University of North Florida - School of Computing
Course Syllabus for

CEN 6036 – Web Engineering

(Graduate – 3 Semester Credits)

Instructor

Karthikeyan Umapathy, Associate Professor
Office location: Mathews (15/ 3214), 1 UNF Drive, Jacksonville, FL 32224
Phone: 904-620-1329 (office)
Email: k.umapathy@unf.edu

Course Information

Catalog Description

In this course, topics covered include the application of software engineering principles and techniques to the development, deployment, and maintenance of high quality Web-based systems and applications; markup languages, distributed objects, hypermedia and Web integration; architecture and security issues; client side and server side technologies; distributed technologies; data integration across heterogeneous Web sources.

Prerequisite: CEN 6016 - Engineering of Software I or equivalent.

Instructor's Perspective about the Course

The complexity of software systems including Web applications has increased over the decades. Designing software is nowadays beyond the algorithm efficiencies, effective data structures, and transactional processing capabilities. Most real world applications are not written from scratch but rather heavily relies on existing libraries, frameworks, components, and web services. Development of large scale web applications involves careful integration of reusable software components to ensure that the resulting applications are robust and maintainable. Thus, complex Web-based software systems involves application of software engineering techniques but as also system integration principles and tactics. The necessity to integrate, reuse, and maintain large collections of software components has raised some challenges for engineering quality large and complex Web application systems. A new kind of problem has emerged, namely that of the overall system structure, which arises a new criterion for software product: does the system have a good architecture that is understood by stakeholders and developers? Thus, students

Syllabus Contents

learning to create and document software architectures is an important and must have skillset. I have designed this course with software architectures as the central topic.

This course will address some of the issues associated with large-scale Web application development including architectural design and documentation, and service-oriented computing technologies. In the first block of the course, we aim to understand the concepts behind software architectures for large-scale Web-based systems as well as to design, recognize, evaluate and document software architectures. In the second block, we extend our understanding of service-oriented architecture. In particular, we focus on principles behind service-oriented software engineering, and approaches and methods for efficient service production in service ecosystems.

Learning Outcomes

Upon completion of the course, students should be able to:

- Employ techniques to analyze and evaluate software architectures on a real-world large-scale web-based software systems.
- Create and document a reference architecture for a non-trivial Web-based technological product.
- Present findings of case study analysis of software architectures of a family of large-scale web-based software systems.
- Envision an innovative product for a wicked problem and develop an architecture for the product that utilizes service-oriented computing technologies
- Write a research-in-progress paper on a Web engineering topic that utilizes Design Science Research methodology and adheres to appropriate academic standards.

Method of Teaching

Lecture, in-class activities, group projects, research papers, and presentations

Reference Books

There is no prescribed textbook for this course. See below for recommended reference materials for different course topics.

Web Engineering

1. Web Engineering: A Practitioner's Approach by Roger Pressman and David Lowe (ISBN: 9780073523293)

Software Architecture

2. Software Architecture in Practice by Len Bass, Paul C. Clements, and Rick Kazman (ISBN: 9780321815736)

3. Documenting Software Architectures: Views and Beyond by Paul C. Clements, and et al. (ISBN: 9780321552686)

4. Software Architecture: Foundations, Theory, and Practice by Richard N. Taylor, Nenad Medvidovic, and Eric Dashofy (ISBN: 9780470167748)

Service-Oriented Computing

5. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services by Thomas Erl (ISBN: 9780131428980)

6. RESTful Web Services by Leonard Richardson, and Sam Ruby (ISBN: 9780596529260)

7. Building Microservices by Sam Newman (ISBN: 9781491950357)

Design Science Research Methodology

8. Design Research in Information Systems: Theory and Practice by Alan Hevner, and Samir Chatterjee (ISBN: 9781441956521)

Method of Evaluation

Assessment Item	Team Assessment	Individual Assessment
Web Technologies Architecture Case Study		
Deliverable 1 – System Tutorial	1%	4%
Deliverable 2 - System Architecture	5%	10%
Deliverable 3 - Architecture Comparison	5%	15%
Deliverable 4 - Reference Architecture	5%	5%
Case Study Project Presentation	5%	5%
Design Science Research Paper		
First Draft		5%
Interim Draft		5%
Final Draft		10%
New Product Vision Paper		
Product Pitch Presentation		5%
Final Vision Draft		10%
Class Participation		5%
Sub Total	21%	79%
Total	100%	

