Topics

- Quiz 2
- Software Development Processes (cont)

Homework

- HW2 due
- HW3 out
- Quiz 3 next Tues
Incremental Development  [“Rapid Dev.”, McConnell]

How it works:

- waterfall-like requirements and design steps
- show (deliver) software in each stage
  - partition system into subsystems in beginning
  - implement one subsystem at a time
  - each stage (increment) is a traditional waterfall approach

Q: Advantages
- earlier feedback and signs of progress, subsystems could be used as released

Q: Disadvantages
- requires careful planning of subsystems (to avoid backtracking)
Iterative Development  [“Rapid Dev.”, McConnell]

How it works:

- build initial prototype and extend (iterate) until accepted
- finish testing and then release final product
- each iteration focuses on visible aspects (to customer/users)

Q: Advantages

- identify and adapt to changes early, time/budget constraints

Q: Disadvantages

- can be hard to estimate time/budget early on
- can lead to “do-stuff-until-we-run-out-of-time-or-money” (common scrum misconception)
- can degrade to code-and-fix if not careful
Incremental vs Iterative

Assume we want to build a word processor with features for:

- creating/editing text
- organizing text (e.g., cut and paste)
- formatting text (e.g., font size, style, margins)
- etc.

**In Incremental Development:**

- collect and analyze requirements
- define subsystems (e.g., viewing, editing, managing files, ...)
- order subsystem releases: managing files, viewing, editing, ...
- 1st increment: files; 2nd increment: viewing, ...

**In Iterative Development:**

- develop an initial prototype, e.g., basic features for opening/saving/editing
- then start iterating prototype
- e.g., first iteration might allow for bold-face text and margins
- next iteration some additional styling and cut/paste
- next on copy and font size
- and so on
Agility & the Scrum Development Process ...

Scrum uses a combination of incremental and iterative Approaches

- We’ll go over more of the details of the process later

- Initial planning phase like in incremental

- ... but less “comprehensive” and “document-heavy” (more later)

- Short “sprints” ... e.g., 1-2 weeks long
  - that include more requirements analysis, design, code, test, planning
  - like increments in that end product of sprint is ideally “released”
  - like iterations in that get feedback and incorporate into future sprints

- Each sprint has
  - a goal (what functionality are you tackling?)
  - something new to show customer at end of sprint (visibility)
  - review with customer/users & plan next sprint

- “Product Backlog”: unfinished features, tasks, etc, for project
- “Sprint Backlog”: unfinished features, tasks, etc, for sprint
Requirements

The plan

- “user stories” (agile/scrum technique)
- requirements in general

Requirement Statement (or just requirement)

- a single feature or characteristic of the system
- needed by the customer to make the system useful

Requirement Specification

- a set of prioritized requirement statements
- together make up the requirements of the product
- ... captured through the product backlog in scrum
User Stories

A specific kind of requirement statement

- describes one thing the software must do (for user)
- written in “language the customer understands”
- written “by the customer”
  - not literally ... in the “voice” of the customer/user
  - driven solely by customer need ... even if dev wrote it
- short and sweet!
  - usually 1 sentence
  - should fit on one side of a 3x5 card
- Meant to start discussion about functionality & features