



Department of Computer Science and Software Engineering

Course Portfolio for CSSE 372: Software Project Management

Author/Instructor: Shawn A. Bohner

Software Project Management is a course rarely taught at the undergraduate level. Undergraduate software engineering curriculums are often full of the technical courses, leaving little room for some of the key management science issues. Further, undergraduate students often view "management" as a task that they will probably not face until they have been engineers for a few years. A savvy instructor debunks this myth early and soundly before proceeding to engage students in what will become one of the most important courses in their scholastic experience.

This course portfolio is intended to equip the software project management instructor¹ with the means to design and execute an effective software project management learning experience for software engineering students. The intended target are students in their junior year with at least a basic software engineering course such as Introduction to Software Engineering, or a course that outlines the software engineering life cycle such as Software Requirements Analysis and Specification. Software development and/or maintenance experience, and an ability write and communicate effectively make this course more meaningful.

The purpose of this course portfolio is to provide a living document of the Software Project Management course (CSSE 372) as it develops and evolves. It provides a means for the instructor(s) to learn from and reflect on experiences teaching software project management to undergraduate students.

Level	3 rd year, Junior Year
Design and Implementation Language/Model	Programming language and design method agnostic
Typical number of Students enrolled	15-35
Fraction of full-time student effort/year	1/12 th
Number of weeks	10, plus finals
Contact hours per week	4+, during 4 one-hour sessions
Pre-requisite Knowledge	Introduction to Software Engineering or Software Requirements course.
Lab hours per week	None
Description of Labs	None

Table 1: Summary of Course Characteristics

¹ Instructors at Rose-Hulman Institute of Technology initially, but it is expected to expand to the software engineering teaching community at large as more experience and confidence is gained. This course has been taught once and is planned to be taught again in Fall 2010.

The purpose of the CSSE 372 course is to introduce the student to fundamental project management principles and augment them with key practices that support the everydayness of software change. Students examine the basics of project management in the context of software projects and discover key elements of planning, organizing, staffing, directing, and controlling through developing their own software project plans and forming teams.

Teaching Philosophy: Software project management for undergraduate students is typically a foreign concept and unlike many engineering, scientific, or programming-oriented courses, it requires careful introduction of what is often considered "soft" or imprecise science. That is, many management concepts can sound more like heuristics or common sense, rather than the tangible concepts practiced in engineering. Hence, a key objective is to make these often-misunderstood concepts real-world, worthy of practice, and fun.

To this end, the teaching philosophy for software project management must be to introduce management concepts as mechanisms for making student's work lives more predictable and time efficient. For example, encourage students to take all the tasks from their courses during the term and create a schedule for them. Take them from what they know (design and programming) to "how to" design and program (plan) a successful project; then to exercise the plan over a number of canonical situations. An interrogative style of discourse coupled with a great deal of practicum is key. Active learning examples keep the students engaged in what otherwise would be considered a task intense course with no purpose.

At the undergraduate level, the emphasis is on the foundational principles - the basis of what students need for a successful career in industry and/or graduate studies. Software engineering is especially well-suited for a blend of theory and practicum. In the CSSE Software Project Management course, endeavor to instill fundamental principles of management in the nascent software engineering practice. This is accomplished with a blend of discussions, active learning exercises, and discovery experiences planning and executing a real-world project with actual clients. Management are often best learnt in the context of a team exercise producing software - the fellow students are tangible teachers too. The lessons are much more meaningful when confronted with real issues - even in a safe environment like a class project over a term. These experiences give credence to the other topics that do not lend themselves to the classroom experiences.

Context: In the Bachelors of Science in Software Engineering (BSSE), the Software Project Management course provides an early place in the sequence of program courses. Prior to the junior year, our students take largely the same courses as our Computer Science students, with the exception of Software Quality Assurance (CSSE 376), which is introduced the last term of the sophomore year. The Software Project Management course only has one prerequisite, Data Structures and Algorithm Analysis, a sophomore programming course. The following is a detailed layout of our Software Engineering program. The figure below depicts the sequence of our BSSE courses.

