Professor Hausi A. Müller PhD PEng FCAE
Department of Computer Science
Faculty of Engineering
University of Victoria

http://www.engr.uvic.ca/~seng321/
https://courses1.csc.uvic.ca/courses/201/spring/seng/321

Announcements

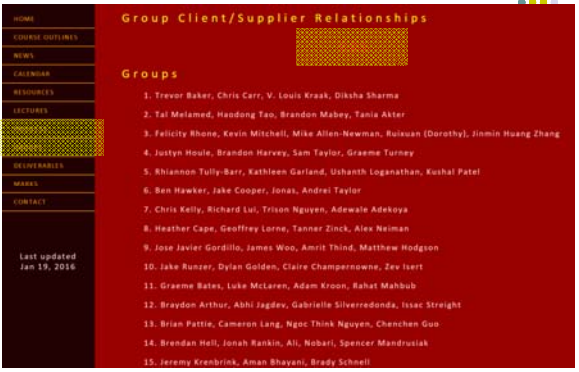
- New class room as of Wed
  - MAC 288 (original one)
- Midterm rescheduled due to lab clash
  - Fri, Feb 26 in class confirmed !!
- Course website updated
  - Split Projects page into Groups and Projects pages
- Assignments/Deliverables
  - S0, C0, S1, C1 specs posted
  - Group website spec posted
  - www.engr.uvic.ca/~seng321/deliverables.html
- Projects and Groups
  - www.engr.uvic.ca/~seng321/projects.html
  - www.engr.uvic.ca/~seng321/groups.html

SENG 321 Calendar

First day of classes	Tue, Jan 5
Labs begin	Tue, Jan 12
Reading break	Feb 8-12
Midterm	Fri, Feb 26 (confirmed !!)
Easter break	Mar 25-28
Project presentations	Mar 29-31
Last day of classes	Fri, Mar 31

Detailed course calendar: deliverables deadlines
http://www.engr.uvic.ca/~seng321/calendar.html

Groups



Last updated
Jan 19, 2016

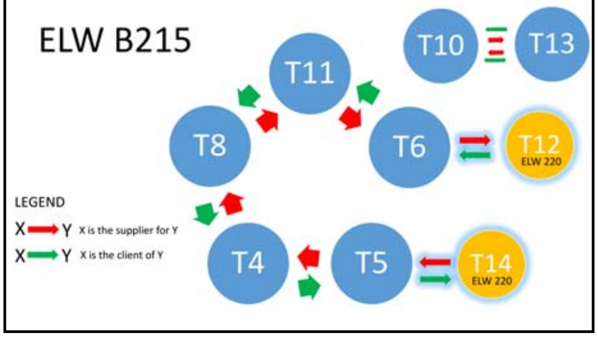
Group Client/Supplier Relationships

Groups

1. Trevor Baker, Chris Carr, V. Louis Kraak, Diksha Sharma
2. Tal Melamed, Haodong Tao, Brandon Mabey, Tania Akter
3. Felicity Rhone, Kevin Mitchell, Mike Allen-Newman, Ruixuan (Dorothy), Jinmin Huang Zhang
4. Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney
5. Rhiannon Tully-Barr, Kathleen Garland, Ushanth Loganathan, Kushal Patel
6. Ben Hawker, Jake Cooper, Jonas, Andrei Taylor
7. Chris Kelly, Richard Lui, Trison Nguyen, Adewale Adekoya
8. Heather Cape, Geoffrey Lorne, Tanner Zinck, Alex Neiman
9. Jose Javier Gordillo, James Woo, Amrit Thind, Matthew Hodgson
10. Jake Runzer, Dylan Golden, Claire Champernowne, Zev Isert
11. Graeme Bates, Luke McLaren, Adam Kroon, Rahat Mahbub
12. Braydon Arthur, Abhi Jagdev, Gabrielle Silverredonda, Issac Streight
13. Brian Pattie, Cameron Lang, Ngoc Think Nguyen, Chenchen Guo
14. Brendan Hell, Jonah Rankin, Ali, Nobari, Spencer Mandrusiak
15. Jeremy Krenbrink, Aman Bhayani, Brady Schnell

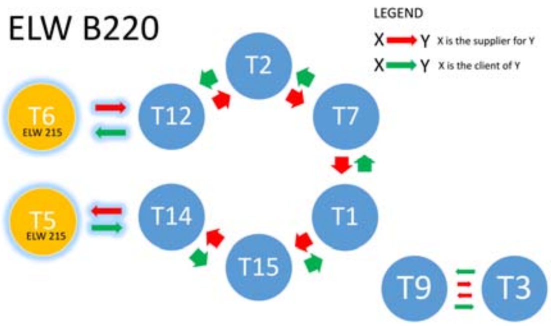
Harshit Jain

ELW B215



Priya Angara

ELW B220

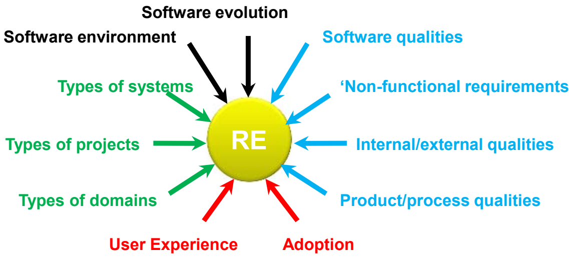


Students must participate in all project presentations in class & labs
No show results in a 25% reduction in the mark for that presentation

