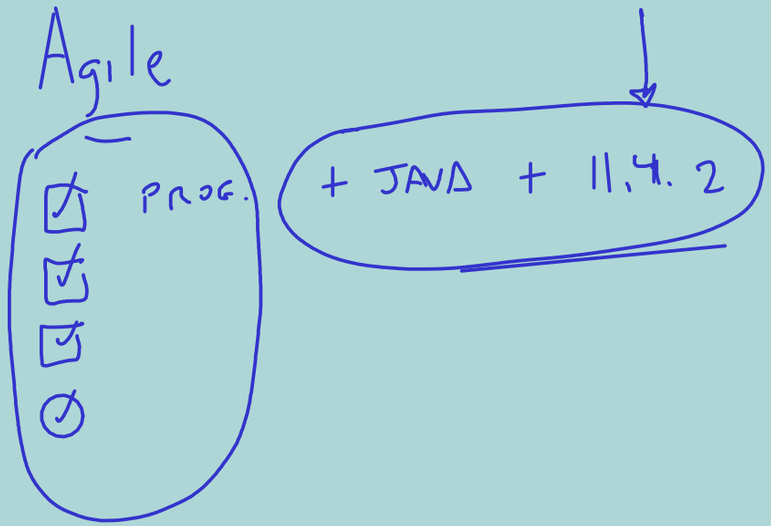
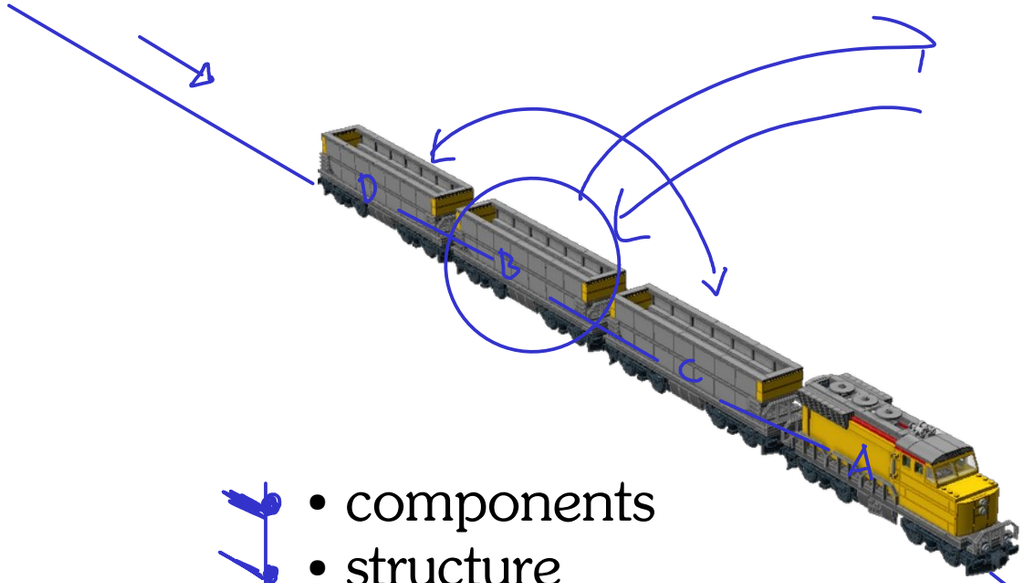
# Plan for Today

- Software engineering intro
  - **identify** and **understand** cause of problems
  - **propose** and **implement** mitigating solutions
  - **measure** effectiveness
  - **refine** and **repeat**
- Development models

