TIMG 5006
MANAGEMENT OF SOFTWARE ENGINEERING PROJECTS

Fall 2015
Department of Systems and Computer Engineering
Carleton University

Professor Weiss
SP
michael_weiss@carleton.ca

This course outline is a living document. Improvements may be made as necessary during the term. Version 0.3

Instructor availability

The instructor is available via e-mail any time. Office hours before class by appointment (online/offline).

Calendar description

TIMG 5006 [0.5 credit] Management of Software Engineering Projects
Models for the development of software. Software project management tools. Quality control. Risk assessment and management. Examples are drawn from the development of new technology products.

Precludes additional credit for TTMG 5006 (no longer offered).
Prerequisites: TIMG 5001 or TTMG 5001, TIMG 5002 or TTMG 5002

Course objectives

This course examines topics relevant to the management of software engineering projects. It emphasizes the perspective of the engineer or computer scientist in the role of technical project manager, responsible for planning and controlling the development activities that result in the delivery of software products. We examine theory, processes, methods, and tools. Although our scope includes well-established traditional practices, we are particularly interested in emerging practices, on-going research, and exploring the controversies within the field. The specific focus of the course will be on opportunity development, agile practices, and open source.

Rationale

This course is designed to build capability and knowledge in the management of large, complex and changing software systems. Students will learn about different perspectives on managing software projects, gain familiarity with the practitioner and research literature, and become proficient with practical managerial skills which can add value in their companies.

Benefits

This course prepares students to undertake thesis research or applied projects in the areas of software project management, process improvement, management of risk and quality, design and development of software products, and the management of operations within software-intensive technology companies.

Class Sessions

This course will be offered in-class and online. Remote students can participate by logging into conference room “TIMG 5006” with password “student” at present.carleton.ca. For the audio portion of the conference, you can call into the conference server using the phone numbers provided on login, or using the built-in VOIP feature (click on the headset icon). When using VOIP, you must use a headset.

Please see the tutorials on participating in an online classroom on the present.carleton.ca site.

For the weekly sessions there will be assigned readings and tasks.

During the student group presentation sessions, groups will be asked to make short presentations on their assignments (max. 10 minutes except where explicitly stated otherwise; please practice so you stay on time). Each group decides who presents what and in which order. Before 6 p.m. EST the day prior to when presentations are due, each group will distribute to all members of the class the slides to be presented the next day. No exceptions.

The course material and recordings of the class sessions will be made available on the Moodle learning content management system at http://cms.sce.carleton.ca.

Student Evaluation
Course participants are required to complete two group assignments and participate actively in class (discussion and assigned tasks). To determine the course grade, these weights apply:

• Assignment 1 (group)	20%
• Assignment 2 (group)	40%
• Class participation (individual)	10%
• Exam (individual)	30%

Assignments submitted late and presentations not made will receive a grade of zero. All students in a group receive the same grade. Final grade reports will follow Carleton University guidelines.

Assignment 1 (20%)
Groups of 3-4 people, or depending on number of people enrolled in the course. These can, but don't have to be the same groups as for assignment 2 (see note below). However, for practical reasons we cannot have more than 10 different topics. Groups need to be of similar size. Groups of 1-2 people are not allowed for this assignment. If there are too many groups, I will be ask smaller groups to merge into larger groups.

Document a pattern that addresses a software project management problem. A pattern is a reusable solution to a common problem. Patterns follow a specific format. Each pattern must discuss why this problem is a challenge, and present a solution to the problem from the literature and or personal experience. The pattern also needs to discuss the consequences of applying the solution, and describe known uses of the solution.

- Format: pattern to be posted to the course wiki (max 1500 words)
- Provide feedback on the first draft of all patterns by other groups: in-class workshop

A good reference on writing patterns is Bergin, J. (2013), *Writing Patterns*, Slant Flying Press. The so-called Alexandrian format used by Bergin (2013) and in the papers by Bergin is the recommended format.

Deadlines:

- Pattern topic due on **September 23**
- Workshop of first version (20 min per pattern) on **October 21**
- Final version due on **November 25**

Assignment 2 (40%)
Groups of 3-4 people, or depending on number of people enrolled in the course. As noted above, for practical reasons we cannot have more than 10 different topics. Groups need to be of similar size.

