

OMSE 551: Strategic Software Engineering

This template includes placeholders for material that will become available throughout the Quarter but are not necessarily there now.

Catalog Description: Where traditional software engineering focuses on the development and maintenance of individual systems strategic software engineering addresses the development of multiple systems over time. Significant gains in productivity, cost, and schedule can result from systematic improvement of the software development process and systematic reuse of life-cycle products over multiple developments. This course covers the principles, methods, and tools for strategic software development including process modeling and improvement, developing programs as of families of systems, and systematic approaches to code generation and the reuse of non-code products including requirements, documentation, and design.

Objectives: Students will understand the principles of strategic software engineering and be able to apply them to adapt their software development processes and technologies to improve quality, reduce costs, and meet changing business needs. Students shall:

- Understand business case models for strategic software development and how to analyze a particular case.
- Understand principles and techniques for formal process modeling and continuous process improvement.
- Understand basic economics of developing a program family and be able to construct a simple business case.
- Demonstrate the ability to apply a domain analysis and engineering process to develop families of software systems.
- Demonstrate the ability to adapt a formal process model to achieve long term technical and business goals.

Required Text:

Software Product-Line Engineering: a Family-Based Software Development Process, David M. Weiss, Chi Tau Robert Lai, Addison Wesley, 1999, ISBN 0-201-69438-7

Optional Text:

Software Product Lines: Practices and Patterns, Paul C. Clements and Linda M Northrop, Addison-Wesley, 2001, ISBN 0201703327

In the past, we have used the Clements/Northrop book as a text. The most up-to-date version of this book is now on line, superseding the published version. Students may still wish to have the hard copy text but can complete the reading assignments from the online SEI site (<http://www.sei.cmu.edu/productlines/framework.html>) without buying the book.

Reading and Assignments

The reading assignment for each week should be completed before class.

We will use the D2L "Discussion" tool for inter-class discussions and to air questions about assignments. You should try to compose your answer to discussion questions before reading other students posts. You are required to read the other posts by students in your group and make a substantive response to at least two of them. If the class is large, I will divide it into groups of 10 or fewer to limit the amount of reading required.

Any other written assignment for a week will be due following week before class and should be emailed directly to me unless the directions for the assignment state otherwise.

Syllabus

Week 1: Introduction to Strategic Software Engineering

Overview of course content, and grading. Introduction to the Strategic Software Engineering view of software development. Root causes of the ongoing "software crisis." Inefficiencies inherent in a conventional (sequential) software development paradigm. Viewing a company's overall software development process: key concepts and objectives of process modeling and continuous process improvement. What it means to take a "product-line" view of software development. Overview of product line development strategies and the resulting efficiencies of scale.

- Nature of the "software crisis"
- Inefficiencies of sequential development
- A strategic view of software development
- The Celsius Tech case study

Reading:

Reread ([download](#)) Brooks, F. "No Silver Bullet"

Reread the chapter on Celsius Tech (Ch. 16 in 1st edition, Ch. 15 in 2nd) in *Software Architecture in Practice* (from OMSE 532: Software Architecture). If you do not have a copy of the book, you can read the original report on which

the chapter is based ([download](#)).

([download](#)) Gibbs, W. T., "Software's Chronic Crisis"

Resource: [([download](#)) A Case Study in Successful Product Line Development, SEI Technical Report. This is the original report on Celsius Tech. It provides more detail than the architecture text, particularly on the organizational and management aspects.

Assignment: On-line [discussion](#) of how things changed for Celsius Tech to take a product-line development approach. Compose and post your own views, then discuss the posts with classmates:

- The development process
- The major products of development
- The company organization

Week 2: Process Modeling and Process Improvement I

Review of concepts of software engineering processes and process improvement. The process-improvement process. Overview of process specification: why we specify processes, qualities of a good process spec, contents of a good process spec. Process abstraction and modeling: discussion of modeling approaches. Concepts of process modeling

- The process modeling process
- Process improvement vs. product improvement
- Discussion of CMM
- Discussion of assignment
- Process specification and modeling

Readings:

[Kirby 90] Kirby, J. Jr., Lai, R.C.T., Weiss, D.M.; a Formalization of a Design Process, Proc. 1990 Pacific Northwest Software Quality Conf., Oct. 1990, pp. 93-114

(Review) [Parnas 85b] Parnas, D.L., Clements, "A Rational Design Process: How and Why to Fake It," IEEE TSE, SE-12, pp. 251-257, March 1985.

[Paulk 98] Paulk, et al, "The Capability Maturity Model for Software"

Assignment: On-line discussion of process development relative to your organization.