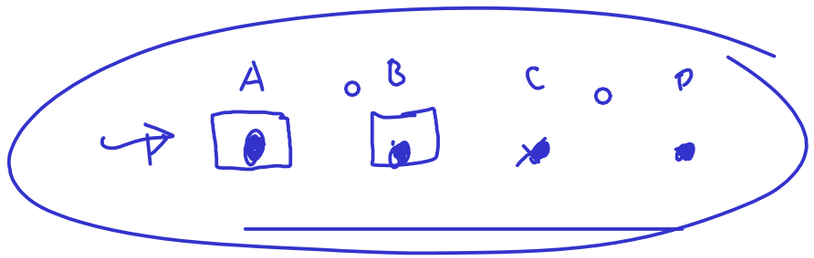


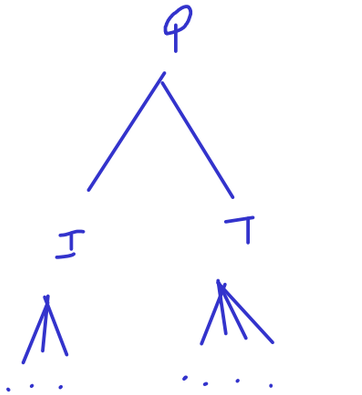

Lecture 3 – 25 September

# Software Development

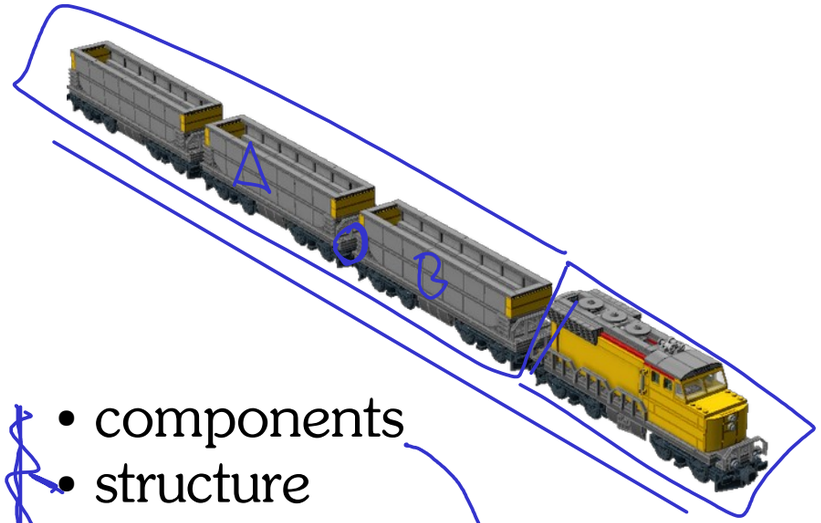



- components
- structure
- action



