

# ***CMPT 470 – Advanced Software Engineering***

## ***Course Outline***

### **Meeting Times**

Lectures: Monday, Wednesday, Friday at 1:30-2:20pm in Arts 213

### **Instructors**

Class Instructor: Nathaniel Osgood, Thorvaldson 280.6, 966-6102, [osgood@cs.usask.ca](mailto:osgood@cs.usask.ca)

Office Hours: Thursday 3:30 – 5pm and by appointment.

### **Texts**

There is one required text for CMPT 470.

Weinberg, G.. *Quality Software Management, Vol 1: Systems Thinking*, Dorset House Publishing, 1991. ISBN: 0932633226

Additional readings from a variety of sources will be provided as PDFs on the course website. Please see the table of additional sources below.

### **Course Contents**

This course builds on the understanding of software engineering presented in CMPT 370 and (to a lesser degree) CMPT 371. The focus this semester is on *quality* software engineering. This discussion covers all technical phases of the software process including requirements, software architecture and design, language-software engineering interface, improving the quality of code, diagnostic measures such as peer reviews and testing, debugging, uses and misuses of metrics, formal approaches, issues in maintenance and operations. While the course discusses a broad constellation of technical and non-technical considerations that impact software engineering quality, most of the tools and suggestions presented concern software engineering practices and processes. This focus reflects the fact that issues in managerial decision making that impact quality form an important area of discussion in CMPT 371.

All students must be properly registered in order to attend lectures and receive credit for this course.

### **Workload**

Students' grades will be determined by a combination of assignments, results of an exploratory investigation into a topic in software engineering of the student's choosing, the final exam and class participation.

The due dates and weighting for submissions are as follows:

<b>Deliverable</b>	<b>Detail</b>	<b>% Mark</b>	<b>Due Date</b>
Problem set 1	Quality tradeoffs, Causal loop diagrams, dynamics	7%	Oct 6
Exploratory topic 1 (ET1)	Introductory proposal	N/A	Sep 20
Exploratory topic 2 (ET2)	Intermediate findings	10%	Nov 1
Problem set 2		8%	Nov 27

Exploratory topic 3 (ET3)	Exploratory project final report	20%	Dec 4
Final Exam		30%	TBD

In order to provide additional opportunity for students developing their understanding through discussion of course material, much of the course content is to be covered in selected readings. Class sessions will highlight components of this material, but students will be held responsible for reviewing and digesting all readings prior to class, and in discussing this material within class. Reflecting the importance attached to in-class discussion, a significant fraction (25%) of students' grades will be based on class participation. Most of this mark will reflect the participation in discussion of readings held in class. In recognition of differences in communication styles and interests among students, this participation score will also reflect interaction in office hours.

Assignment and course marks will be assigned on an individual basis.

## Topic Plan and Readings

Lecture slides will be provided via the course website when possible but are not guaranteed for all classes. Readings from Weinberg and other excerpts distributed electronically are shown below. Additional readings and URLs will be shown in the notes for the topic in question.

A preliminary lecture schedule is included below. Please note that because of the emphasis on class discussion, the exact timing of particular lectures (and associated reading) is subject to change. Updated schedules will be distributed throughout the term.