	Fall	Winter	Spring
Soph. Year			376
Junior Year	371 372	374	375 373
	377 497	498	499

Legend:

- CSSE - Abbreviated Dept. Name
- 371 – Software Requirements Analysis and Specification
- 372 – Software Project Management
- 373 – Formal Methods
- 374 – Software Arch & Design 1
- 375 – Software Construction & Evolution
- 376 – Software Quality Assurance
- 377 – Software Arch & Design 2

Freshman Year

Fall Term

CSSE	120 Introduction to Software Development...	4
MA	111 Calculus I	5
PH	111 Physics I.....	4
RH	131 Rhetoric & Composition	4
CLSK	100 College and Life Skills	1

Sophomore Year

Fall Term

	<i>Credit</i>		<i>Credit</i>
CHEM		105 Engineering Chemistry	4
CSSE		232 Computer Architecture I ...	4
MA		221 Differential Equations and Matrix Algebra I.....	4
MA		275 Discrete & Combinatorial Algebra I.....	4
			16

Winter Term

CSSE	220 Object-Oriented Software Development.....	4
MA	112 Calculus II.....	5
PH	112 Physics II.....	4
HSS	Elective.....	4

Winter Term

	<i>Credit</i>		<i>Credit</i>
CSSE		230 Data Structures and Algorithm Analysis.....	4
CSSE		333 Database Systems	4
MA		375 Discrete & Combinatorial Algebra II	4
		Domain track course.....	4
			16

Spring Term

ECE	130 Introduction to Logic Design.....	4
MA	113 Calculus III.....	5
HSS	Elective.....	4
Science	Elective.....	4

Credit

	<i>Credit</i>		<i>Credit</i>
CSSE		304 Programming Lang. Con.	4
MA		Elective.....	4
CSSE		376 SW Quality Assurance	4
RH		330 Technical Communication	4
			16

Junior Year

Fall Term		Credit
CSSE	371 Software Requirements & Specification	4
CSSE	372 Software Project Mgmt	4
MA	381 Introduction to Probability with Statistical Applications Domain track course.....	4
		16

Winter Term		Credit
HSS	Elective	4
CSSE	374 Software Arch. and Des.....	4
CSSE	332 Operating Systems.....	4
		16

Spring Term		Credit
CSSE	373 Formal Methods....	4
CSSE	375 SW Construction & Evolution Elective.....	4
HSS	Domain track course or free elective.....	4
		16

Senior Year

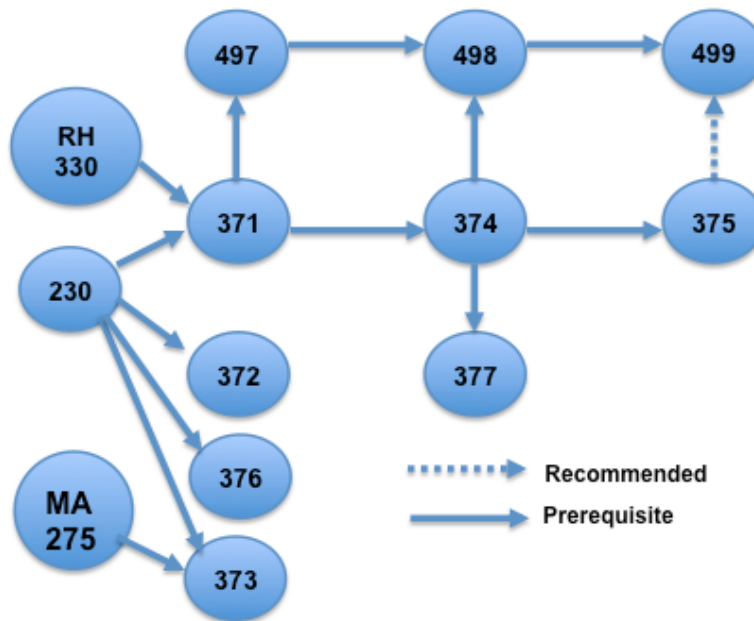
Fall Term		Credit
CSSE	377 Software Architecture and Design II	4
CSSE	497 Senior Project I	4
HSS	Elective.....	4
		16

Winter Term		Credit
CSSE	498 Senior Project II.....	4
CSSE	Elective.....	4
HSS	Elective.....	4
Free	Elective	4
		16

Spring Term		Credit
CSSE	499 Senior Project III	4
HSS	Elective	4
Free	Elective	4
		12

Total credits required: 192

The following is the prerequisite tree for the BSSE program. Note: CSSE 371 and 372 also require junior standing.



Course Design

Overview: Managing software projects entails fundamental principles from mainstream project management plus key practices that support the everydayness of software change. For the Software Engineering program, this course provides context for many of the other software engineering courses showing the integration of methods and tools to control software. Students examine the basics of project management (planning, organizing, staffing, directing, and controlling) in the context of software projects. Through the text, a series of directed readings, active learning exercises, and case studies, students will be exposed to a blend of classic and emergent techniques to grasp a thorough understanding of key software project management principles. Topics to be covered in this course include:

1. Basic principles of project management
2. Software life cycle processes
3. Software estimation
4. Forming and managing Project Team
5. Software risk planning and management
6. Software project planning
7. Managing software projects to a plan
8. Adv. Topics: Earned Value, Critical Chain

Learning Outcomes: Upon successfully completing this course, a student should be able to.