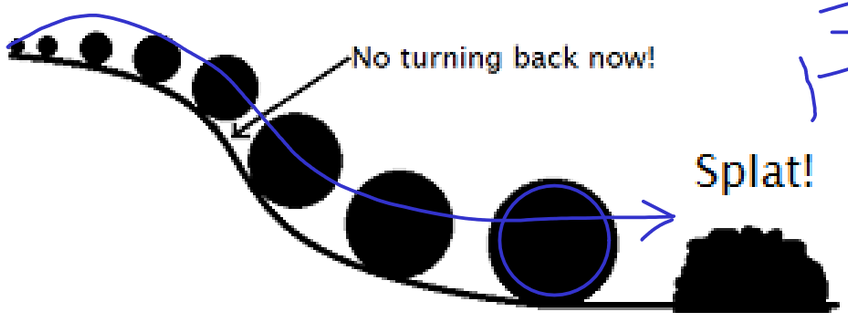



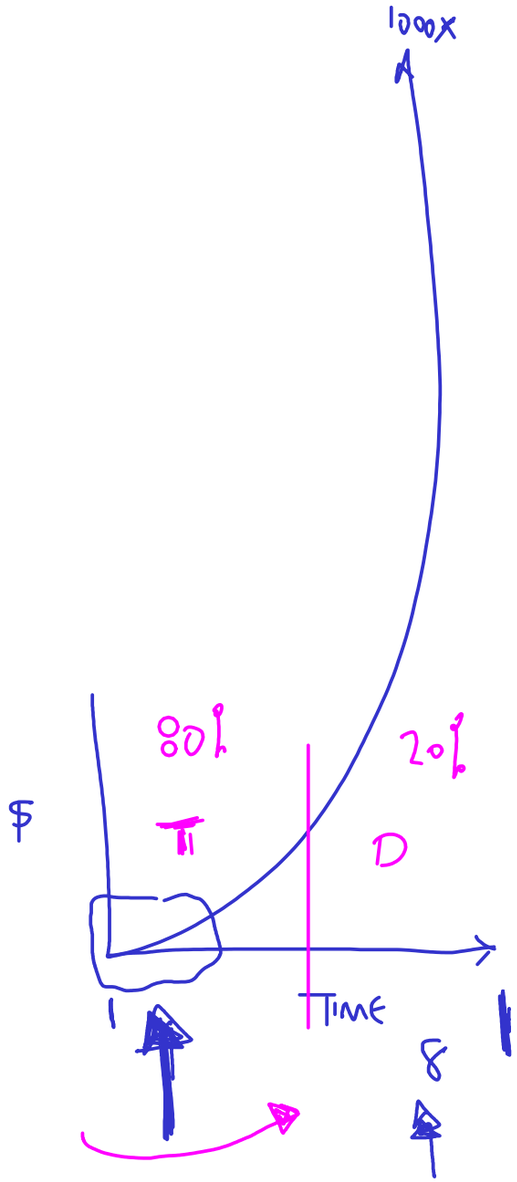
# Software Development




- components
- structure
- action

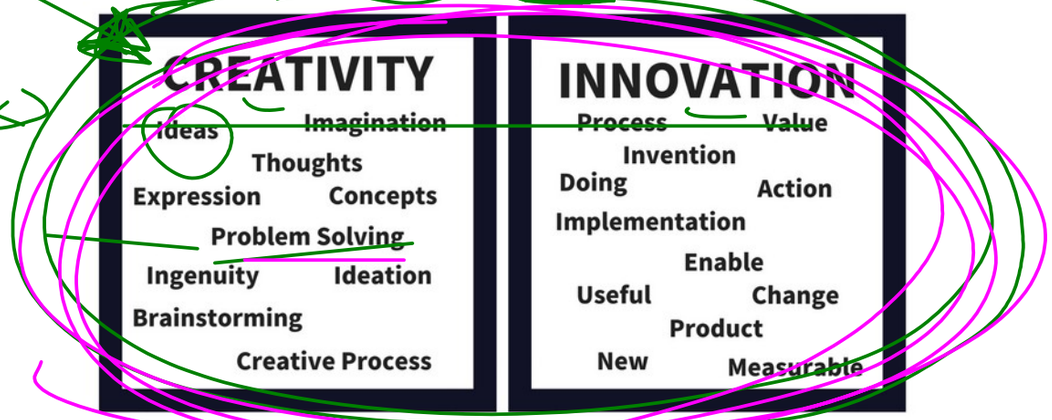
CLASS → A B

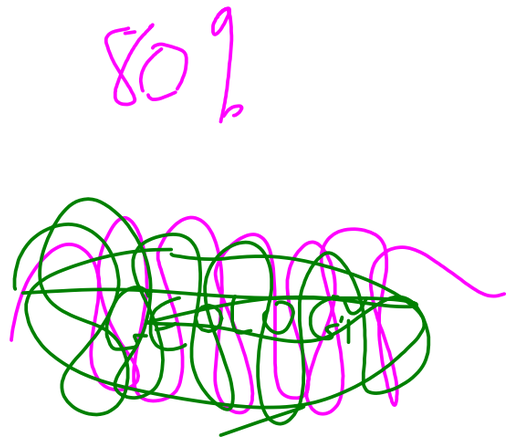