Letter grades will be based on:
94 – 100 = A
90 – 93.99 = A-
87 – 89.99 = B+
84 – 86.99 = B
80 – 83.99 = B-
77 – 79.99 = C+
70 – 76.99 = C
60 – 69.99 = D
less than 60=F

The penalty for cheating or plagiarizing on assignments will be F grade in the course. Work which is similar beyond coincidence will automatically be considered cheating by all parties.

Late Assignments
There will be a penalty of 10 % per day for late submission of assignments (including weekends and holidays).

Course Passing Requirement

In order to pass this course, at minimum 50% of grade points must be obtained for each assessment item. If you received less than 50% grade points for one of the assessment item, then you will receive F as the final letter grade.

Academic Dishonesty

UNF will not tolerate academic dishonesty in any form as it is contrary to the process of learning. Students should demonstrate academic integrity in all of their course works. Students who violate university rules on academic dishonesty will be punished with the most severe penalty allowed by the university policy.

Please review the University policy on academic misconduct at:
https://www.unf.edu/president/policies_regulations/02-AcademicAffairs/EnrollmentServices/2_0640P.aspx

The policy on academic integrity and misuse of computer equipment and computer accounts found at http://www.unf.edu/ccec/computing/Policies___Guidelines.aspx.

Violations of Academic Integrity

Under this heading the University of North Florida Student Handbook identifies several types of violations; these include but are not limited to: cheating; fabricating and falsifying information or citations; submitting the same work for credit in more than one course; plagiarizing; providing another student with access to one's own work to submit under this person's name or signature; destroying, stealing, or making inaccessible library or other academic resource material; and helping or attempting to help another person commit an act of academic dishonesty. The University of North Florida authorizes any instructor who finds evidence of cheating, plagiarism, or other wrongful behavior that violates the University of North Florida Academic Integrity Code to take appropriate action. Possible action includes, but is not limited to, failing the student on the work in question, failing the student for the course, notifying the appropriate academic dean or Vice President for Student Affairs, and requesting additional action be taken. The consequences of a breach of academic integrity may result in an F, which is unforgivable, regardless of withdrawal status.

Course Outcomes, Assignments, and Activities Matrix

Course outcomes	Assignments/Assessments	Activities Performed by Students
Employ techniques to analyze and evaluate software architectures on a real-world large-scale web-based software systems.	Case Study Deliverable 1 – System Tutorial Case Study Deliverable 2 - System Architecture Case Study Deliverable 3 - Architecture Comparison	Download and install a large-scale Web application system, report on system details, develop a tutorial to install, run, and use the system, and analyze the source code of the system to identify module, component and connector, and allocation views.
Create and document a reference architecture for a non-trivial Web-based technological product.	Web Technologies Architecture Case Study Deliverable 4	Identify variants and invariants with architecture of selected systems, describe the invariants and the variation points using three viewtypes (module, component and connector, allocation), and produce a diagrammatic view of reference architecture for the family of selected systems.
Present findings of case study analysis of software architectures of a family of large-scale web-based software systems.	Case Study Presentation at the SoC Symposium	Present the findings from the case study analysis project at the symposium in a poster format.

Course outcomes	Assignments/Assessments	Activities Performed by Students
Envision an innovative product for a wicked problem and develop an architecture for the product that utilizes service-oriented computing technologies	Product Pitch Presentation New Product Vision Paper	Identify a real-world practical problem, describe an ideal product solution for the problem, develop high-level architectural solution using service-oriented technologies, develop implementation plan to develop the product, present the product vision ideas as a 10 minute product pitch presentation, and write a vision paper following expected academic standards.
Write a research-in-progress paper on a Web engineering topic that utilizes Design Science Research methodology and adheres to appropriate academic standards.	Research Paper – First Draft Research Paper – Interim Draft Research Paper – Final Draft	Identify an applied research problem, read relevant literature, write summary of relevant and important articles, develop a research plan following design science research methodology, develop high-level architecture for research prototype, develop mockups for the prototype, develop an evaluation plan, and write an academic research paper that meets standards of a well-known conference relevant to course topics.

