



Stevens Institute of Technology

Syllabus

SSW565: Software Architecture and Design

Semester taught: Summer 2009	Start and end date of the semester: http://www.stevens.edu/registrar/
Professor Name: Gregg T. Vesonder Office address: none – adjunct faculty	Office Hours: by appointment
Office phone number: E-mail address: gvesonde@stevens.edu	Course Web Address: http://homepage.mac.com/vesonder

Overview

This course provides the student with a working knowledge of the terms, principles and methods of Software Architecture, Design and code refactoring. It focuses on the latest techniques, 4+1 architecture, architectural styles, agile design with software patterns and refactoring, while emphasizing non functional requirements, component based design and best industry practices

Prerequisites

Bachelors degree in Computer Science or equivalent, SSW540

Learning Goals

After taking this course, the student will be able to:

- Construct an architecture using the 4+1 model emphasizing architectural styles and non functional requirements driven by the latest software architecture processes
- Base software design on components, patterns and classes both with agile and more traditional processes using a domain driven approach
- Refactor code
- Begin or accelerate a continuous learning approach to software architecture and design

Pedagogy

The course is a lecture course with substantial in class participation. We will use several different example applications to illustrate the principles discussed in class. The first half of the course will be based on a collection of papers on architectural issues. The second half of the course incorporates two text books. Students are encouraged to draw from their current work experience and relate it to principles in class and also by keeping a log that relates class study to current issues. There will be frequent homework assignments and a mid term and a final.

Required Text(s)

E. Evans. Domain-Driven Design: Tackling complexity in the heart of software, Addison-Wesley, 2004, ISBN 0-321-12521-5

Fowler, M. Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999, ISBN 0-201-48567-2

Required Readings

Readings will be assigned for each week. See weekly descriptions in the table below.

Assignments

Class Participation - To enhance the learning experience, all students are expected to participate in class discussion board by responding to the discussion topics posted by the professor and the postings by other students.

1. Most classes have a homework assignment and they must be turned in promptly
2. Logbooks will be kept relating the principles learned in class with their software experience. There should be at least five entries and they will be collected at the tenth class.
3. There will be a mid term on software architecture and a final on software design and refactoring.
4. Class participation is essential

The assignments and their weights are as shown below:

1. Class Participation, homework, discussion	15%
2. Logbook	15%
3. Mid Term	35%
4. Final	35%
TOTAL	100%

Please note that assignments in this class may be submitted to www.turnitin.com, a web-based anti-plagiarism system, for an evaluation of their originality.

Course Schedule

Class	Subject	Assignment Due
1	Introduction to Architecture and Software Design	<ol style="list-style-type: none"> 1. Introduction to Software architecture 2. The UML 3. Reading: •http://www.sei.cmu.edu/architecture/definitions.html
2	Architecture Styles, Arch connectors	<ol style="list-style-type: none"> 1. Architecture Principles 2. Architecture Process 3. 4+1 Architecture 4. Dvorak paper, Fowler paper "Who needs an architect" - email to you, Vitruvius- <u>On Architecture</u>, Book 1 chapters 1-3, http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Vitruvius/home.html
3	Architecture a Broader View	<ol style="list-style-type: none"> 1. Architectural Styles 2. Architectural connectors 3. "connector" paper, Mehta, Medivdovic & Phadke, 2000 4. Garlan and Shaw "Introduction to Software Architecture"
4	Performance Modeling, Interface Design, Arch modeling	<ol style="list-style-type: none"> 1. Architecture Styles - examples 2. Architecture Description Languages 3. Architecture views 4. Architecture Simplicity 5. Readings: Sha, "Using simplicity to control complexity" and Maeda, <u>The Laws of Simplicity</u>
5	Web-Based Application Architecture	<ol style="list-style-type: none"> 1. Non Functional Requirements: concurrency, performance, security 2. Architecture Review 3. Information Architecture 4. Service Oriented Architecture 5. Readings: Starr Architecture Reviews
6	Special Topics in Architecture	<ol style="list-style-type: none"> 1. Risk 2. Agents 3. Domain Specific Design 4. Architectural Modeling example 5. ATAM 6. Mid Term Review 7. Reading: Tracz and Hall
7	Mid Term	<ol style="list-style-type: none"> 1. Study!
8	Domain Driven Design, First Principles	<ol style="list-style-type: none"> 1. "Traditional" Software Design 2. Introduction to Domain Driven Design 3. Software Patterns 4. Readings: Evans, chapters 1-4
9	Domain Drive Design, the Model	<ol style="list-style-type: none"> 1. Building Blocks of Model Driven Design 2. Entities, Values and Services 3. Readings: Evans Chapters 5-7

Class	Subject	Assignment Due
10	Domain Driven Design, the Implementation, Logbooks due	<ol style="list-style-type: none"> 1. Modules 2. Supple Design 3. Component Driven Design 4. Readings: Evans Chapters 8-13
11	Domain Driven Design, the Lifecycle	<ol style="list-style-type: none"> 1. Strategic Design 2. Communicating Design to Stakeholders 3. Introduction to Refactoring 4. Readings: Evans chapters 14-17; Fowler chapters 1-3
12	Refactoring and special topics.	<ol style="list-style-type: none"> 1. More Refactoring 2. Big Refactoring 3. Special Topics 4. Review
13	Final Exam	