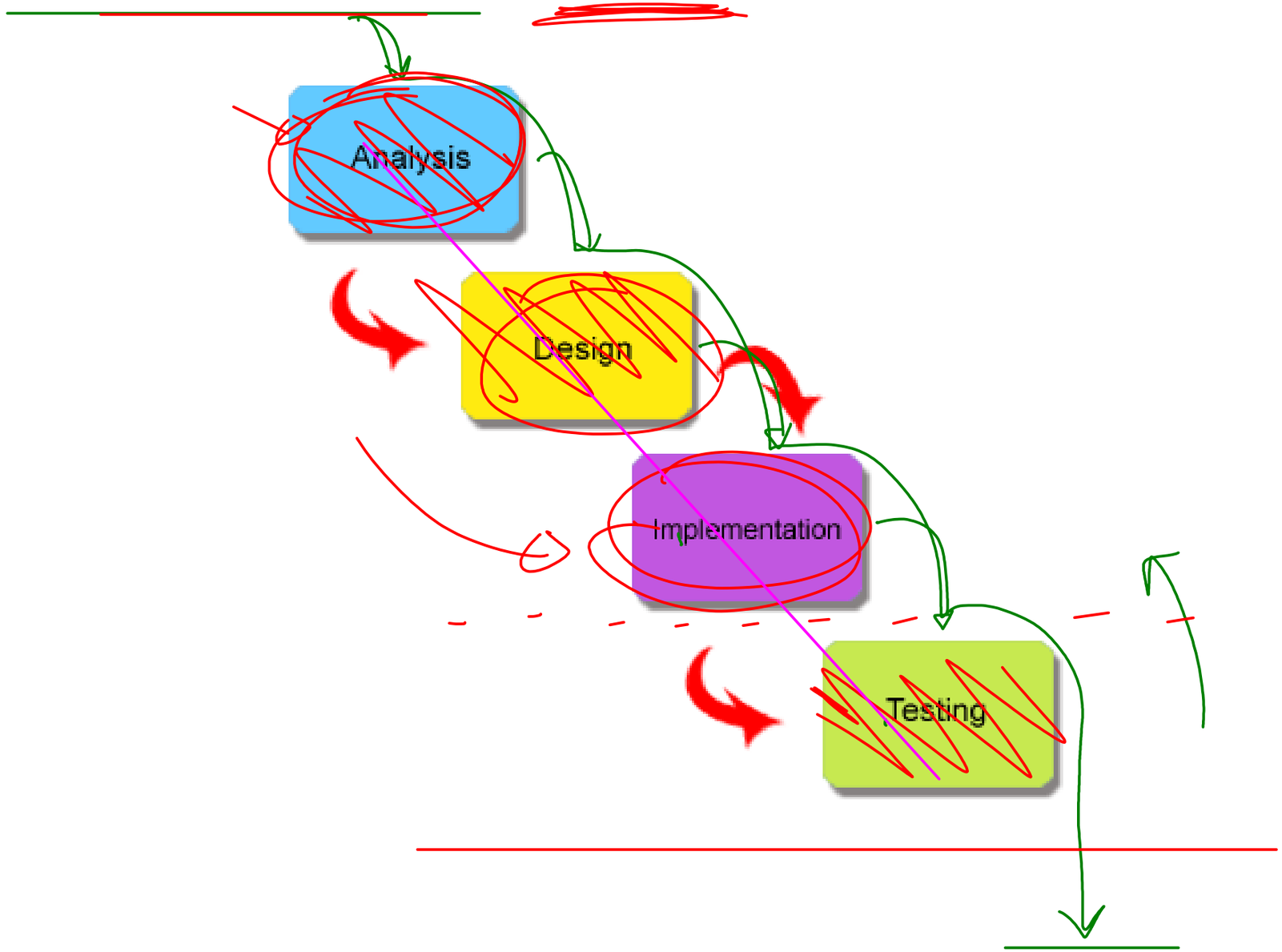
- Software engineering intro
  - **identify** and **understand** cause of problems
  - **propose** and **implement** mitigating solutions
  - **measure** effectiveness
  - **refine** and **repeat**