Week 3: Process Modeling and Process Improvement II

Process specification and modeling. Techniques for modeling software processes.

Developing and using software process models in the organizational context: enactment, analysis, testing, baselining, and maintaining. Discussion of modeling methods in the literature. Introduction to a process modeling technique. Viewing software process models as a product of software engineering.

- Application of process models
- Discussion of modeling methods

Lecture

Link to video [here](#).

Readings:

Scrum Process Model: Scrum is a tiny process that will illustrate some of the basic elements of process specification in the Eclipse framework. Here is the [online Scrum specification](#). If needed, you can find a brief overview of Scrum in Wikipedia [here](#).

Distributed Collaboration Process: This is a set of process specifications developed as part of an OMSE practicum using the same tool we will use for our exercise. The approach is based on the concepts learned in this course and shows what can be done with quite minimal tooling. Note that there are a set of common process components defining Roles, Tasks, and Products. These common components are then used in the creation of different processes (a family of processes, more accurately). You can find a more complete overview of the project [here](#). You should read the introduction and study a few of the process definitions to get a sense of how they are defined. Note that you can also download the original definitions of the processes from the [here](#) if you want to see how they are constructed using the templates.

Assignment:

For this week's assignment, use the OMSE 556 process definition tool to create a process specification. Build a model of the "Rational Design Process" described in [Parnas 85b]. I am also OK if you want to do something like translate the Scrum process to the tool. More detailed instructions are given here for accessing the templates.

I expect you to consult with one another to get the tools set up and create your specifications. I have created threads in the Week 3 Discussion section to support this.

Week 4: Product Line Development Overview

Overview of product line development processes. Software engineering foundations for developing program families including key principles – role of modularization, information hiding, and abstraction. Understanding and using the FAST PAST process model. An example development including work products.

- In class presentation of process models and discussion of lessons learned
- Foundations for product-line development: key assumptions and principles

Readings:

Weiss/Lai text Chapter 1, 2, 3

Clements/Northrop Chapter 1-3 (SEI Framework [Introduction](#) and [PL Essential Activities](#))

[Parnas 76] "On the Design and Development of Program Families"

Assignment: On-line discussion of your experience with program families.

Week 5: The FAST Product Line Development Process

Overview of the FAST process: Economic Modeling, Application Engineering and Domain Engineering. Economics of product-line development using FAST. Understanding and using the FAST PASTA formal process model provided in the text.

- Domain analysis
- Domain engineering
- Application Engineering

Readings

Weiss/Lai text Chapter 4, 5, 8 (skim 6-7 if needed to understand the PASTA process model)

Clements/Northrop Chapter 4 (SEI [Software Eng. Practice Areas](#))

Assignment

Take home midterm covering weeks 1-5. Download here and email directly to me when finished.

Week 6: Domain Analysis and Domain Engineering

Products, methods and principles of Domain Analysis. The Commonalty Analysis process and products: how to determine the scope of a family of systems, how to determine and define the common requirements of the family, how to determine and define the variabilities that distinguish family members. Economic modeling:

assessing and modeling the return on investment from developing software as product lines.

Readings:

[DeBaud 99] "A Systematic Approach to Derive the Scope of Product Line"

[Bockle 04] "Calculating ROI for Software Product Lines"

Clements/Northrop Chapter 5 (SEI [Technical Management Practice Areas](#))

Assignment:

1) **Project proposals: submit proposal for term project.** Read the section on the term project below.

2) Here are the assignment instructions. We will revise the Commonalty Analysis document for next week then the Module Guide for Week 8. If you are interested in modifying the code, a complete package of the FWS code from Weiss is in the file FWS.tar with some general instructions on setting up the environment.

Week 7: Application Engineering and Domain Specific Languages

Formal model of the domain engineering process. Development of the application engineering environment. Designing and developing product-line architectures. Generation technologies for code, documentation and other work products. Introduction to composition-based and compiler-based approaches to product generation.

- Walkthrough of the FWS example generation technology
- Presentation/Discussion of Commonalty Analysis part of FWSOS
- Code generation, domain modeling and domain specific languages

Readings:

Assignment:

Complete part 2 of the FWSOS project (revision of the module guide and uses hierarchy). The link to the module guide is given in Week 6. Here is a common version of the FWSOS commonality analysis to use.