Design a card / board game for training a software project management skill:

- The game should focus on a specific skill, not cover a broad range of skills: eg a game on using Kanban for iteration planning is preferred to a game to train people on agile development in general
- Format: detailed requirements will be provided

Deadlines:

- Pitch game idea in class (3 min) on **September 30**
- Presentation of first version (10 min + 10 min for questions) on **November 4**
- Final presentation on last day of classes (10 min + 10 min for questions)

Class participation (10%)
Active class participation is an important component of this class:

1. Participation in class discussions (contribute to lessons learned at the end of each class, lead 4 discussions in class or online, provide feedback on the assignments of your classmates).
2. Post three key insights from one chapter of the Adrenaline Junkies book to the course wiki.

Take-home exam (30%)
The exam will handed out during the last class: **December 2**
The exam is due on: **December 9** at 6 pm, submit online

Group work and free loaders
Group work is an important component of this course. You may elect to work in the same group to prepare both assignments or work in two different groups. Group conflicts are to be dealt with by the group in a way that is fair, fast and without personal attacks. The instructor does not settle group disputes.

The instructor will dissolve a group that is late submitting an assignment. A group of three is expected to deliver better work than a group of two.

Free loaders are not welcome anywhere. This course is no exception. The best way to deal with free loaders is to not include their names in the first page of the group assignments. If a student's name does not appear in an assignment submitted by his or her group, the student must submit his or her own assignment. Failure to do so, the student will receive zero for the assignment. There is zero tolerance for free loaders.

Students with disabilities
Students with disabilities who require academic accommodations in this course are encouraged to contact the Paul Menton Centre (PMC) for Students with Disabilities to complete the necessary forms. After registering with the PMC, make an appointment with me in order to discuss your needs at least two weeks before the first assignment is due. This will allow for sufficient time to process your request

Plagiarism
Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offence that will not be tolerated. Please refer to the section on instructional offences in the Graduate Calendar for additional information. Plagiarism is against the TIM culture. A case of plagiarism will be referred to the Chair of the Department and the Carleton University Ethics Committee. The instructor will not deal with the matter directly. The university has clear processes to deal with students who are suspected of plagiarism.

Administrative details
These are the rules of conduct for this course:

- Please notify the instructor via e-mail, if you will not attend a class.
- **You must be prepared for each class.** You do so by reading the material assigned and being prepared to discuss in class how what was read can be applied in product development organizations.
- Each presenter must make his/her slides available to all other students by noon the day before.

Better Journals
Journal of Project Management
IEEE Transactions on Engineering Management
IEEE Software
ACM Communications
ACM Transactions on Software Engineering and Methodology
Empirical Software Engineering
Management Science

Management of Software Engineering Projects: Schedule

Date	Topic	Readings
Sep 2	Session 1: Introduction	Course outline
Sep 9	Session 2: Management challenges	Sauer & Reich (2008) Baskerville et al. (2003) Royce (2005) Noll et al. (2010)
Sep 16	Session 3: Understanding the customer I Discovering opportunities	Leonard & Rayport (1997) Lehtola et al. (2009) Weiss (2012a) Weiss (2012b)
Sep 23	Session 4: Understanding the customer II User-centered design Topic of assignment 1 due	Beyer et al. (2004) Gulliksen et al. (2003) Martin et al. (2009) TBD
Sep 30	Session 5: Planning and execution I Feedback and agility Pitch game idea for assignment 2	Cao & Ramesh (2007) Cao & Ramesh (2008) Sharp & Robinson (2004) Bergin (2006b)
Oct 7	Session 6: Planning and execution II Estimation and testing	Petter (2008) Molokken-Ostvold & Jorgensen (2005) Bergin (2006a) Erdogmus et al. (2005)
Oct 14	Session 7: Planning and execution III Managing uncertainty	Yap (2006) Erdogmus et al. (2006) Fichman (2005) Karlström & Runeson (2005) Hoda et al. (2009)
Oct 21	Session 8: Workshop of first version of assignment 1	
Oct 28	Fall break **No classes**	
Nov 4	Session 9: Presentation of first version of assignment 2	
Nov 11	Session 10: Architecture and organization	Coplien & Bjørnvig (2010) Harrison & Coplien (1996) Nan & Kumar (2013) TBD
Nov 18	Session 11: External contributors	Ebert (2010) Agerfalk et al. (2015)
Nov 25	Session 12: Reuse and software product lines Revised assignment 1 due	Clements et al. (2005) Kircher & Hofman (2012) Bosch (2009) Weiss (2011) Dittrich (2015)
Dec 2	Session 13: Final presentation of assignment 2	
Dec 9	Exam due at 6 pm	