Deliverables

Web Technologies Architecture Case Study

At the heart of every well-engineered Web applications is its software architecture. Software architecture deals with the key design decisions underlying a software system, reflected in the system's high-level building blocks. This project will involve architectural recovery of large and complex open source systems and the comparison of their architecture according to important architectural properties (e.g., extensibility, reliability, performance). You will perform a case study analysis on the architecture of a set of actual systems related to a Web technologies topic, and report on your findings.

Team experience (*Group contract due: Week 2*)
Students are expected to form a team of 3 or 4 students for the case study project.

Deliverable 0 – Systems Selection (*Due: Week 3*)
Each team member should select an open source system that is similar to each other. Select systems that are similar to each other based on feature sets or product domain categories. You can use OpenHub (https://www.openhub.net) for finding appropriate open source software system. OpenHub will indicate which projects are similar to each other. Selected system should be large, architecturally interesting, and relevant to Web technologies.

Deliverable 1 – System Tutorial (*Due: Week 6*)
Each team member should describe the selected system in regards to what it does, the main features, the number of lines of code, the number of contributors, number of commits, number of downloads, the main programming languages used in the system, and estimated cost. Then, write a step-by-step tutorial on (1) where and how to download the source code, (2) how to compile and install the system including dependencies of the

system, (3) how to run system, (4) how to use the system, and (5) demonstrate the system features using a case study example.

Deliverable 2 - System Architecture (*Due: Week 10*)
Each team member should describe module views, component and connector (C&C) views, and allocation views for the selected system. Software architecture practice utilizes the concept of architectural views which are representation of a set of system elements and the relations associated with them. There are three architectural view types: module, component-and-connector (C&C), and allocation. Module views displays static structure of the system as a set of implementation units, C&C views displays system runtime behavior characteristics as a set of cooperating units, and allocation views displays how the system is allocated to structures in the system's environment.

Deliverable 3 - Architecture Comparison (*Due: Week 14*)
Each team member will compare three aspects (functional or non-functional) of the selected system's architecture. The three aspects do not need to be of the same granularity or of the same nature, but at minimum one aspect should be fine-grained.

Deliverable 4 - Reference Architecture (*Due: Week 16*)
In this deliverable, describe a reference architecture for the family of selected systems you analyzed as a team. During generational evolution of the system, changes in the architecture are very likely to occur. System architectures should encompass these variations into their design. Identify at least five invariants and three variation points in your systems. Then, describe the invariants and the variation points using three viewtypes (module, components and connectors, allocation). For each viewtype, identify the invariants and variation points, and indicate how each system fulfill (or not) these invariants and variation points.

School of Computing Symposium (*Due: Week 16*)

CEN 6036 students have option of presenting their case study analysis project work as a poster at the School of Computing Symposium this spring to be held on Friday, April 22, 2016 from 3 PM to 6 PM. Note that you should plan to arrive at the University Center around 2 PM to have sufficient time to check-in and set up your poster.

April 22nd is one of the spring 2016 study break days. If you are not able to present at the symposium due to study break, please contact the instructor during the first week of the course, i.e., on or before January 14, 2016. Instructor will identify alternative public speaking presentation opportunity for those who are not willing to present at the symposium.

Design Science Research-in-Progress Paper

Each student is required to write a research proposal for an applied research project. In the proposal, clearly state a research problem that will have practical impact, develop high-level architecture and design solution for the problem, and develop evaluation plan to assess the utility of (implemented) solution. Topic selected for the research paper should be in an area that is cross-section of topics covered in the course and student's thesis topic. In the research paper, students are mandated to use Design Science Research methodology to conduct their research work.

Topic Selection (*Due: Week 4*)
Submit following information in regards to selected topic for the research paper: briefly describe specific topic that will be focus of the research paper, explain why this topic is relevant to the course, briefly describe how this topic relates to your master thesis, briefly describe objectives of the research paper and how you intend to achieve it, indicate to which conference you intend to model the paper to it meets expected academic standards, and finally, explain why you selected this conference.