Week 8: Emerging Technologies and the Business Context

Evolution of product lines: moving toward adaptable code generation. Organizational and business issues in applying strategic software engineering. Structuring an organization to develop software processes and product lines – key roles and responsibilities. How functional units in the organization must relate and interact to ensure alignment of strategic business goals and technical development decisions

Readings:

[Hammond] "Improving productivity and quality with domain-specific modeling"

[Metaedit] Domain specific modeling compared to UML

[Bosch] Case study on car periphery system product line at Bosch

Clements/Northrop Chapter 6 (SEI [Organization Management Practice Area](#)) particularly the sections on Organizational Planning, Organizational Risk Management, and Structuring the Organization

Links to a number of articles and other material on domain modeling [here](#)

Resources:

[Metaedit+](#) is a commercial tool for building domain specific languages and supporting code generators (i.e., it is a form of application-generator-generator). You can [download](#) an evaluation copy and run through a couple of their tutorials if you want to see what one looks like.

Assignment:

Work on class projects.

Week 9: Thanksgiving

Thanksgiving break. No class, work on class projects. WebCT discussion of any issues.

Week 10: Project Presentations

Presentation and critique of student projects. See the following section on the Project/Term Paper. Students present the findings and lessons learned from their independent work to the class.

Assignment: Hand in written term projects, Eluminate presentations.

Week 11: Project Presentations

There is no final exam. We will use finals week to conclude presentations if there is not sufficient time in Week 10 for all of the presentations.

Project/Term Paper

Every student is required to do a project or term paper and present findings and results to the class. The goal of the project is to gain hands-on experience applying some of the concepts and material from the course to a real problem, preferably from your own work experience. You should scope the project so that it adequately demonstrates the ability to apply class material but can be completed by Week 10. The project consists of three parts:

1. *Proposal*: develop a written proposal for your project. The proposal should describe a) what you plan to do, b) which aspect from the course you will apply and how, and c) what products you plan to produce. Usually a page is sufficient. You can see an example of such proposals from previous years: Borhn, and Dukart.

2. *Written report or other products*: a written report and possibly other products (e.g., adaptable code, a commonality analysis document, an economic analysis, etc.). The usual scope of these is roughly ten pages or so depending on the subject. Here are a couple of examples: Borhn, and Dukart.

3. *Presentation*: prepare a presentation (e.g. PowerPoint) on your project for class. The goal of the presentation should be to convey what you learned from your project to your classmates. Your presentation should not run longer than 20 minutes so everyone will have a chance to speak. Here are a couple of examples: Borhn, and Dukar

I am pretty open to suggestions for topics as long as you can convince me that you will really learn something in depth related to Strategic Software Engineering. For example you might do a commonality analysis of a product at work or do an in-depth study of available product-line case studies. I have posted a couple of examples of project reports and presentations from last year under WebCT/Course Content that you can look at.

Due dates: Proposal: Week 7
 Project/Term Paper: Week 10
 Presentation: Week 10 (continues to Week 11 if necessary)

Resource Material

I. Links of Interest

Grady Campbell, Prosperity Heights Software: Grady Campbell invented the first well-defined software product line process based on earlier work on the A-7E project (with David Parnas and yours truly). This evolved into the Synthesis process at SPC. Because Synthesis was considered proprietary to SPC's member companies, his seminal contributions are not widely known. He gives a brief history of its origins under "Background." [The web site](#) gives a rich overview of the many facets of product-line engineering including Domain-Specific Software Engineering, Reuse-Driven Process Improvement, and Adaptable Components.

[The Software Engineering Institute](#) (SEI) has an active program in collecting and disseminating information about software product lines. This site is often where material is gathered before it is turned into a book (e.g., Clements and Northrop).

II. Papers of Interest

Perspectives on Software Product Lines *Report on First International Workshop on Software Product Lines: Economics, Architectures, and Implications, Workshop #15 at 22 nd International Conference on Software Engineering (ICSE)*

Introduction to Synthesis: By Campbell, Faulk, and Weiss. To my knowledge, the first paper describing a systematic product line process and the precursor to the FAST process. Was considered proprietary at the time and we were not allowed to publish it in the open literature.

III Application Modeling and Generation

One can gain understanding of the application engineering process by looking at application generations. Applications generators are simply programs that produce programs. Below are two examples of application generators for creating video games. Both programs are by the same [company](#).

Working through the tutorial for either program should give you some insight into both application modeling and the design of (sophisticated) application engineering environments. As you do the tutorial, look back at the slides on application engineering (Week 5) and consider how this system meets the goals of application engineering.

A. Klik-and-Play was a relatively early (but sophisticated) application and is available for free. You can download the program here. A tutorial is available at [this site](#).

B. The Games Factory is a newer version. A working demo can be downloaded from the Clickteam site [here](#).