Readings
The course is organized around the content of the following books, complemented by articles:

DeMarco T., Hrushka, P., Lister, T., McMenamin, S., Robertson, J., & Robertson, S. (2008), *Adrenaline Junkies and Template Zombies*, Dorset House Publishing.

Coplien, J. & Bjørnvig, G. (2010), Lean Architecture for Agile Software Development, Wiley.

Bergin, J. (2012), Agile Software: Patterns of Practice, Joe Bergin Software Tools.

Ebert, C. (2012), *Global Software: A Guide to Distributed Development, Projects, and Outsourcing*, IEEE/ Wiley.

Coplien & Bjørnvig (2010) and Ebert (2012) can be accessed as an e-book in the library. De Marco et al. (2008) can be purchased as an inexpensive ebook from Dorset House. Bergin (2012) is available as an ebook on Amazon. The content we cover in class is covered by earlier versions of the patterns published in a series of papers, so the book is not essential for the class, but since it is more comprehensive than the papers, it might be worthwhile.

Readings for Session 2
Sauer, C., & Reich, B. (2008), Rethinking IT project management: evidence of a new mindset and its implications, *International Journal of Project Management*, 27, 182-193.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., & Slaughter, S. (2003), Is Internet-speed software development different?, *IEEE Software*, 20(6), 70-77.

Royce, W. (2005), Successful software development style: steering and balance, *IEEE Software*, 22(5), 40-47.

Noll, J., Beecham, S., & Richardson, I. (2010), Global software development and collaboration: barriers and solutions, *ACM Inroads*, 1(3), 66-78.

Readings for Session 3
Leonard, D. & Rayport, J. (1997), Spark innovation through empathic design, *Harvard Business Review*, 75, Nov-Dec, 102-115.

Lehtola, L., Kauppinen, M., Vähäniitty, J., & Komssi, M. (2009), Linking business and requirements engineering: is solution planning a missing activity in software product companies?, *Requirements Engineering*, 14(2), 113-128.

Weiss (2012a), User frustrations as opportunities, *TIM Review*, April, http://timreview.ca/article/546

Weiss (2012b), Creating Customer Value Propositions for Technology Products, EuroPLoP

Readings for Session 4
Beyer, H., Holtzblatt, K., & Baker, L. (2004), An agile user-centered method: Rapid contextual design, *Extreme Programming and Agile Methods – XP/Agile Universe*, LNCS 3134, 527-554, Springer.

Gulliksen, J., Göramsson, B., Boivie, I., Blomkvist, S., Perrson, J., & Cajander, Å. (2003), Key principles for user-centred systems design, *Behaviour and Information Technology*, 22(6), 397-409.

Martin, A., Biddle, R., & Noble, J. (2009), XP customer practices: A grounded theory, *Agile Conference*, 33-40, IEEE.

TBD

Readings for Session 5
Cao, L., & Ramesh, B. (2007), Agile software development: Ad hoc practices or sound principles, *IT Professional*, 9(2), 41-47.

Cao, L., & Ramesh, B. (2008), Agile requirements engineering practices and challenges: an empirical study, *IEEE Software*, 25(1), 60-67.

Sharp, H. & Robinson, H. (2004), An ethnographic study of XP practice, *Empirical Software Engineering*, 9, 353–375.

Bergin, J. (2006b), Patterns for agile development practice, Part 3, *PLoP* (2006). [also Bergin (2012) TBD]

Readings for Session 6

Petter, S. (2008), Managing user expectations on software projects: lessons from the trenches, *International Journal of Project Management*, 26(7), 700-712.