Date	Topic	Required Readings	PSet dist/du
Sep 6	Introduction & Overview	N/A	
Sep 8	Fundamental quality concepts	<b>W1:1</b> ,CST	ET1/
Sep 11	Fundamental quality concepts 2		ET1/
Sep 13	Fundamental quality concepts 3		
Sep 15	Quality cultures	<b>W1:2</b> ,W3:10	
Sep 18	Quality systems thinking 1: Feedback & dynamics	<b>W1:6</b> ,CLD	PS1
Sep 20	Quality systems thinking 2: Systemic view of projects	<b>W1:7</b> ,TOP,E:4	/ET1
Sep 22	Quality in Software Requirements 1	W4:16	ET2/
Sep 25	Quality in Software Requirements 2	W2:19,R:11	
Sep 27	Design Principles & Domain-Centric Design 1	W4:22,R:16, D:{1,2,3}	
Sep 29	Domain-Centric Design 2	B:2.6.1.2	
Oct 2	Language support for design	TBD	
Oct 4	Aspect-Oriented Programming	G:8	
Oct 6	Formal Methods & Issues 1	P:28 (Excerpt),#, DBC	/PS1
Oct 9	Thanksgiving		
Oct 11	Formal Methods & Issues 2	O:TBD	
Oct 13	Domain specific languages/metalinguistic abstraction	LWB,LL	
Oct 16	Domain specific languages/metalinguistic abstraction	G:11, $\lambda$	
Oct 18	Domain specific languages/metalinguistic abstraction	TBD	
Oct 20	Software architecture 1	SAP:{5,7},B:2.3.2	PS2/
Oct 23	Software architecture 2	TBD	
Oct 25	Quality coding 1	B:2.4.5,C:8	
Oct 27	Quality coding 2	WNG, HNG (skim)	
Oct 30	Integration and quality	C:29, T:18	
Nov 1	Diagnosis of Quality Issues: Overview	<b>W1:12</b>	
Nov 3	Detecting failures: Peer Reviews	W2:12, K:{2,11 [Excerpt]}	/ET2
Nov 6	Detecting failures: Test Design 1	HRD, <b>W1:13</b> , N.3,V.17	ET3/
Nov 8	Detecting failures: Test Design 2	S:TBD,CTM	
Nov 10	Remembrance Day		
Nov 13	Detecting failures: Test Design 3	REF:pp91-101, BRK:AppA,TTL	
Nov 15	Detecting failures: Test Design 4	DIFF	
Nov 17	Identifying the faults behind failures	<b>W1:14</b> , FAL, USR, C:23	
Nov 20	Software measurement 1	W2:12,15,SEM	
Nov 22	Software measurement 2	C:22.7, P:{15,22}, DRK, PRF:14	
Nov 24	Resolving faults & triage	W1:15, MNG:15	
Nov 27	Maintenance & Operations: Versioning	SAP:14, <b>W1:11 (review)</b>	/PS2
Nov 29	Maintenance & Operations: Refactoring	F:3	
Dec 1	Maintenance & Operations: Reverse Engineering	TBD	
Dec 4	A Missing Piece: Management Quality Impacts	W1:18	/ET3
TBD	Final Examination	N/A	N/A

NB: The schedule above – including the contents and dates of both lectures and management meetings – is subject to change. On occasion, updated schedules may be provided.

“X:n” denotes chapter number  $n$  within the source abbreviated by X (as described by the section on readings below). “X:ppN-M” denotes page numbers N through M within the source abbreviated by X.

## Readings

Readings for the course will be drawn from a variety of sources. The schedule above uses the abbreviations shown in the first column.

