Software Engineering 301:
Software Analysis and Design

Introduction

Prof. Robert Walker (Rob)

Agenda

- Why is software development in the real-world different?
- What are software analysis and design and how do they fit into software development?
- How is the course structured?
- How will you be evaluated?

Introductory programming assignment

- Working by yourself
- Precise specification
- Ambiguities? Complain, ask for clarification
- Implement it
- Run it to “make sure” it works
- Fix some problems (maybe)
- When done, forget about it

Real software development

- More than just building one data structure
- Scale matters
  - 100s of LOC or billions of LOC of source?
  - Weeks or years? New version how often?
  - 1 developer or 1000 developers?
  - Clear customer, or aiming for a market?
- Change is a constant reality
  - Changing ideas, needs, people
- "Messy"
  - Conflicting ideas, conflicting goals
- *How can we cope?*

Software engineering

The application of a disciplined approach to the development, operation, and maintenance of software, and the study of these approaches

- Traditionally, software engineering revolves around:
  - Models
  - Activities
  - Process (ordering and interleaving of activities)

Activities in development

- What should be done?
  - Read description/talk to people
    - **Requirements elicitation (or gathering)**
  - Write down the needs in some manageable form
    - **Requirements specification**
  - Has anything been missed? Realistic? Self-contradictory?
    - **Requirements analysis**
- How should it be done?
  - **Design**
- Do it.
  - **Implementation**
- Has it actually been done?
  - **Testing**

Additional activities

- Deployment
- Maintenance & enhancement
- Management
- Configuration
- Communication (incl. documentation)

Software development is not algorithmic

- Too uncertain
- Too in need of balancing a lot of external concerns
  - Time, money, people, technical issues, business issues, changing environment, ...

- BUT, software development can still be *disciplined*
  - Identification and application of best practices
  - Conscious decisions about when and how to apply these

Sound depressingly complicated?

- It *is* complicated
- But that is why human beings do it (and not machines)
- It is also fun:
  - Social interaction, investigating problems, math and science, tools and ideas, human strengths and weaknesses …
  - Software development can involve almost any factor you can think of

What is this course about?

- In a nutshell, this course is an introduction to
  - analyzing and solving the problems ...
  - encountered in building ...
  - non-trivial software ...
  - in the real world

- We will only briefly look at requirements, process, and evolution
- The focus of the course is on analysis and design

Agenda

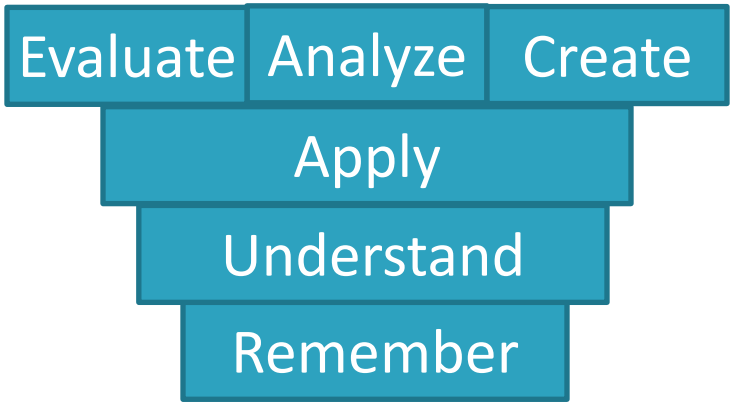
- ~~Why is software development in the real-world different?~~
- What are software analysis and design and how do they fit into software development?
- How is the course structured?
- How will you be evaluated?

Bloom's taxonomy

- Six levels of skills or objectives

Original (1956)

Revised (2000)



Analysis and design

- Design
  - (to decide on) the structure and behaviour of the system
- Analysis
  - [Wikipedia] process of breaking a complex topic into smaller parts in order to gain a better understanding of it
- Software development is all about analysis, evaluation, and creation!

Agenda

- ~~Why is software development in the real-world different?~~
- ~~What are software analysis and design and how do they fit into software development?~~
- How is the course structured?
- How will you be evaluated?

