Objectives

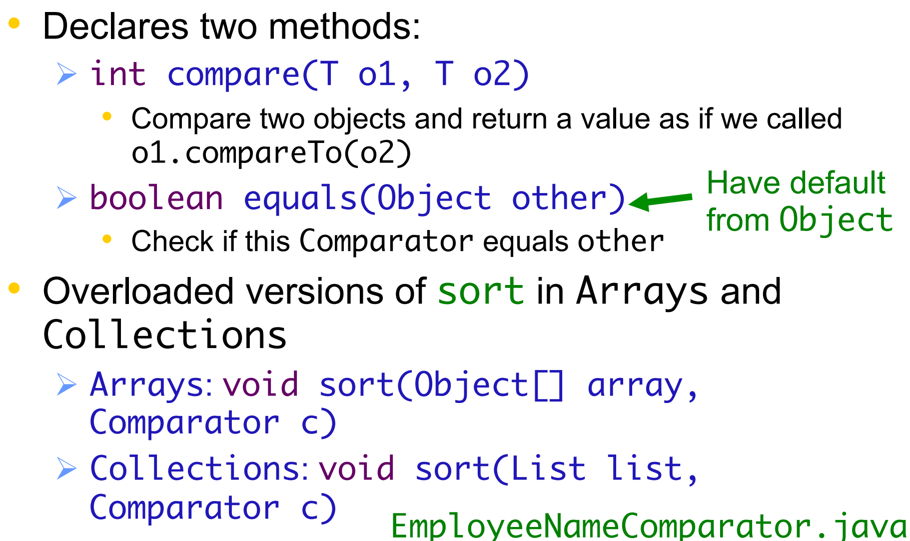
- Java wrap-up
  - Compilation benefits
  - Comparing with Python
- Software Development

COMPARATORS

Alternative Sorting

- What if object is `Comparable` but does not sort the way you want?
  - Special case
    - Don't want to change class
    - Don't have access to class
  - Example: want to sort strings so capital and lowercase letters are the same
- Use `Comparator` interface

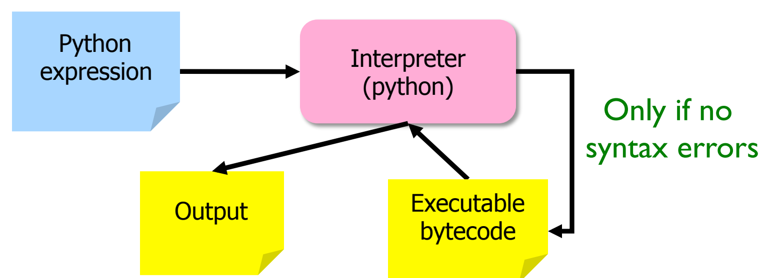
Comparator<T> Interface



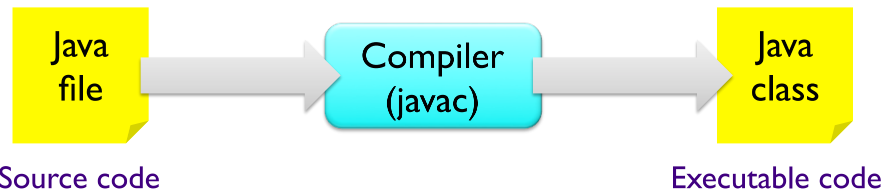
COMPILATION

Python Interpreter

1. Validates Python programming language expression(s)
   - Enforces Python syntax rules
   - Reports syntax errors
2. Executes expression(s)



Java Compiler



Source code

Executable code

- Lexical analysis, parsing, semantic analysis, *code generation*, and *code optimization*
- Code optimization: dead code eliminator, inline expansion, constant propagation, …

Compiling

- Translates high-level programming language to machine code or byte code
  - Java: .java → .class == bytecode
- Compiler optimization techniques
  - Generate *efficient* bytecode/machine code
  - Examples: get rid of unused local variables, transform loops, inline method calls
  - In Java: static typing for additional gains
- Can execute generated code multiple times
  - Performance gain
  - Interpreted → have to re-verify the code each time executed

