Software Development Life Cycle

The Waterfall Method

- This is the oldest method for development
- It assumes that the software is conceived, developed and delivered
- Development always moves forward

The Waterfall Method

Waterfall: Specify

- What exactly does the user want?
- What does the program do?
- What is the input?
- What is the output?
- What does it look like?

- This stage is critical if you are to develop the program the user wants!!

Waterfall: Specify

- This is the point where many projects fail or waste significant amounts of time
- If the software developer does not clearly understand what the user wants, there is no chance that the software will ever be written properly
- Problems also occur when the customer doesn't really know what they want!

Waterfall: Design

- In this stage, determine HOW you will solve the problem
- No code yet
- Break the problem into sub-problems
- Determine the way you will proceed
- Determine which team members will tackle which tasks

Waterfall: Code

- Finally you can write code
- Stage 1 is needed so you write the correct code
- Stage 2 is needed to save time
  - you have already decided what you need
  - a clear design can greatly improve the speed and quality of code

Waterfall: Test

- Here you test your code
- How will you test your code?
- What needs to be tested?
  - inputs, outputs, stability
- How much is enough testing?
  - varies greatly depending on the customer
  - some customers can tolerate “buggy” code, while others demand/require rock-solid code

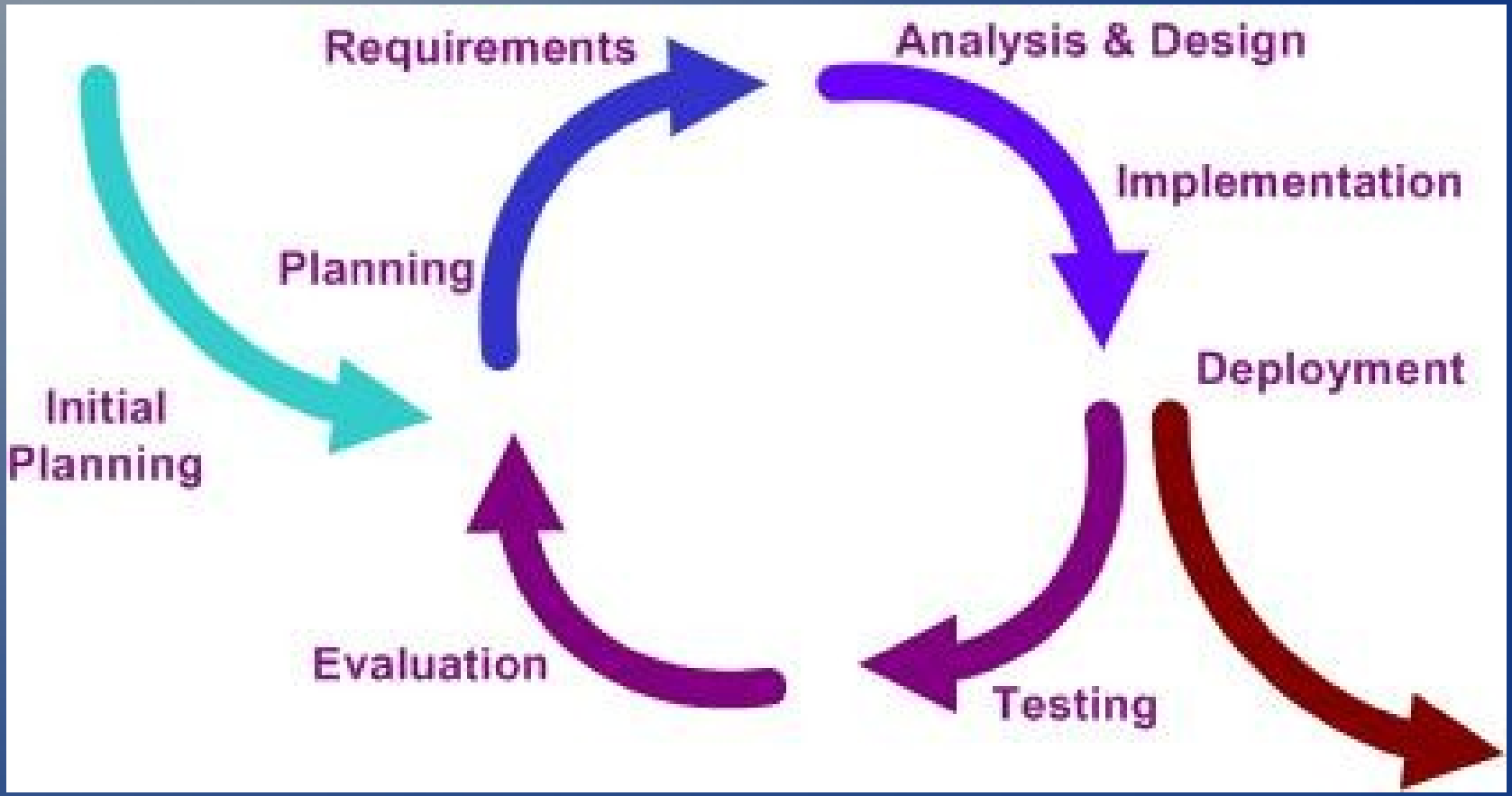
Waterfall: Deliver

- Here you provide the program to the user and ensure that it functions
- Maintenance is also a part of this stage
  - if your testing was not good enough, this is when you will hear about it!
  - some companies will knowingly ship faulty code and try to fix it in the field
    - this is among the worst practises in software development

Iterative Method

- The waterfall method does not represent the real world of software development
- In reality, there are many places where “steps” are repeated
- There is also no place for correcting errors in the waterfall method
  - even the best designs and code have errors that must be corrected after they are discovered

Iterative Method



- Image courtesy of Wikipedia - http://en.wikipedia.org/wiki/Image:Iterative_development_model_V2.jpg

Iterative Method

- In this method smaller steps are taken at first, then larger and larger steps
- In some respects this model carries out the waterfall method continuously (in a loop)
- After each step, the product is tested and evaluated (possibly with customers)
- Each step is used to provide feedback on the design of the next step

Iterative Method

- Small improvements made with each step
  - For example, all the sub-programs are likely identified and constructed with few details at first
- A shell version of the program may be developed early to show customers
  - this also helps the customer refine their own expectations of the software

Iterative Method: Evaluation

- Uncover design faults earlier
- Assist customer in refining their ideas
- Improve common understanding
- May miss out on good structure due to lack of planning
- Time tables are difficult to determine

Agile Methods

- These are a type of iterative method
- Built on feedback as the controlling force for the project rather than planning
  - rather than cycling through steps, the process jumps to whichever step needs the most attention
- Seem to provide better products but are harder to develop schedules for
- Best known type is Extreme Programming

Extreme Programming

- Initial steps may take very little time
- Phases are intentionally incomplete
- Automated tests are included as part of the programming
- Pairs of programmers work until the code segment passes the tests
- Incomplete segments are demonstrated to user

Exercises

In a small group (1-3 students), work through the Requirements (what features do you need?) and Design (how will you implement those features?) phases for a Hotel Booking System.

You will need to play the role of customer and software developer in determining the requirements.

Prepare a document that matches the features to the design element that addresses that feature.