



# Stevens Institute of Technology

## WebCampus.Stevens

### Syllabus

#### SSW565: Software Architecture and Design

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

#### 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 webct course and each week will follow this schedule: Monday morning an annotated lecture in pdf format will be made available, on Wednesday two discussion topics will be provided and you must participate in at least one discussion each week – the discussion ends the following Tuesday and on Thursday a weekly quiz is made available that must be completed by midnight Sunday. During the semester you also are required to keep a log recording your insights on software engineering. Examples of student log entries can be found at my blog, <http://vesonder.typepad.com/universe/>. At the end of the tenth week at least five entries in “doc” or “txt” format should be collected in one file and emailed to me using the webct email system. In addition during the semester there will be a midterm and a final. The midterm will occur on the 7<sup>th</sup> week. The midterm covers software architecture. The final is not cumulative and covers design and refactoring.

#### 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	20%
2. Logbook	20%
3. Mid Term	30%
4. Final	30%
<b>TOTAL</b>	<b>100%</b>

Please note that assignments in this class may be submitted to [www.turnitin.com](http://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"> <li>1. Introduction to Software architecture</li> <li>2. The UML</li> <li>3. Reading:  <a href="http://www.sei.cmu.edu/architecture/definitions.html">http://www.sei.cmu.edu/architecture/definitions.html</a> </li> </ol>
2	Architecture Styles, Arch connectors	<ol style="list-style-type: none"> <li>1. Architecture Principles</li> <li>2. Architecture Process</li> <li>3. 4+1 Architecture</li> <li>4. Dvorak paper, Fowler paper "Who needs an architect"            Vitruvius- <u>On Architecture</u>, Book 1 chapters 1-3,  <a href="http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Vitruvius/home.html">http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Vitruvius/home.html</a> </li> </ol>
3	Architecture a Broader View	<ol style="list-style-type: none"> <li>1. Architectural Styles</li> <li>2. Architectural connectors</li> <li>3. "connector" paper, Mehta, Medivdovic &amp; Phadke, 2000</li> <li>4. Garlan and Shaw "Introduction to Software Architecture"</li> </ol>
4	Performance Modeling, Interface Design, Arch modeling	<ol style="list-style-type: none"> <li>1. Architecture Styles - examples</li> <li>2. Architecture Description Languages</li> <li>3. Architecture views</li> <li>4. Architecture Simplicity</li> <li>5. Readings: Sha, "Using simplicity to control complexity" and Maeda, <u>The Laws of Simplicity</u></li> </ol>
5	Web-Based Application Architecture	<ol style="list-style-type: none"> <li>1. Non Functional Requirements: concurrency, performance, security</li> <li>2. Domain Specific Architecture</li> <li>3. Service Oriented Architecture</li> </ol>
6	Special Topics in Architecture	<ol style="list-style-type: none"> <li>1. Architectural Modeling example</li> <li>2. Agility</li> <li>3. Reading: Hall</li> </ol>
7	Domain Driven Design, First Principles	<ol style="list-style-type: none"> <li>1. "Traditional" Software Design</li> <li>2. Introduction to Domain Driven Design</li> <li>3. Software Patterns</li> <li>4. Readings: Evans, chapters 1-4</li> </ol>
8	Domain Drive Design, the Model	<ol style="list-style-type: none"> <li>1. Building Blocks of Model Driven Design</li> <li>2. Entities, Values and Services</li> <li>3. Readings: Evans Chapters 5-7</li> </ol>
9	Domain Driven Design, the Implementation, Logbooks due	<ol style="list-style-type: none"> <li>1. Modules</li> <li>2. Supple Design</li> <li>3. Component Driven Design</li> <li>4. Readings: Evans Chapters 8-13</li> </ol>
10	Domain Driven Design,	<ol style="list-style-type: none"> <li>1. Strategic Design</li> <li>2. Communicating Design to Stakeholders</li> </ol>

Class	Subject	Assignment Due
	the Lifecycle	<ul style="list-style-type: none"> <li>3. Introduction to Refactoring</li> <li>4. Readings: Evans chapters 14-17; Fowler chapters 1-3</li> </ul>
11	Refactoring	<ul style="list-style-type: none"> <li>1. More Refactoring</li> <li>2. Big Refactoring</li> <li>3. Special Topics</li> <li>4. Readings: Fowler 4-12</li> </ul>
12	Special Topics	<ul style="list-style-type: none"> <li>1. Finish Refactoring</li> <li>2. Special Topics</li> </ul>
13	Outsourcing	Outsourcing