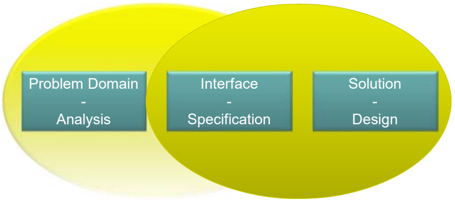
Project Deadlines and Marks

1.	Call for Project Proposals		**6 Jan (Class)**
2.	Request for Proposal (RFP)		**8 Jan**
3.	Project selection		**12 Jan (Lab)**
4.	Team selection		**14 Jan (Lab)**
5.	Related work (S0)	5%	**22 Jan (Lab)**
6.	Project website up and running	5%	**26 Jan (Lab)**
7.	RFP2 Informal Requirements Specification (C0)	5%	**29 Jan (Lab)**
8.	Formal Requirements Spec (S1)	10%	**16 Feb (Lab)**
9.	Customer Feedback on S1 (C1)	5%	**18 Feb (Lab)**
10.	Detailed Requirements Spec (S2a)	10%	**1 Mar (Lab)**
11.	Prototype demo (S2b)	5%	**3 Mar (Lab)**
12.	Customer Feedback on S2a-b (C2)	5%	**8 Mar (Lab)**
13.	Final Requirements Spec (S3a)	15%	**15 Mar (Lab)**
14.	User Manual (S3b)	10%	**22 Mar (Lab)**
15.	Customer Feedback on S3a-b (C3)	5%	**24 Mar (Lab)**
16.	Demo Final Project (S4)	10%	**29,31 Mar (Lab)**
17.	Customer Feedback on S4 (C4)	5%	**29,31 Mar (Lab)**
18.	Instructor and TA Evaluations (S5)	5%	**1 Apr**

Requirements Engineering Many Forces at Work

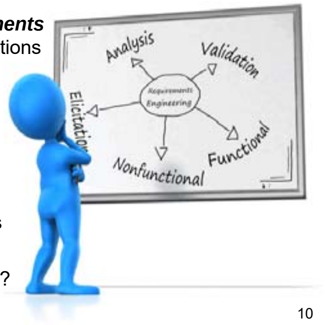


Separation of Concerns



Functional and Non-functional Requirements

- ***Functional requirements*** describe system functions or services
- ***Non-functional requirements*** is a constraint on the system or on the development process
- What's the difference?



What is Quality (Pressman)?

- Conformance to explicitly stated requirements, standards, and implicit characteristics
- Functional and non-functional **requirements**
  - Foundation from which quality is measured
  - Lack of conformance ← → lack of quality
- Explicitly documented development **standards**
  - Development criteria guide manner software engineered
  - Criteria not followed → lack of quality
- Implicit characteristics expected of professionally developed software
  - Often go unmentioned (e.g., desire for good maintainability)
  - Even if explicit requirements met, failing to meet implicit requirements suggest suspect software quality

Quality Factors

- **Correctness:** fulfill specifications
- **Reliability:** perform function with required precision
- **Efficiency:** resources & code required to perform function
- **Integrity:** controlled access to software / data
- **Usability:** effort required to learn / operate / interpret
- **Maintainability:** effort to test program to ensure functionality
- **Flexibility:** effort required to modify operational program
- **Portability:** effort to transfer to other environments
- **Reusability:** extent to which components can be reused
- **Interoperability:** effort to couple system with another

Software qualities

- Software engineering is concerned with software qualities
- Qualities (a.k.a. “ilities”) are goals in the practice of software engineering
- The qualities are usually expressed as non-functional requirements during the early design stages
- External qualities
  - visible to the user
  - reliability, efficiency, usability
- Internal qualities
  - the concern of developers
  - they help developers achieve external qualities
  - verifiability, maintainability, extensibility, evolvability, adaptability

Software qualities …

- Product qualities
  - concern the developed artifacts
  - maintainability, understandability, performance
- Process qualities
  - deal with the development activity
  - products are developed through process
  - maintainability, productivity, timeliness

Discussion

Are security requirements satisfied?

- How can we measure security quality?
- Can security quality be measured using static analyses?
- What can be measured using static analyses?
- How can we instrument the code to validate security requirements?
- What can be measure using dynamic analyses?

Evolution of Software Systems

Successful requirements engineering will take the projected evolution of a software system into account

Different Phases in Software Development Cycle

1. Requirements
2. Architecture
3. Design
4. Coding
5. Testing
6. Maintenance

- Most emphasis during undergraduate program is on phases 3-5
- Which phase is hardest, costliest, and most time consuming?
- Which phase lasts the longest?