LANGUAGE COMPARISON

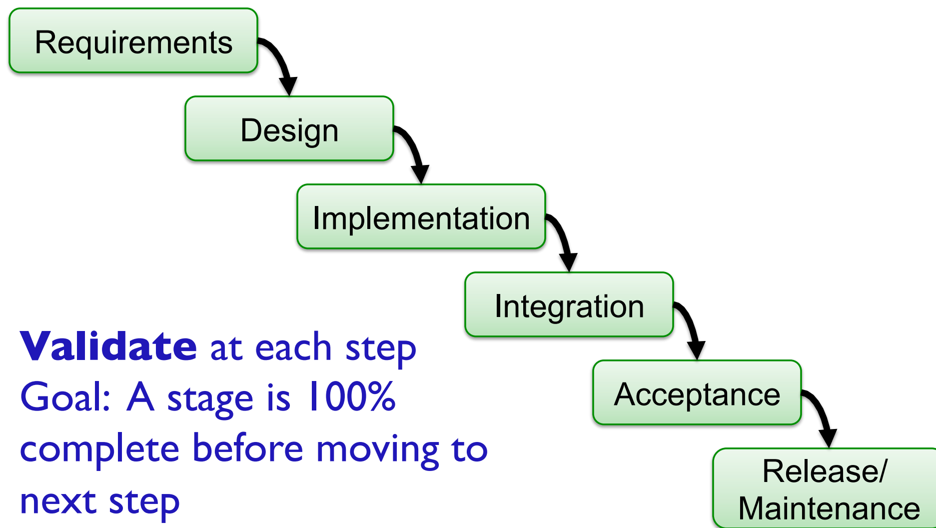
Language Comparison

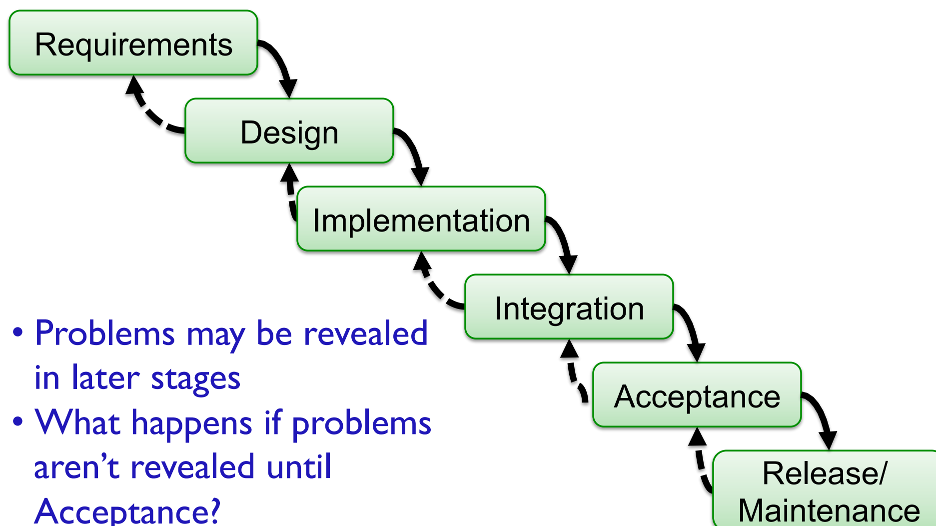
Java **Python**

Rest of the semester

- Shift from learning Java, specifically, to learning how to develop software (abstractly) with Java as our implementation/example
- Why Java?
  - Popular language
  - Many frameworks and tools for Java
  - Java's structure allows for strict adherence to design techniques
- Just a start
  - You'll need to continue learning more "on the street"

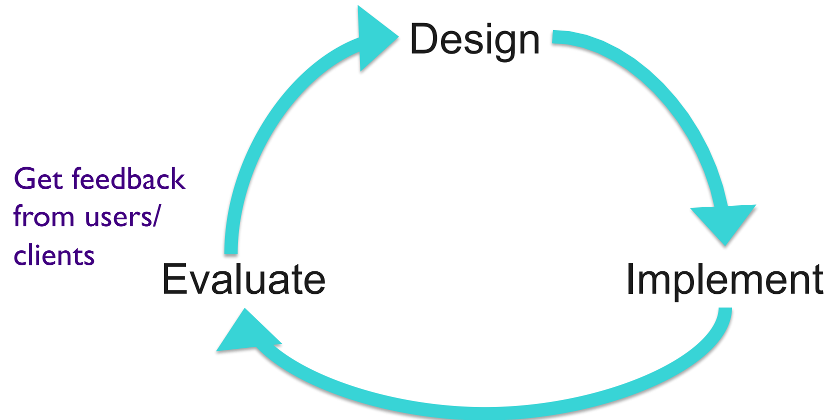
SOFTWARE LIFE CYCLE


Traditional Software Engineering Process: Waterfall Model
Requirements
Design
Implementation
Integration
Acceptance
Release/ Maintenance
Validate at each step
Goal: A stage is 100% complete before moving to next step
Oct 10, 2016
Sprenkle - CSCI209
13