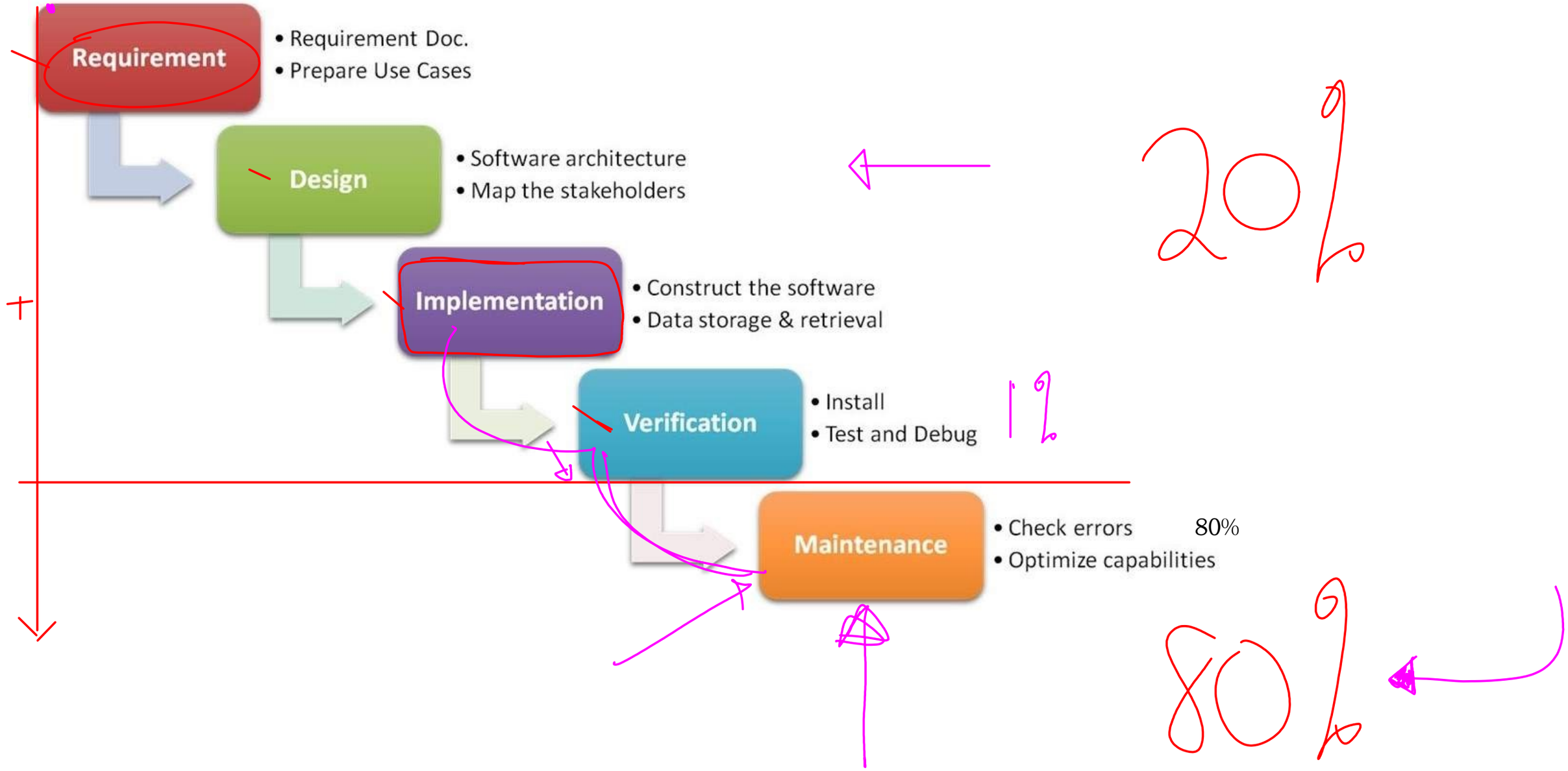
- Our reality
  - software engineering: we don't understand engineering
  - computer science: we don't understand science
  - field that builds:
    - we don't know how to build
    - we don't know how things are built or why

