lecture_2_2:

Software Development Process

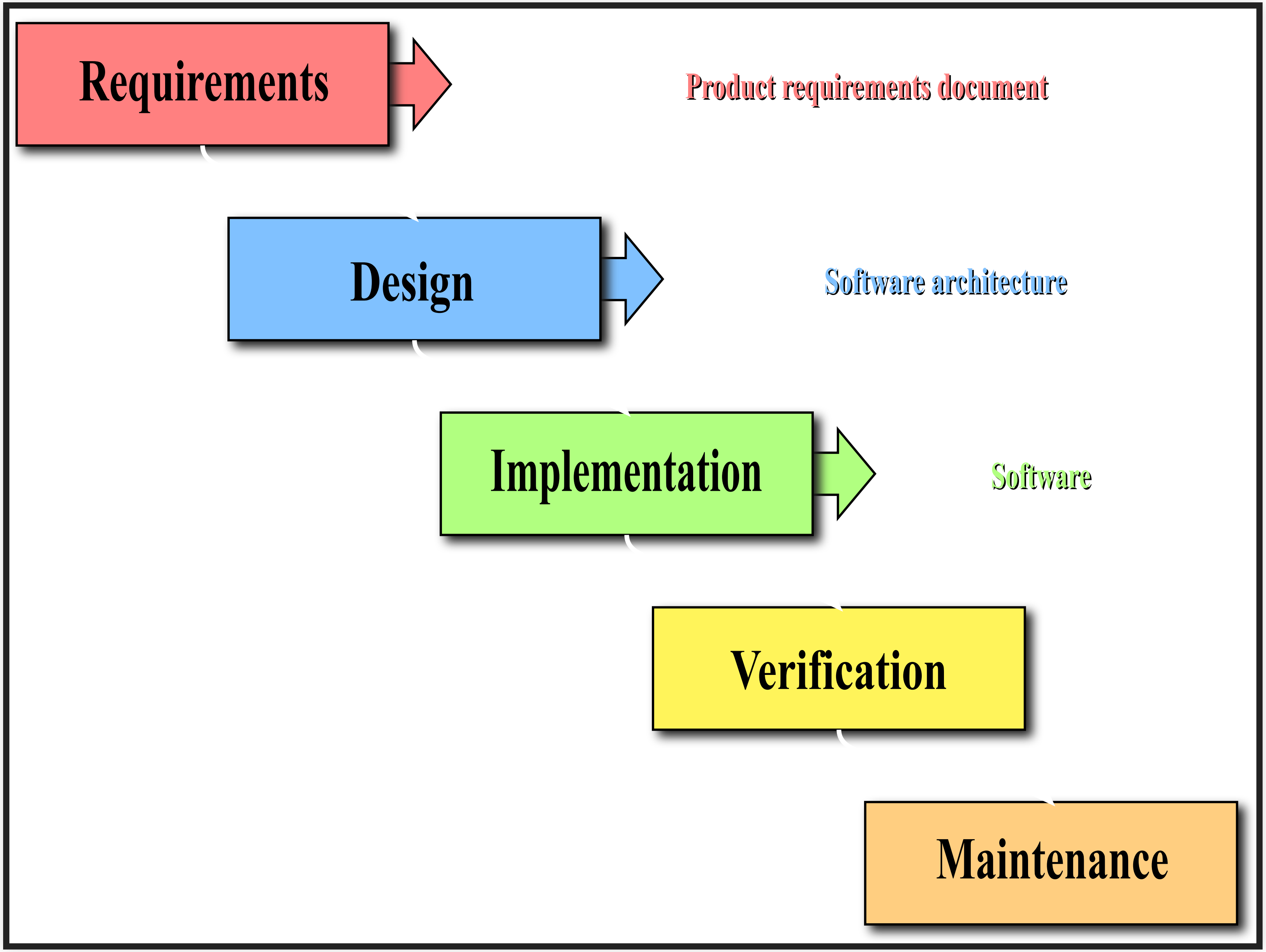
comp-prog/lecture_2_2
Matt Gottsacker
Marquette University High School

Last modified: 09.05.2019

Software development process: overview

- There is usually a question on the AP test about software dev.
- This section will shape how you approach programming problems, especially the group projects. Your README files will demonstrate your understanding of this section.
- Programming is engineering.

Waterfall methodology (pg. 54)



Requirements

Product requirements document

Design

Software architecture

Implementation

Software

Verification

Maintenance

1. Requirements specification

- Understand and dictate what `problem` the software will solve.
- The textbook states problems clearly. In the wild, problems are not always clearly defined, so developers need to interact with stakeholders frequently.

2. System analysis

- Based on requirements specification, determine outputs of the software.
- Determine how to reach the outputs.
- Determine what the program's inputs are.
- System analysis involves constructing/enumerating any formulas required.

3. System design

- Describe the algorithm being used.
- Break down the problem into components that are solved by subsystems.

4. Implementation / development

- Write code. Create the program.

5. Testing

- Check that the code meets requirements and does not have any bugs / mistakes.
- How do you know it works?
- For your group projects, I will require you to describe test cases that you use. For example, you may test it using simple values, or check complex calculations with a calculator.

6. Deployment

- Release the software for install, or host it on a website.

7. Maintenance

- Security updates
- Feature improvements
- Fix bugs
- Incorporate changing requirements

Waterfall vs. Agile

- Waterfall development is effective for small projects with clear requirements.
- For many development teams, the software to build has changing requirements that are not clear at the project's inception. These teams are often cross-functional and smaller. They use the Agile development methodology.

Waterfall vs. Agile

end