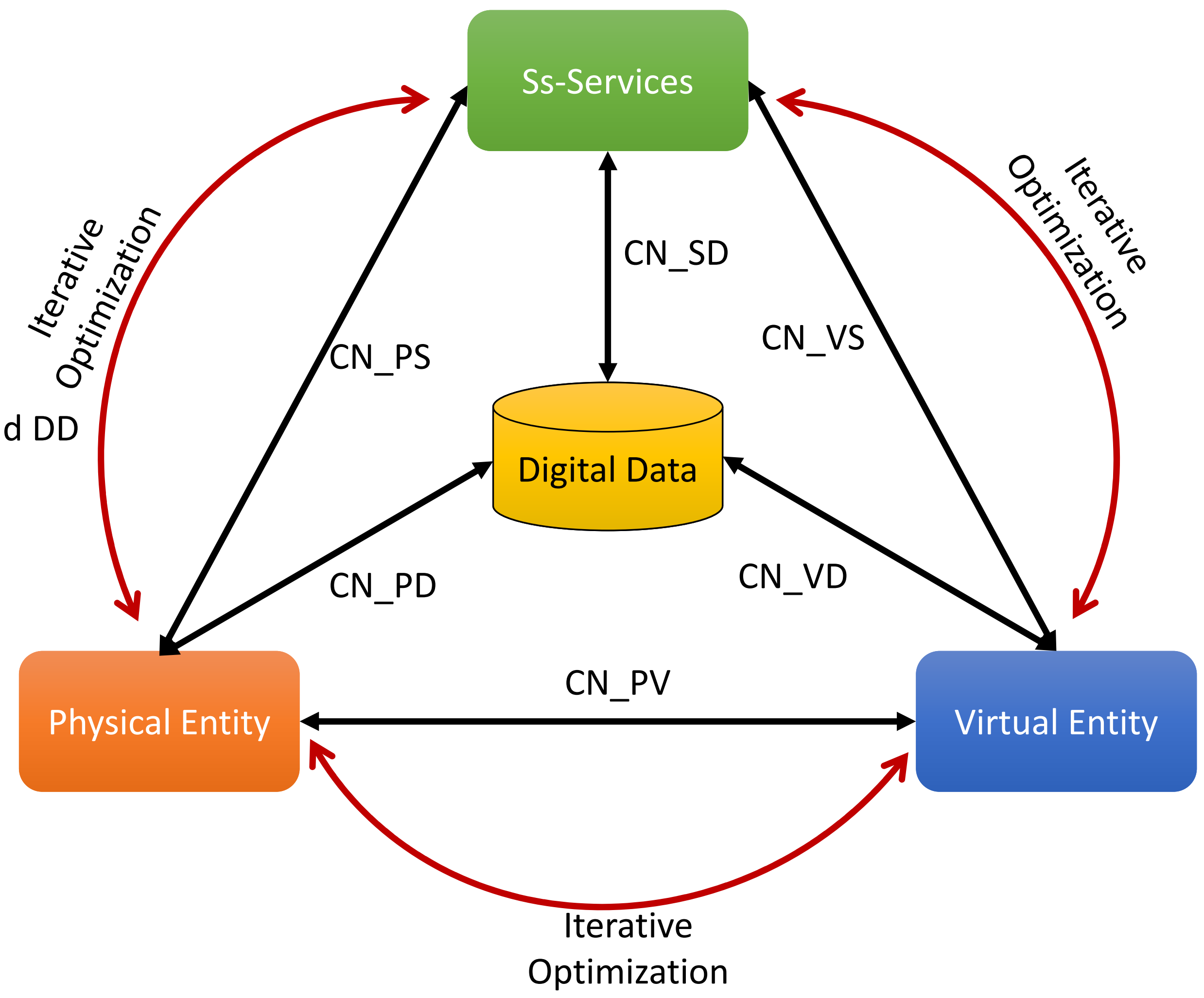
Systems Engineering Process – V Process



Virtual Systems Engineering Process – V Process



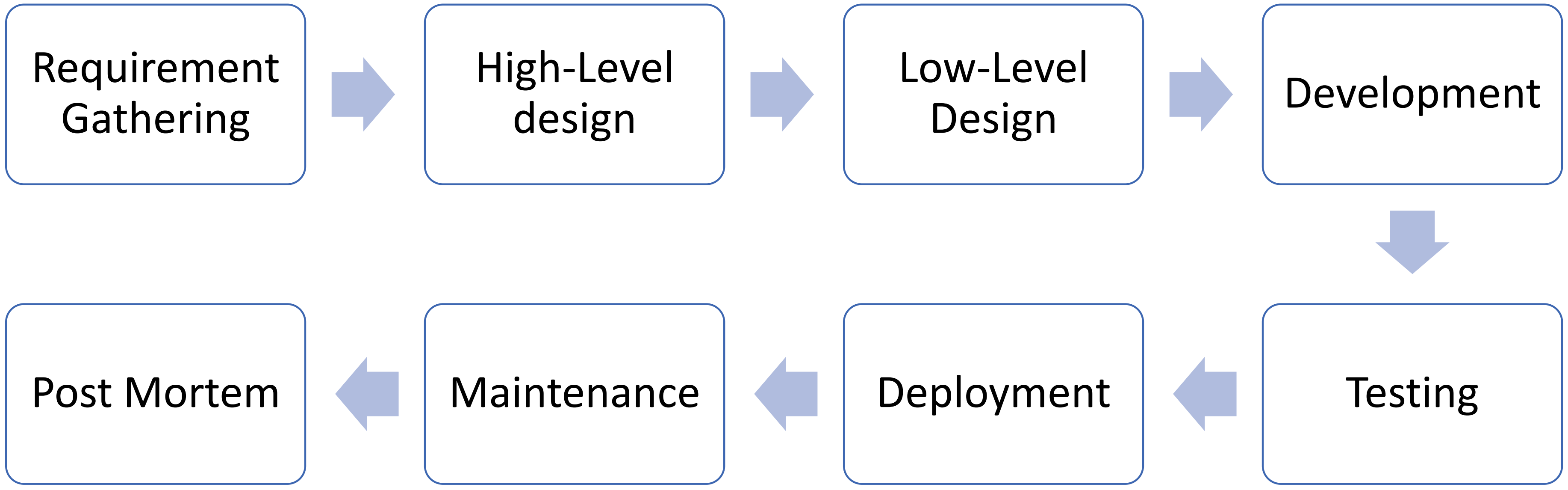
Virtual MDSE Digital Twins



CN_SD: Connection between Services (Ss) and DD
CN_PD: Connection between PE and DD
CN_VD: Connection between VE and DD
CN_PS: Connection PE – Services Data
CN_VS: Connection VE – Services Data
CN_PV: Connection PE and VE

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Software Development Process



Requirement Gathering