1. Explain fundamental elements of Software Project Management
2. Explain and employ contemporary software life cycle processes, activities, and work products
3. Estimate software project effort, cost, and schedule for an intermediate size project
4. Identify, analyze, and manage software project risks
5. Create and maintain a software project schedule
6. Create a plan for an intermediate size software project and manage to the plan as project evolves
7. Formulate software project teams in terms of roles and responsibilities
8. Plan, organize and conduct effective meetings

Please note that these objectives provide an essential set of skill targets, but by no means is comprehensive. To truly master software project management, experience is needed before advancing to the more challenging topics or even the essential topics at deeper levels of Bloom's taxonomy of understanding.

Prerequisites: CSSE 230 (Software Development), RH 330 (Technical Comm.), and Junior standing; co-requisite with CSSE 371 (Software Requirements Analysis and Specification).

Textbooks: Students often need information sources that guide the learning content of the course. Some are traditional texts with fully developed exercises and the like, while others are specific treatises that provide advanced concepts or references. Given the this is an introductory course on software project management, a very pragmatic text was selected from the vast corpus of project management texts. This is typically augmented with papers for more specific treatments of topics not well addressed in the text. For example, the following was recently used:

- "Applied Software Project Management" by Andrew Stellman and Jennifer Greene; O'Reilly Media (2005). ISBN-10: 0596009488
- Readings will be also be assigned from relevant papers (e.g., case studies).

Grading:

Examination(s)	30%	Quizzes	5%
Homework/Case Studies	25%	Final Project Plan	30%
Presentation/Participation	10%		

Expectations: Students are expected to attend and participate in class. Students are required to use the CSSE372 course website to obtain relevant information, and interact with instructor and other students. Announcements and assignments are conveyed via Rose-Hulman email addresses and/or posted on the website. Students are expected to work on some assignments with other team members.

Late Submissions: Please note that homework and project deliverables are due at the specified time on the specified day. Late quizzes will not be accepted. Homework assignments, and project deliverables are also not accepted late, with the following exception: Students have three "late day" credits. They may use one of them on any homework or project assignment, which will allow them to submit that assignment up to 24 hours after the due time. Homework's or project assignments, which are more than 24 hours late, will receive a deduction of at least 10% per late day (or not be accepted at all), depending on the circumstances and the degree of lateness. Students may earn a maximum of one additional "late day" by submitting an assignment or a project deliverable 24 hours before the due date. Students must email the instructor alerting him to the same to obtain the "late day" credit. If the student submit something late for which late day credits are allowed, the instructor will assume that the student wants to use one of his/her late day credits unless instructed otherwise.

Academic Integrity: CSSE Honesty Policy governs class and performance. Joint study is allowed (even encouraged) on some items as expressed by the instructor; however, each student must produce his/her solutions individually. Students must not collaborate on tests or homework that is passed in unless directed by the instructor.

Attendance Policy: Attendance is mandatory (unless with a legitimate excused absence such as illness). If a student cannot make it to class or lab, they are still responsible for all materials covered in class as well as all announcements. Up to 2 unexcused absences are permitted. In accordance with the Rose-Hulman attendance policy, additional unexcused absences may result in you receiving a failing grade for the course.

Laptop Policy: Students may need to use their laptops during some portion of the class period. Hence, they must bring their laptop, a power cord, and a network cable to class. During class discussion, laptops are to remain closed. Laptop use during discussions can be distracting to other students, the instructor, and may also keep the student from focusing on the material.

Writing: Written communication is important in CSSE 372, as it is in the software profession in general. Remember that a software document has several unique and important characteristics:

1. Technical documents are often the result of group authorship, thus it requires planning and final tweaking.
2. Specificity and organization are more important than flow; hence technical documents are often ordered around lists and tables rather than paragraphs.
3. Documents are often the reader's only source of information on the particular subject or product; hence they must be thorough and complete.
4. Documents are often used to answer specific questions; hence, they should facilitate finding specific pieces of information (navigation).
5. Documentation must bridge from general specifications to particulars of implementation and operation, hence it must make abstract concepts concrete and make concrete facts fit generalized concepts.
6. Documentation can be presented in many forms: online via HTML, MS help files, just plain text, and on paper as reference manuals, tutorial, quick reference guides, etc. It is important to choose the correct medium and even more important to write to fit the medium.

Students are encouraged to drop by the instructor's office when they have questions about their documents. Students are also aware of services provided by the Learning Center.