Abbrev.	Source
<b>CST</b>	Evaluating the Cost of Software Quality.
<b>TOP</b>	Software Defect Reduction Top 10 List.
<b>R</b>	Weigers, Karl. <i>More About Software Requirements</i> . Microsoft Press. 2005. ISBN 0735622671.
<b>D</b>	Evans, Eric. <i>Domain-Driven Design</i> . Addison-Wesley Professional. 2003. ISBN 0321125215.
<b>E</b>	DeMarco, T. and Lister, T. <i>Peopleware: productive projects and teams</i> . Dorset House, 1999, 2nd edition. ISBN: 0932633439.
<b>B</b>	Glass, R. <i>Building Quality Software</i> . Prentice Hall. ISBN 0130866954.
<b>G</b>	Czarnecki, K., Eisenecker, U. <i>Generative Programming</i> . Addison-Wesley Professional. ISBN 0201309777.
<b>LWB</b>	Fowler, Martin. Language Workbenches: The Killer-App for Domain Specific Languages?
<b>LL</b>	Little Languages and their Programming Environments
$\lambda$	The Lisp Experience.
<b>P</b>	Pressman, R. <i>Software Engineering: A Practitioner's Approach</i> , McGraw-Hill, 2005, 6th edition. ISBN: 0072853182.
<b>#</b>	The Spec# programming system
<b>DBC</b>	Meyer, B. Applying Design by Contract
<b>U</b>	UML Distilled
<b>O</b>	Warmer, J. and Kleppe. A. <i>The Object Constraint Language: Precise Modeling With Uml</i> . Addison-Wesley Object Technology Series. 1998. ISBN 0201379406.
<b>A</b>	Software Architecture in Practice
<b>C</b>	McConnell, S. <i>Code Complete</i> , Second Edition. Microsoft Press. 2004. ISBN 0735619670.
<b>WNG</b>	Sposky, J. Making Wrong Code Look Wrong.
<b>HNG</b>	Simonyi, C. Hungarian Notation.
<b>T</b>	McConnell, S. <i>Rapid Development</i> . Microsoft Press. 1996. ISBN 1556159005
<b>K</b>	Wiegers, K. <i>Peer Reviews in Software: A Practical Guide</i> . Addison-Wesley Professional. 2001. ISBN 0201734850
<b>N</b>	Kaner, C., Bach, J., Pettichord, B. <i>Lessons Learned in Software Testing</i> . 2001. Wiley. ISBN 0471081124.
<b>V</b>	Tremblay, J.P., Schneider, K., Cheston, G. <i>Software Development in an Object-Oriented Domain</i> . Prentice Hall. 2006. ISBN 0131500562
<b>S</b>	Craig, R. and Jaskiel, S. <i>Systematic Software Testing</i> . 2002. Artech House. ISBN 1580535089
<b>CTM</b>	Marick, B. Classic Testing Mistakes
<b>F</b>	Fowler, M. Refactoring: Improving the Design of Existing Code. 1999. Addison-Wesley.
<b>BRK</b>	Whittaker, How to Break Software
<b>HRD</b>	Whittaker, <i>What is Software Testing? And Why is It So Hard?</i>
<b>DIFF</b>	Beizer, B. Software is Different
<b>FAL</b>	Whittaker, Why Software Fails.
<b>USR</b>	Whittaker, Software's invisible users

<b>MNG</b>	Berkun, S. <i>The Art of Project Management</i> . O'Reilly and Associates, 2005, ISBN 0596007868
<b>PRF</b>	Austin, Robert. <i>Measuring and managing performance in organizations</i> . Dorset House Publishing Company. ISBN 0932633366
<b>REQ</b>	Weigers, Karl. <i>Software Requirements, Second Edition</i> . Microsoft Press. 2003. ISBN 0735618798.
<b>DRK</b>	The Darker Side of Metrics
<b>TTL</b>	A taxonomy and catalog of runtime software-fault monitoring tools.
<b>MET</b>	Fenton. <i>Software Metrics</i> . Chapman & Hall. 1994. ISBN 0412404400.
<b>SEM</b>	Software Engineering Metrics: What Do They Measure and How Do We Know?
<b>Wx</b>	Weinberg, Gerald. <i>Quality Software Management, Vol. x</i>

## Homework Submission and Evaluation

All homework will be distributed and submitted electronically using the WebCT System.

Completed assignments and other deliverables are to be submitted before 12 MIDNIGHT on their respective due dates also shown in the course outline. Because feedback on the answers provided will be provided directly in the submitted file, it is important that you submit your homework in a form that is editable by MS Word or Excel. If you do not have access to the appropriate software, please see the instructor after class.

10% per day will be deducted from late problem sets or term project phases up to a maximum of seven days. Problem sets and term project phases received after seven days will not receive any credit. Under certain extenuating circumstances extensions may be granted. Please contact the instructor or tutor prior to the due-date if an extension is required. Failure to complete the assigned course work will result in failure of the course.

Collaboration among students on problem sets or term project phases to be completed individually is limited to discussing concepts and clarifying issues. Nonetheless, each student is expected to produce his or her own solutions to the homework problems. For a further discussion, please see the section on Academic Honesty.

In addition to the above, the course includes a final exam. Failure to write the final exam will result in failure of the course.

Students are expected to adhere to University of Saskatchewan Academic Honesty policy. More information can be found at

[http://www.usask.ca/honesty/pdf/dishonesty\\_info\\_sheet.pdf](http://www.usask.ca/honesty/pdf/dishonesty_info_sheet.pdf)