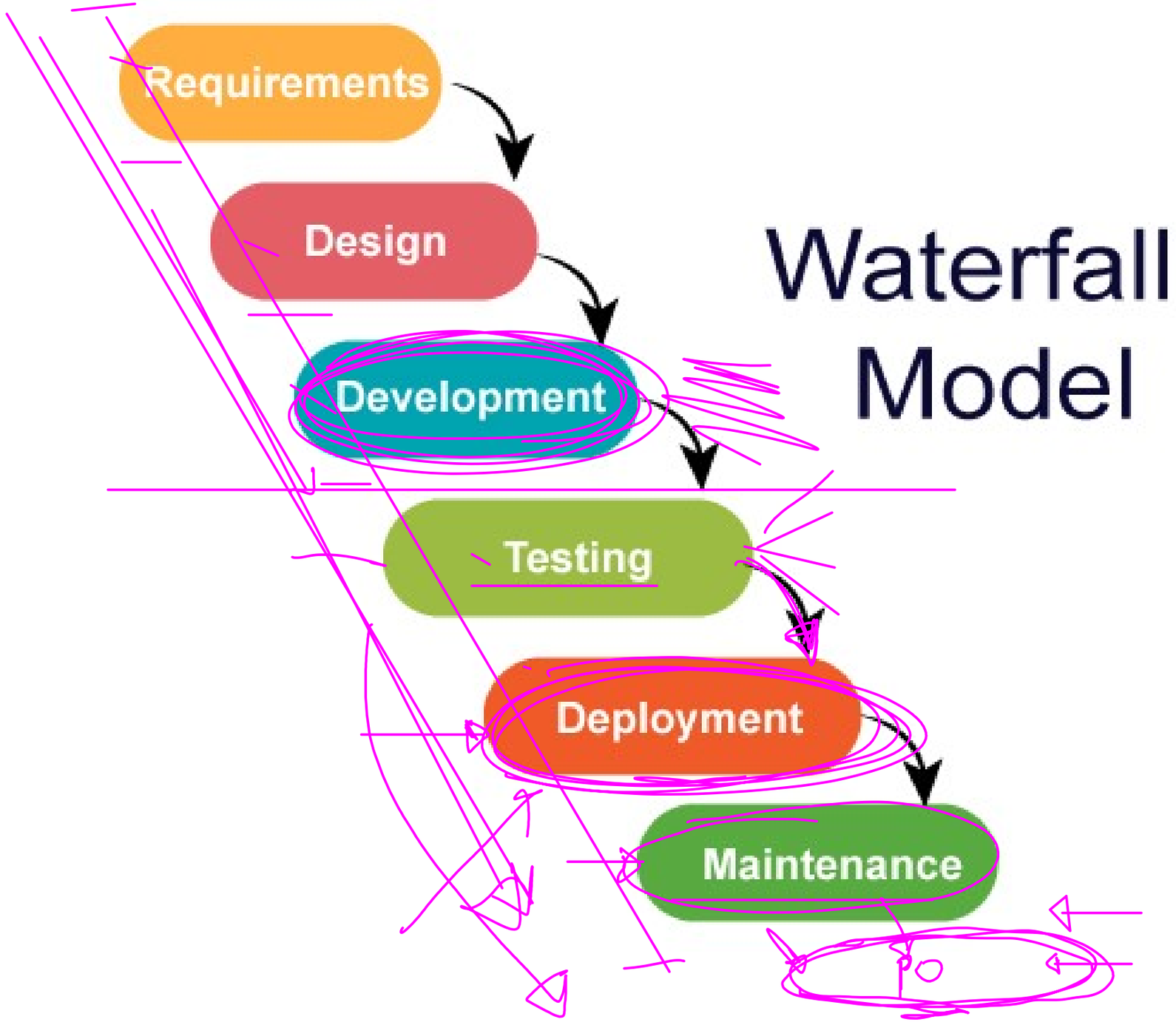


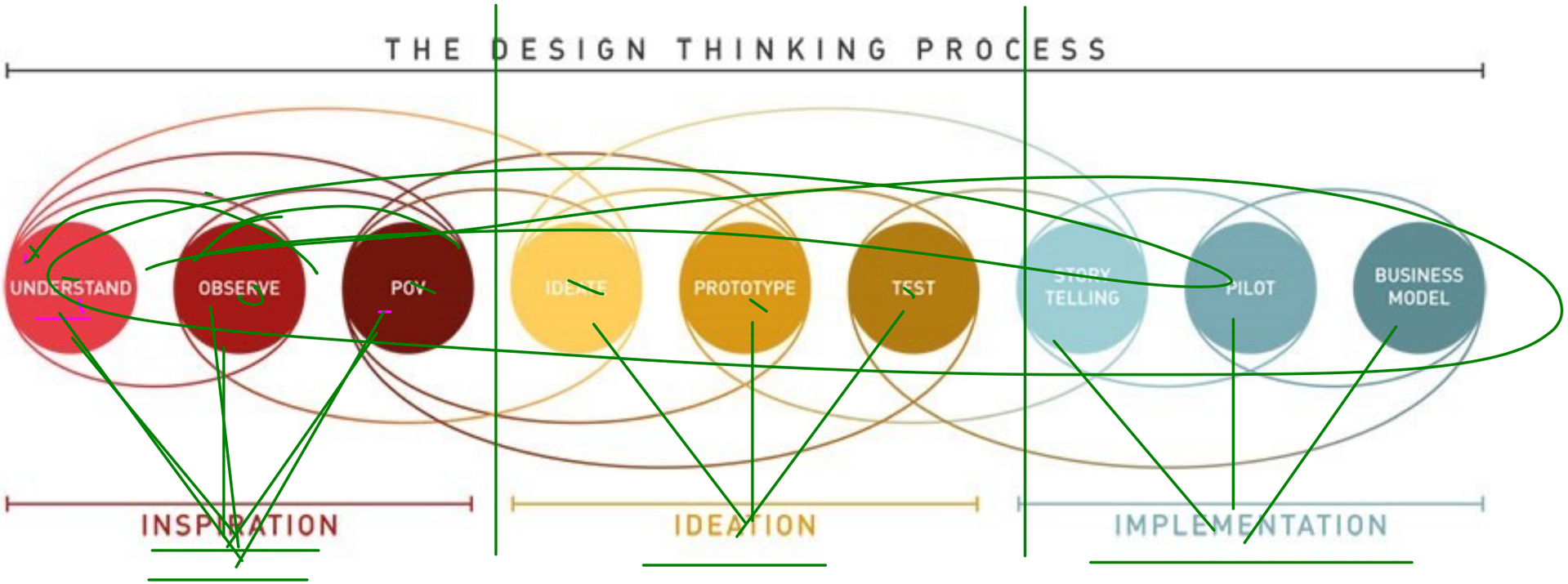

# “Computer programming” (210/211 style)




Requirement
• Requirement Doc.
• Prepare Use Cases
Design
• Software architecture
• Map the stakeholders
Implementation
• Construct the software
• Data storage & retrieval
Verification
• Install
• Test and Debug
Maintenance
• Check errors
• Optimize capabilities




# Waterfall Model


read → understand → plan → execute → verify → reflect