First Draft (*Due: Week 8*)
First draft submission must contain following sections: title, abstract, introduction, and problem definition.

Interim Draft (*Due: Week 13*)
Interim draft submission must contain following sections: update the sections submitted in the first draft, literature review, research methodology, and work-in-progress research prototype.

Final Draft (*Due: Week 16*)
Final draft submission must contain all expected sections fully complete and must be updated as per remarks provided by the instructor on previous draft submissions. Expected sections are: title, abstract, introduction, problem definition, literature review, research methodology, work-in-progress research prototype, evaluation plan, and conclusion.

New Product Vision Paper

Web technology and related gadgets have become a necessity to accomplish our daily activities. The quick pace at which the technology is advancing and it's increasing impact on the society makes it important to reflect on what the future holds. This vision paper assignment welcomes ground-breaking positions, analysis, critiques and inspiring proposals about relevant aspects and emerging future trends of Web engineering in general and service computing in particular. This assignment encourages discussion of radical new directions and potentially disruptive software and system engineering innovations. To achieve this assignment goal, you are expect to write a new product vision paper that aims to advance the state of practice in relevance to service computing and enhance your knowledge on service-oriented computing concepts. Vision paper should describe an innovative product that utilizes service-oriented technology in a novel way to solve a real world problem.

New Product Vision Paper Proposal (*Due: Week 5*)
The vision paper proposal submission should include paper title, short description of the proposed system, and a summary of its novel characteristics, functions, and features. You are strongly encouraged to discuss your choice of new product with the instructor before you submit your proposal.

Product Pitch Presentation (*Presentation Dates: Weeks 13 and 14*)
Each student, on one of the above mentioned dates, will be asked to provide overview of the new product as a 10 minute in-class presentation. Students are expected to present their work-in-progress product vision. The presentation must cover following: target audience and demographic information; problem description and why it is non-trivial; product description including name, function, and price; product highlights and value (why the product will be useful) claims; innovative and uniqueness of the product; architecture and high-level design details (a visual representation recommend); role of service-oriented technologies within the product; product implementation plan; and prototype mockups.

Final New Product Vision Paper (*Due: Week 16*)
Submission must contain all expected sections fully complete. As you may note that vision paper does not have intermediate draft submission. Thus, you are highly encouraged to meet with instructor on regular basis to obtain feedback on your vision paper. Expected sections are: title, abstract, introduction, background, problem description, product description, architecture and design, implementation plan, product tutorial, and conclusion.

Attendance and Class Participation

It is important to attend the class regularly, participate, and contribute in the class discussions. Class participation includes attendance, participation in class discussions, and overall knowledge and interest in the course materials. Class participation grade would be linked to your attendance and participation. While attendance may not necessarily be taken every day, both excessive absences and attendance would be duly noted. In case of excessive absence (<50%), will result in zero grade for class participation. In the case of extreme absence (<25%), you may be given failing grade for the entire course, as the extreme absence equals to not taking the course. If you miss a class, it is your responsibility for obtaining the material that is covered and other information provided in the class. Please note that it is not possible to make up a missed class.

This is a hands-on and project-oriented class. You and your team are expected to provide instructor with updates on project work during every class day. Considerable amount of the class time will be allocated for project work and to conduct discussions with your team members. This allows you to ask instructor any questions that you or team may have. You are expected to use this time to ask questions and obtain feedback on your research and product vision papers. Thus, regular class attendance and participation not only helps you but also your team. Regular class attendance by all team members increases the quality of the work produced by the team. Therefore, students are encouraged to attend class regularly, actively participate in the discussions, ask questions, and voice their opinions respectfully.

Course Schedule

It is expected that the student will come prepared to the class meetings with questions for instructor on the course topics and project related issues. The student is responsible for all topics presented regardless of their coverage.

Please note that below listing of chapters does not mean that all text in those chapters would be covered in this course. Only materials pertaining to course would be covered. Throughout the course, instructor would provide other supplementary materials to provide targeted guidance to team project deliverables.