The course calendar

- Schedule for:
  - Lectures, due dates, midterm exam

Topics

- First half:
  - Brief review of object-oriented concepts
  - The creation and analysis of models
  - Basics of requirements
  - Testing
- Second half:
  - Design
  - Practice
- (All of these things involve analysis)

Web presence

- Desire2Learn
  - Calendar
  - Lecture notes
  - Laboratory exercises
  - Assignments
  - Readings
  - Links to resources

Communication

- Rob's Office hours: TR 12-13 in ICT 546

- Please use the Desire2Learn forums for all non-personal issues
- Email: walker@ucalgary.ca (for personal issues)

Textbooks

- **Required:** *Object-Oriented Software Engineering: Practical Software Development using UML and Java, 2nd Edition*, by Lethbridge and Laganière

- **Recommended** (available on-line through the library):
  - *How to Use Objects*, by Gast
  - *Code Complete*, *2nd Edition*, by McConnell
  - *Clean Code*, by Martin

Assessment

- Assignments (4 of them) 35%
  - A1: simple vending machine 5% [Sep 30]
  - A2: modified VM, modelling 5% [Oct 21]
  - A3: modified (again) VM, testing 10% [Nov 4]
  - A4: complex VM, design 15% [Dec 9]

- Grades back to you within a week of deadline
- A4: grades back to you within 3 days

Assessment

- Lab exercises (15 of them) 15%
  - Each one is pass/fail
  - You pass if you attempted to answer the questions, even if your answers are all wrong
  - You can retry as many times as you want
  - The labs are supposed to (!) tell you where you went wrong, automatically
  - Labs 1-9 are due before the midterm; the rest before midnight on the last day of classes

Assessment

- Midterm exam 20% [Nov 17 in class]
  - Covers modelling, [requirements], testing
- Final exam 30% [date TBD]
  - Covers whole course

- Multiple choice, one page cheat sheet
- To get C- or better, you have to get a C- or better on the exam combination

Due dates

- Due dates have been carefully chosen
  - Must fit around appropriate lectures
  - Allow enough time for successive assignments
  - Allow time for grading
- No extensions except in unusual circumstances
  - TAs cannot grant extensions
  - If you talk to me early about your issue, it is more likely that accommodations can be made for you
- *Use your time wisely; that's why there's a schedule*

Grading

- All assignments and exams will be assigned letter grades, not percentages
  - A-range: mastery, you display deep reasoning abilities
  - B-range: good, but you are weak on some major point
  - C-range: OK, you know the basics but that's it
  - D-range: significant problems in your knowledge/reasoning/skills
  - Then, +/- adjustments within the range

Prerequisites

- CPSC 331 or 319 with at least C-
- To request a prerequisite waiver, please see the CPSC main office or the CPSC website (not me)
- Otherwise, if you don't have the prerequisite, you will be deregistered from the course at some point (not by me)

Writing

- "Writing and the grading thereof is a factor in the evaluation of the student's work."
  - This means in English

- You will also be writing source code in Java
  - Other languages are not an option
  - We won't be teaching you about Java specifics

Plagiarism and collaboration

- You are required to cite all sources of information that you use
- You may talk to each other about assignments, but you must write up your answer individually including the source code
- Don’t look at each other’s source code
- Detailed plagiarism and collaboration policy is on the course website: read it

My philosophy

- People learn more from challenges
- I'm not keen on "busy work" (repetitive, trivial, etc.)
- I'm not keen on straightforward memorization of "facts"
- I want you to learn to ask "why?" and attempt to answer it too
  - analysis & design fit perfectly with that

My questions of you

- I will ask you questions in class
- There are two cases:
  1. I'm checking whether you've understood some point, or whether I need to explain something differently
  2. I'm challenging you: welcome to university

  – I rarely ask "trick" questions
- There is no penalty for giving a wrong answer (in class)!
  – You learn more from making mistakes

Next time

Review of object-oriented concepts