Schedule of Classes: This is an example schedule of classes for CSSE 372 Software Project Management.

Week /Day	Date	Reading for Class	Homework & Project Assignments	Due Date	Topic Content & Notes
0/1	9/2		Quiz0: Survey (on Angel)	9/3	Intro. to CSSE 372. Software Proj. Mgt
0/2	9/3	Preface & Chap-1	HW1: Preliminary Task Schedule for Classes	9/7	Anatomy of a Software Project
1/1	9/6	SW Project Failures paper			SW Failures
1/2	9/7		Case Study 1: SW Project Failure	9/7	SW Project Failures
1/3	9/9	Hard Choices Game Rules			Hard Choices and SW Project Business Case
1/4	9/10	Business Case Paper	Case Study 2: Business Case	9/10	Business Cases
2/1	9/13	Chap-2			Anatomy of a SW Project Plan
2/2	9/14	Software Process Paper	HW2: SimSE Game Exercise	9/21	SW Processes Introduction
2/3	9/16	SimSE Intro.			SimSE Demonstration and Exercise
2/4	9/17		Download, install, & play with COCOMOII.		SW Process Development
3/1	9/20	Chapt-3	Assign Teams		SW Estimation Fundamentals
3/2	9/21		HW3: Estimation with COCOMO II	9/28	Estimating with COCOMO-II
3/3	9/23	Estimation Paper	Case Study 3: Estimation Approaches	9/23	More on Estimation
3/4	9/24				SW Risk Analysis
4/1	9/27	Software Risks Paper			SW Risk Management
4/2	9/28	Chapt-4	HW4: Software Risk Table and Sheets	10/5	Work Breakdown Structures (WBS)
4/3	9/30				1 st Exam Review
4/4	10/1				1 st Examination
5/1	10/4	SW Project Scheduling Paper			Exam 1 Results and Project Scheduling
5/2	10/5		HW5: SW Project Schedule	10/12	Critical Path Method
5/3	10/7	Critical Chain Paper			Critical Chain Method
5/4	10/8				Monitoring and Controlling SW Projects
6/1	10/11	EVA Paper			Earned Value Analysis
6/2	10/12		HW6: Earned Value Analysis	10/19	Earned Value Exercise
Fall Break October 14th and 15th					

Week /Day	Date	Reading for Next Class	Homework & Project Assignments	Due Date	Topic Content & Notes
7/1	10/18	Chapt-9	Final Project Assigned	11/5	Human Side of Project Management
7/2	10/19	Conflict Mgt. paper			Conflict Management
7/3	10/21	Chap 10			Effective SW Project Communication
7/4	10/22				Speaker 1
8/1	10/25				Managing with Measures
8/2	10/26	Agile Project Paper			Agile Project Management
8/3	10/28				Program Mgt. and SW Project Portfolios
8/4	10/29				Speaker 2
9/1	11/1				Team Project Lab
9/2	11/2				Team Project Lab
9/3	11/4				Team Project Lab
9/4	11/5		HW7: Review another Team's SPMP	11/9	Speaker 3 <i>Final Project Plan Due</i>
Week /Day	Date	Reading for Next Class	Homework & Project Assignments	Due Date	Topic Content & Notes
10/1	11/8				Team Project Lab
10/2	11/9				Day 1: Teams Present Final Project Plan
10/3	11/11				Day 2: Teams Present Final Project Plan
10/4	11/12				Final Exam Review
11/1	11/15				Final Exam in Room 11

This schedule of classes for CSSE 372 Software Project Management will also be posted on the course website with links to the requisite readings, case studies, homework assignments, and other resources.

Note that the early part of the course entails specific vernacular and motivational material. If the student knows that the material is important to avoid many of the risks of project failure, there is a motivation to pursue studying and understanding the material at a deeper level. The vernacular for project management is often foreign to engineering students, so considerable time is dedicated to ensuring that they are familiar and comfortable with the essential facets of scope, planning, risks, estimations, and scheduling. As the course progresses, these topics are explored in detail with in-class exercises and specific homeworks designed to bring a level of confidence to the students.

This course is coordinated with the CSSE 371 (Requirements Analysis and Specification) course in the Junior Sequence of courses that act as a guide through a project where the students develop a software product for a real customer. CSSE 371 and 372 start the junior sequence in the fall term, then CSSE 374 (Software Architecture and Design 1) in

the winter term, and finally CSSE 375 (Software Construction and Evolution) completes the sequence in the spring term with a delivery to the client of an operational software product.

In the seventh week of the fall term, the CSSE 372 project teams are formed with 4 students each and the final project assignment is given. This is to develop a software project management plan (SPMP) for the Junior Project. The students are expected to take all that they have learned and apply it to a substantial software project planning effort.