Molokken-Ostvold, K. & Jorgensen, M. (2005), A comparison of software project overruns – flexible versus sequential development models, *IEEE Transactions of Software Engineering*, 31(9), 754-766.

Bergin, J. (2006a), Patterns for agile development practice, Part 2, *EuroPLoP* (2006). [also Bergin (2012) TBD]

Erdogmus, H., Morisio, M., & Torchiano, M. (2005), On the effectiveness of the test-first approach to programming, *IEEE Transactions of Software Engineering*, 31(1), 226-237.

Readings for Session 7

Yap, M. (2006), Value based extreme programming, *AGILE 2006 Conference*, IEEE. [also Bergin (2012)]

Erdogmus, H., Favaro, J., & Halling, M. (2006), Valuation of software initiatives under uncertainty: Concepts, issues, and techniques, in: Biffl, S. et al. (eds.), Value-Based Software Engineering, Springer, 39-66.

Fichman, R., Keil, M., & Tiwana, A. (2005), Beyond valuation: "Options thinking" in IT project managmement, *California Management Review*, 47(2), 74-96.

Karlström, D., & Runeson, P. (2005), Combining agile methods with stage-gate project management, *IEEE Software*, May/June, 43-49.

Hoda, R., Noble, J., & Marshall, S. (2009), Negotiating contracts for agile projects: A practical perspective, *Agile Processes in Software Engineering and Extreme Programming*, LNBIP 31, 186-191.

Readings for Session 10

Coplien, J. & Bjørnvig, G. (2010), Chapters 3 and 5.

Harrison, N., & Coplien, J. (1996), Patterns of productive software organizations, Bell Labs Technical Journal, 1(1), 138-145.

Nan, N., & Kumar, S. (2013), Joint effect of team structure and software architecture in open source software development, *IEEE Transactions on Engineering Management*, 592-603.

TBD

Readings for Session 11

Ebert, C. (2010), Chapters TBD

Agerfalk, P., Fitzgerald, B., & Stol, K. (2015), *Software Sourcing in the Age of Open: Leveraging the Unknown Workforce*, Springer Briefs in Computer Science, Springer [available online in the library]

Readings for Session 12

Schuetz, D. (2010), Transition to product line engineering, *EuroPLoP*, ACM, 23.

Kircher, M., & Hofman, P. (2012), Combining systematic reuse with agile development – experience report, *Conference on Software Product Line Engineering*, ACM, 215-219.

Bosch, J. (2009), From software product lines to software ecosystems, *International Software Produce Line Conference*, ACM, 111-119.

Weiss, M. (2011), Economics of collectives, *International Workshop on Quantitative Methods in Software Product Line Engineering*, ACM, 39.

Suggested Books

Many of these are available online (on Safari via the Carleton library, or on the Web).

Adzic, G. (2009), *Bridging the Communication Gap*, Neuri Limited.

Alexander, I., & Beus-Dukic, L. (2009), *Discovering Requirements*, Wiley

Cho, A. (2011), *The Jazz Process: Collaboration, Innovation, and Agility*, Addison-Wesley.

Elssamiadisy, A. (2008), *Agile Adoption Patterns: A Roadmap to Organizational Success*, Addison Wesley.

Harrison, N., & Coplien, J. (2006), *Organizational Patterns of Agile Software Development*, Addison Wesley.

Highsmith, J. (2004), *Agile Project Management*, Addison Wesley.

Hohmann, L. (2003), *Beyond Software Architecture: Creating and Sustaining Winning Solutions*, Addison Wesley.

Lopp, M. (2007), *Managing Humans*, Apress/Springer.

Meyer, M. (2007), *The Fast Path to Corporate Growth*, Oxford University Press.

Poppendieck, M., & Poppendieck, T., *Implementing Lean Software Development*, Addison Wesley.

Richardson, J., & Gwaltney, W. (2007), *Ship It!*, The Pragmatic Bookshelf.

Rosenberg, S. (2007), *Dreaming in CODE*, Crown.

Royce, W., Bittner, K., & Perrow, M. (2006), *The Economics of Iterative Software Development*, Addison-Wesley.

Ullman, E. (2012), *The Bug: A Novel*, Picador (originally published in 1984)