Stevens Institute of Technology
Syllabus

SSW-565: Software Architecture and Component-Based Design

Professor Name: Robbie Cohen **Office address:** 404 Altorfer **Office phone number:** 201 216-8561 **E-mail address:** rcohen2@stevens.edu Email is always the best way to contact me!	**Office Hours**: TBD and by appointment **Course Web Address:** **http://www.stevens.edu/Moodle**

Overview

This course provides the student with a working knowledge of the terms, principles and methods of Software Architecture and Component-Based Design, introducing students to a wide variety of processes and alternatives that can be deployed in creating and/or evaluating software architectures and designs. Numerous case studies of working software architectures are studies and the student will acquire an appreciation of the role that software architecture and design activities play in the acquisition and/or development of complex software-intensive systems.

Special consideration is given to the role of architecture and design in software assurance for dependability, including performance, reliability and security, Recent advances in design techniques, software patterns, component based design and design refactoring are introduced.

Prerequisites

SSW 540 or equivalent experience

Learning Goals

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

- Construct an architecture using the "4+1" model, emphasizing architectural styles and nonfunctional requirements
- Base software design on components, patterns and classes using a domain driven approach
- Re-factor code to improve understanding or modification
- Apply a continuous learning approach to software architecture and design

Pedagogy

The course includes texts and readings drawn from the current academic literature on software architecture and design, and lectures/discussions that present and synthesize that literature. Numerous example software architectures are used to illustrate the principles discussed in class. The course focuses on fundamental software architectural concepts and issues, as well as various design techniques including Domain Design. Students are encouraged to draw from their current and prior work experiences and relate those experiences to the principles discussed in class. Students are required to keep a **log** that relates their course study to current architectural and design issues arising in their other classes, at work or in their general reading.

Class Participation - To enhance the learning experience, all students are expected to participate in class discussions, conducted as online Forums for webCampus students. Class participation is an essential component of the course. Please let the professor know, in advance, if you will be unable to meet your course responsibilities in any given week.

Texts

Required

1. **Beautiful Architecture**, edited by Diomidis Spinellis and Georgios Gousios. O'Reilly Media, Inc., 2009. ISBN: 978-0-596-51798-4.
2. **Domain-Driven Design: Tackling complexity in the heart of software**, authored by Eric Evans. Addison-Wesley, 2004, ISBN 0-321-12521-5.

Recommended, but not required

Just Enough Software Architecture: A Risk-Driven Approach, authored by George Fairbanks. Published by Marshall & Brainerd, 2010. ISBN 978-0-9846181-0-1.

Required Readings

Most weeks, assigned readings include 1 or more chapters from a text plus 1 or more journal articles found as resources on the course Moodle site. Students are expected to have completed the assigned reading **BEFORE** attending a live or recorded lecture event each week.

Assignments

- **Assigned readings** are listed on thecourse schedule for each week. For web campus students, each week begins on Monday with the posting of lecture slides and/or notes for the week.
- **Quizzes** or **assigned written work** should be submitted by noon, Eastern time, on **Thursday** of the week when due.
- **Logbooks** relating the principles learned in this course to each student's software experiences should be kept and updated at least weekly. There should be at least 13 dated entries in the log, one for each week of class. Logs will be collected before the 14th week of class, and individual assignments, entered in the logs may be collected earlier.
- There will be a single **examination** covering all of the course material at the end of the course.

Grading

Grades will be based on the following distribution of credits:

1. Class and Discussion Forum participation	20%
2. Written Assignments and Quizzes, if any	35%
3. Logbook	15%
4. Open book Examination	30%
TOTAL	**100%**

Final grades will be awarded in accordance with the following scale:

Grade	Percentage
A	93 - 100
A-	90-92
B+	87 - 89
B	83-86
B-	80-82
C	70 – 79
F	<70

See the ***Course Schedule*** for reading and other assignments by week.

SSW 565 Course Schedule for Topics, Readings and Assignments						
Wk	**Topic**	**Reading Assignments**			**Written Assignment**	
		BA[1]	**DDD[2]**	**Article[3]**	**Assignment[4]**	**Due**
1	What is Software Architecture	1		Perry & Wolf[a]		
2	Styles & Frameworks	2		Garlan & Shaw[b]	Submit Best/Worst	Your turn
3	Building Blocks	6		Kruchten[c] Mitra[d]	Checklist	
4	Architecturally Significant Requirements	4	1 - 3, + p 507-510		Submit G1: Checklist	G1
5	Component Based Design		4 - 6		ASRs	
6	NFRs	3, 7, 8		Sha[e]	Submit G2: ASRs	G2
7	Designing Real Systems	5, 12		Chappell[f]	Fist Design	
8	Architecture Evaluation		7 - 9	Maranzano[g]	Submit G3: First Design	G3
9	Supple Design & Patterns	11	10 - 13		Better Design	
10	Strategic Architecture and Design	p 372 - 379	14	Avgeriou etal.[h]	Submit G4: Better Design	G4
11	Big Architecture/Big Design		15 - 17		Evaluate a "Better design"	
12	Additional Methods of Architecting & Designing			Fairbanks[i] McCabe & Polen[j] Hofmeister[k]	Submit G5: Evaluate	G5
13	Architecture & Design Research			Taylor and Hoek[l]		Log
14	Final Exam					

Notes for Course Schedule:

1. Chapters or pages (p) in the ***Beautiful Architecture*** text.
2. Chapters in the ***Domain-Driven Design*** text.
3. Journal Articles are listed here and found on Moodle in the assigned week.
    a. "Foundations for the Study of Software Architecture" by Dewayne Perry and Alexander Wolf. **ACM SIGSOFT Software Engineering Notes** 17:40-52, 1992.
    b. "An Introduction to Software Architecture", by David Garlan and Mary Shaw in **Advances in Software Engineering and Knowledge Engineering**, 1993.
    c. "Architectural Blueprints—"The '4 + 1' View Model of Software Architecture" by Philippe Kruchten. **IEEE Software** 12, November 1995.
    d. "Documenting software architecture, Part 2: Develop the system context", by Tilak Mitra. **IBM developerWorks**, 2007.
    e. "Integrate Mixed-Criticality Components with Formalized Architecture Patterns" by Lui Sha, 2008.
    f. "Introducing SCA" by David Chappell. David Chappell & Associates Internet Document, 2007.
    g. "Architecture Reviews: Practice and Experience" by Joseph F. Maranzano et al., **IEEE Software**, 2005.

h. “Toward Using Architectural Knowledge” by Paris Avgeriou, Patricia Lago and Philippe Kruchten. **ACM SIGSOFT Software Engineering Notes**, March 2009.
i. “The Risk-Driven Model: Just Enough Architecture” by George Fairbanks, **Crosstalk** Nov-Dec 2010, pp. 8-11.
j. “Effective Practices for Object-Oriented System Software Architecting”, by Rich McCabe and Mike Polen, **Crosstalk** June 2005.
k. “A General Model of Software Architecture Design Derived from Five Industrial Approaches” by Christine Hofmeister, Philippe Kruchten, Robert L. Nord, Henk Obbink, Alexander Ran and Pierre America. **The Journal of Systems and Software** 80: 106-127, 2007.
l. “Software Design and Architecture: The once and future focus of software engineering” by Richard N. Taylor and André van der Hoek, **IEEE Future of Software Engineering** (FOSE’07), pp. 226-243, 2007.

4. See Moodle for each detailed assignment.