Feedback in Waterfall Model
Requirements
Design
Implementation
Integration
Acceptance
Release/ Maintenance
• Problems may be revealed in later stages
• What happens if problems aren't revealed until Acceptance?
Oct 10, 2016
Sprenkle - CSCI209
14

Iterative Design

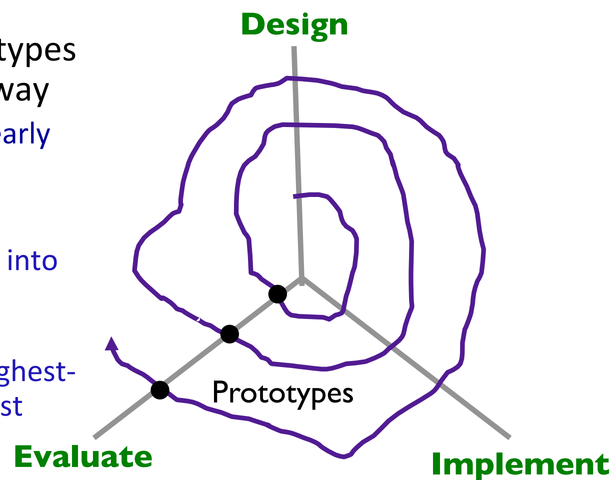
Get feedback from users/ clients



Spiral Model

Consistent with agile development

- Idea: smaller prototypes to test/fix/throw away
  - Finding problems early costs less
- In general…
  - Break functionality into smaller pieces
  - Implement most depended-on or highest-priority features first

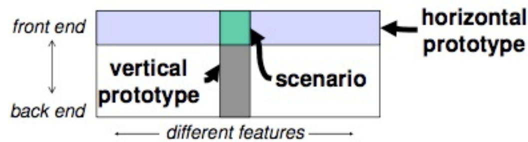


Radial dimension: cost

[Boehm 86]

Prototypes

- Purpose/Dimensions
  - Functionality
  - Interaction
  - Implementation
- Fidelity:
  - Low: omits details
  - High: closer to finished project
  - Multi-dimensional
    - Breadth: % of features covered
      - Only enough features for certain tasks
    - Depth: degree of functionality
      - Limited choices, canned responses, no error handling

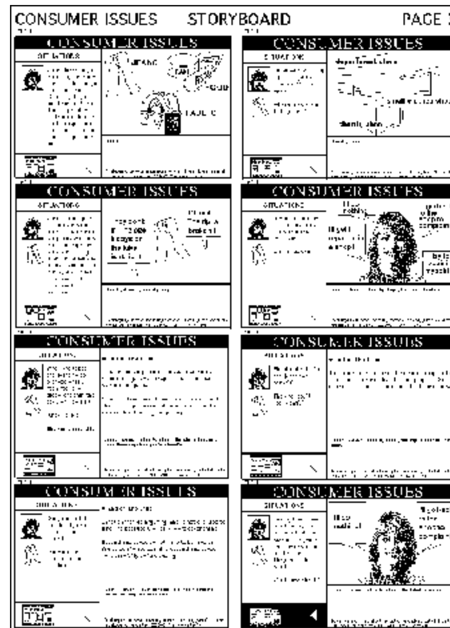


From Nielsen, *Usability Engineering*

Oct 10, 2016 Sprenkle - CSCI209 17

Low Fidelity Prototypes

- Media: Paper
- Examples: storyboard, sketches, flipbook, flow diagram



Oct 10, 2016 Sprenkle - CSCI209 18

High Fidelity Prototypes

- Media: Flash, HTML (non-interactive), PowerPoint, Video
- Examples: Mockups, Wizard of Oz

Virtual Peer for Autistic Children

http://www.articulab.justinecassell.com/projects/samautism/index.html

How to Implement an Effective Solution

- Understand the problem (interact with people)
- Understand external constraints (interact with people)
- Design an effective solution to the problem
- While designing the solution, design some tests to verify that the problem is solved (and remains solved)
- Code the effective solution to the problem
- Teach other team members about your solution to the problem (interact with people)

Spiral Model Steps

- Design a {method, class, package}
- Implement the {method, class, package}
- Test the {method, class, package}
- Fix the {method, class, package}
- Deploy the {method, class, package}
- Get feedback
  - Probably will require modifications to design
  - May even need to rollback a previous version
- Repeat