Assignments: Homework or project assignments are assigned regularly (typically one a week until the project is assigned in the 7th week). Unless otherwise requested, students must post these on course website in the associated drop boxes.

Homework and projects are necessary instruments for tracking progress of students. A typical student will work approximately 7-9 hours outside of lectures each week on this course (depending on background). This is a demanding course covering a great deal of material - students must avoid falling behind on the assignments. While this course is demanding, it is also rewarding for those that want strong understanding of software engineering as a discipline. Appendix B provides an example of a typical CSSE 372 homework assignment. Appendix C provides an example of a final project assignment complete with the templates and examples.

Case Studies: Case studies are discussed in class. Students must read the case study, prepare (and submit) a brief write-up of their understanding and opinion of the case study prior to the class. The write-up should be no more than 1/2 page in length and is collected in class. The discussion in class must have participation from each of the students as it counts towards their grade. If students have an excused absence from class, this can be sent to the instructor via email the day of the class. Appendix D provides an example of a Case Study assignment.

Examinations: There are two examinations that test the student's absorption of the material from this course. While exams are imperfect vehicles for judging a student's progress, they do serve as a measurement milestone and benchmark for the instructor. Appendix E illustrates a typical examination for CSSE 372. Note the breadth and depth questions to demonstrate the respective knowledge from the course objectives.

Quizzes: Daily Quizzes are a "test first" technique used to engage students each day in the topics covered in class. These are developed from the course and lesson objectives. They are designed to focus the student on the material covered during class and offer a good study medium for later examinations. The following is an example of the quizzes given each day in class.

CSSE 372 – Software Project Management → Quiz 7

1. For an informal Business Case, what are the key elements needed and how detailed do they need to be?
2. What are the major elements of a software project plan?
3. What are the four “P’s” of a software project?
4. What are the five basic Software Planning Steps?
5. How can you ensure that what your clients want is what they need?
6. What does planning do for a software project?
7. When would you be less inclined to use a software project planning software package?
8. Were most of the concepts covered today clear? If not, what is the “muddiest point” we covered today?

Daily Lesson materials: Each day, students have slides of the lesson material provided via the course website. These usually contain from 15-25 slides that are presented in a 50 minute session along with any active learning exercises, case studies, homework reviews, and the like. The slides have relevant material for the lesson learning objectives, which are derived from the course learning objectives. The objectives are couched in Bloom Taxonomy vernacular to ensure an appropriate level of learning expectations for the material. Appendix F provides an example of a day's lesson slides.

Course Evaluation and Feedback: Students are free to provide feedback about the course at any time. If they feel uncomfortable talking with the instructor directly, there is an anonymous feedback box under the course website where they can provide feedback throughout the term; It is checked once a day and will have response to feedback in a reasonable time. There will also be two anonymous "plus-delta evaluations" of the course where students can offer suggestions on how to improve the course and its delivery. The following are examples of the questions asked in the plus delta evaluations.

Plus/Delta Course Evaluation

1. Please rate the pace of the lecture portion of the class:
 - 1) Much too fast
 - 2) Somewhat too fast
 - 3) Somewhat too slow
 - 4) Much too slow

2. What things are working well for your learning in the lecture portion of the class?

3. What specific things could I do differently to improve your learning during the lecture portion of the class?

4. Please choose the rating that best reflects how helpful the daily quizzes are to your learning:
 - 1) Very helpful
 - 2) Somewhat helpful
 - 3) Somewhat unhelpful
 - 4) Very unhelpful

5. What things are working well with regard to the daily quizzes?

6. What specific things could I do to improve the daily quizzes?

7. Please choose the rating that best reflects how much of the assigned reading you have done:

- 1) All of it
- 2) Most of it
- 3) Little of it
- 4) None of it

8. Please choose the rating that best reflects the difficulty of the homework, not including the project milestones:

- 1) Much too difficult
- 2) A bit too difficult
- 3) A bit too easy
- 4) Much too easy

9. Please choose the rating that best reflects how helpful the homework (not including the project milestones) has been to your learning:

- 1) Very helpful
- 2) Somewhat helpful
- 3) Somewhat unhelpful
- 4) Very unhelpful

10. What things are working well with regard to the homework?

11. What specific things could I do to improve the homework?

12. Please choose the rating that best reflects your work load in this course relative to other courses you've taken at Rose-Hulman:

- 1) Much higher than average
- 2) Somewhat higher than average
- 3) Somewhat lower than average
- 4) Much lower than average

13. You may use this space for any other comments or suggestions that you have regarding the course. Thank you very much for taking the time to complete this survey.