Identify Costumer

Identify

- What costumer wants
- What costumer needs

Document requirements

High-Level design

Platform → Data design → Project Architecture

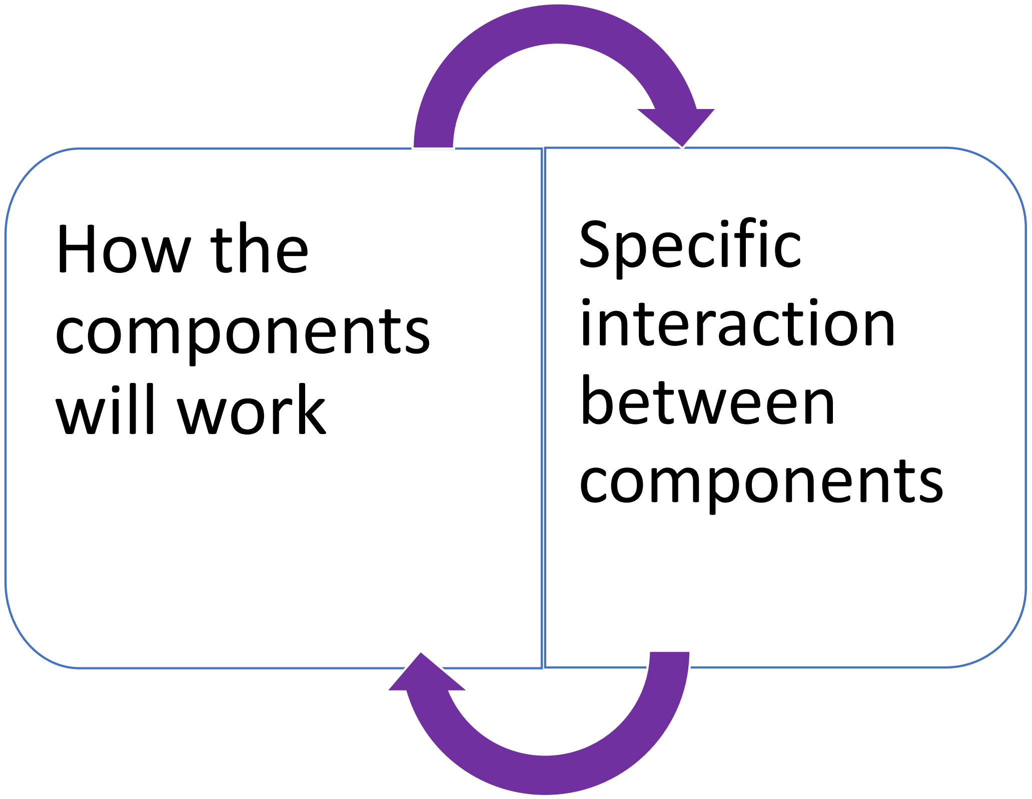
Example

- Data base
- Classes
- User interfaces
- External interfaces

Specify: how **components interacts** and what they **do**

No details in how they do it

Low-Level design


How the components will work
Specific interaction between components

Development

- Implementation phase: code generation and bug detection

Testing (1 of 2)