Week	Topics	Chapters	Due Dates
1	Course Introduction and Syllabus Review		**Team Information**
2	Web Engineering and Application Design Principles	TB1 – Chapters 2, 8 to 11 and 13	**Team Contract**
3	Software Architectures in SDLC	TB2 – Chapters 1 to 3, and 15 to 17	**Case Study Project – Deliverable 0**
4	Documenting Architectural Views and Styles	TB2 – Chapters 18 and 21	**Research Paper – Topic Selection**
5			**Vision Paper - Proposal**
6		TB2 – Chapters 4 to 14	**Case Study Project – Deliverable 1**

Week	Topics	Chapters	Due Dates
7	Architectural Tactics and Quality Attributes		
8	Reference and Product Line Architectures	TB2 – Chapter 25	**Research Paper – First Draft**
9	Design Science Research Methodology	TB3 – Chapters 1 to 3	
10	SOA and Web Service Technologies	TB4 – Chapters 3 and 4	**Case Study Project – Deliverable 2**
11	Spring Break		
12	RESTful Web Services	TB5 – Chapters 3 and 4, and 8 to 10	**Vision Paper – Product Pitch Presentation**
13	Integrating Applications using Web Services	TB4 – Chapters 8 to 14	**Research Paper – Interim Draft** **Vision Paper – Product Pitch Presentation**
14			**Case Study Project – Deliverable 3** **Vision Paper – Product Pitch Presentation**
15	Microservices and Internet of Things	TB6 – Chapters 1 and 2	
16	Final Week		**Case Study Project Presentations at SoC Symposium** **Case Study Project – Deliverable 4** **Research Paper – Final Draft** **Vision Paper – Final Draft**

Legends

TB1 – Web Engineering: A Practitioner's Approach by Roger S. Pressman and David Lowe
TB2 – Software Architecture in Practice by Len Bass, Paul Clements, and Rick Kazmann
TB3 – Design Research in Information Systems by Alan Hevner and Samir Chatterjee
TB4 – Service-Oriented Architecture by Thomas Erl
TB5 – RESTful Web Services by Leonard Richardson and Sam Ruby
TB6 – Building Microservices by Sam Newman

Course Modifications

Instructor reserves the right to modify course including schedule, assignment specifications, assignment score distributions, grading criteria, and other relevant aspect to meet the student's needs or due to unexpected events.

Other Remarks

Students with Disabilities

Students with disabilities who seek reasonable accommodations in the classroom or other aspects of performing their coursework must first register with the UNF Disability Resource Center (DRC) located in Building 57, Room 1500. DRC staff members work with students to obtain required documentation of disability and to identify appropriate accommodations as required by applicable disability laws including the Americans with Disabilities Act (ADA). After receiving all necessary documentation, the DRC staff determines whether a student qualifies for services with the DRC and if so, the accommodations the student requires will be provided.

DRC staff then prepares a letter for the student to provide faculty advising them of approved accommodations. For further information, contact the DRC by phone (904) 620-2769, email (drc@unf.edu), or visit the DRC website (http://www.unf.edu/drc/).

Military and veteran students who return from combat exposure may be utilizing the post 9/11 GI bill to continue postsecondary education goals and may need both physical and academic accommodations. Contact Ray Wikstrom, Director of Military and Veterans' Resource Center by phone (904) 620-2655, email (ray.wikstrom@unf.edu).

Satisfactory Progress Policy

The School of Computing enforces the "one repeat" rule for all prerequisite and core courses offered by the School for its major programs. Students who do not successfully complete a prerequisite or core requirement for a School of Computing course on the first attempt (i.e., earn a grade of D, F, W, WP or WF) will be granted one chance to repeat the course. Students who do not successfully complete a prerequisite or core requirement within two attempts will not be permitted to register for courses offered by the School in future semesters. This stipulation applies whether or not the student has declared a major in a School of Computing program.
http://www.unf.edu/ccec/computing/PoliciesGuidelines/Satisfactory_Progress_Policy.aspx

Continuity of Instruction Plan

In the event of disruption of normal classroom activities due to an emergency such as hurricane, pandemic or other unforeseen event or combination of events, the format of this course may be modified in order to enable completion of the course requirements. In that event, you will be provided an addendum to this syllabus that will supersede this version. It is your responsibility as a student participant to be proactive during any emergency to find instructions that I will post on Blackboard which you should check daily.