Bugs in code

Interaction with other components

”Testing shows the presence, not the absence of bugs” Edsger W. Dijkstra

- Programmers test their own code
- Testers test the code
- Test the system integrating piece by piece of components

Every change needs to be retested

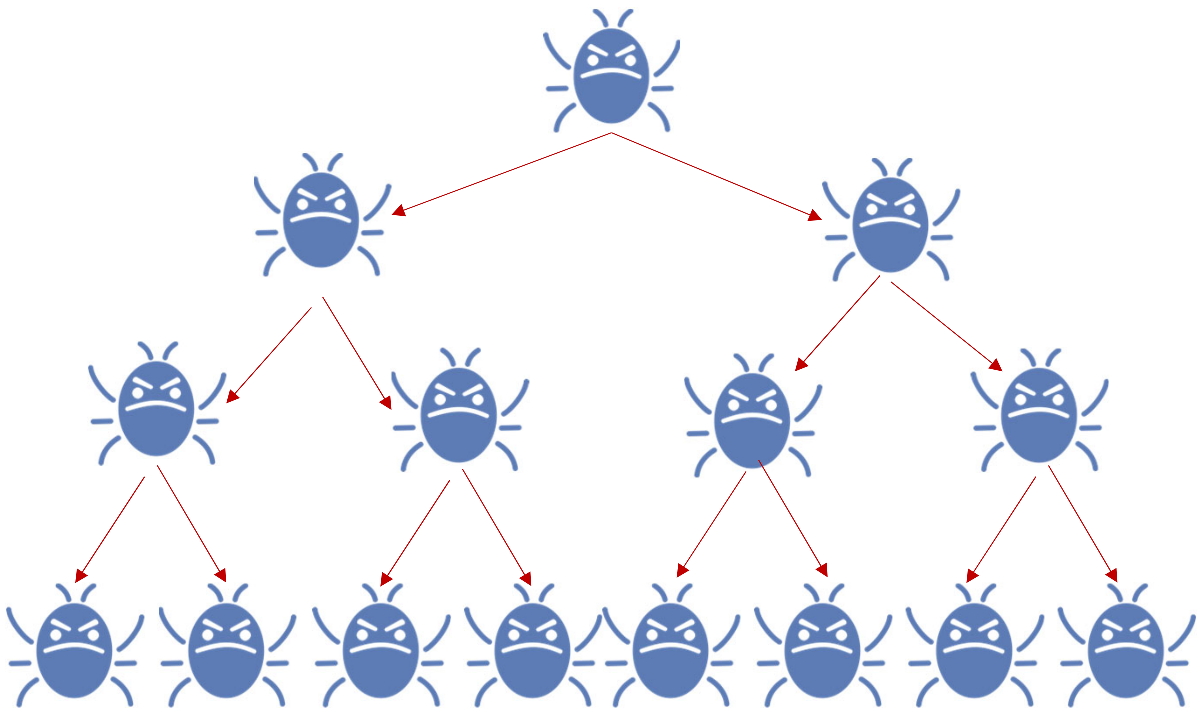
Testing – Counting Bugs (2 of 2)

Requirements

High Level design

Low Level design

Development

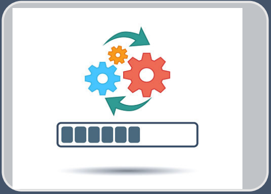


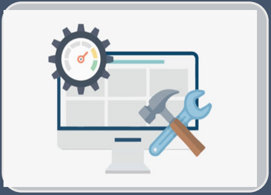
“ the longer a bug remains undetected, the harder it is to fix”.

Deployment

- New Hardware
- New network
- Training
- Onsite support
- Parallel operation with old version
- Data maintenance of old and new version
- Bug fixing

Maintenance


The modification of a software product after delivery to correct faults, to improve performance or other attributes.


80% of maintenance effort is used for non-corrective actions – functionality enhancement to the system

Bug tracking and fixing

Post Mortem

- Evaluation of the project
  - What went well
  - What went wrong
  - Improve the process
  - Problems prevention