

CptS 421: Software Design Project I

Instructor: Li Tan, Ph.D.

Assistant Professor
School of Engineering and Computer Science
Washington State University, Tri-Cities

Email: litan@tricity.wsu.edu

Fall 2016

Outline

- 1 Overview
 - Senior design project in nutshell
- 2 Class Policies
- 3 Get Started
 - Homework
 - Workflow and Team Activities

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- The course provides a platform to practice software engineering in a disciplined manner. You will exercise,
 - established industrial practices.
 - You will establish a quality-oriented team and personal process.
 - principles and techniques from classroom.
 - You will apply software development principles, models, and design patterns to the project. creativity and management skills.
 - You will closely work with your team members to deliver a software project.

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...
- **To learn,**

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...
- **To learn, to think,**

Objectives

Goal

Provide a hands-on experience of developing medium- to large-size software products.

- Senior design project is a step stone between classroom learning and your future career as:
 - 1 a software engineer;
 - 2 a graduate student and later a researcher;
 - 3 an entrepreneur;
 - 4 ...
- **To learn, to think, and to have fun.**

Fast Facts

- CptS 421 is the first course on a year-long sequence of software development project involving all five framework activities:
 - **Communication:** Project initialization, requirement elicitation.
 - **Planning:** scheduling, quality process, team and individual processes, ...
 - **Design:** architecture/component design, behavior design, class design, ...
 - **Implementation.**
 - **Deployment:** delivery, support, revisions, ...

- Projects generally contain a research component in it.
 - You are expected to practice and improve independent research and problem-solving skills.

Fast Facts

Result-Oriented and Interactive Learning Experience

- Focus on executions, i.e., project development, and individual problem solving skills.
- Feature an interactive learning environment.
 - Class activities will include lectures, student presentations, and group discussions and meetings.
 - Student presentations, class discussions, and lectures on Tuesdays;
 - Each team will set up a weekly meetings with me;

Fast Facts

Homeworks and Exams

- No written midterms;
- No traditional Q&A homeworks;
- Instead, you may be assigned with two types of homeworks:

Type I Project-oriented homeworks designed to help you learn the tools and the techniques that will be used in your team project;

Type II Research-oriented homeworks with topics relevant to your projects, e.g., data visualization, mobile computing etc.

Fast Facts

Homeworks and Exams

- No written midterms;
- No traditional Q&A homeworks;
- Instead, you may be assigned with two types of homeworks:

Type I Project-oriented homeworks designed to help you learn the tools and the techniques that will be used in your team project;

Type II Research-oriented homeworks with topics relevant to your projects, e.g., data visualization, mobile computing etc.

Fast Facts

Homeworks and Exams

- No written midterms;
- No traditional Q&A homeworks;
- Instead, you may be assigned with two types of homeworks:

Type I Project-oriented homeworks designed to help you learn the tools and the techniques that will be used in your team project;

Type II Research-oriented homeworks with topics relevant to your projects, e.g., data visualization, mobile computing etc.

Fast Facts

Homeworks and Exams

- No written midterms;
- No traditional Q&A homeworks;
- Instead, you may be assigned with two types of homeworks:

Type I Project-oriented homeworks designed to help you learn the tools and the techniques that will be used in your team project;

Type II Research-oriented homeworks with topics relevant to your projects, e.g., data visualization, mobile computing etc.

Resources

Websites

Course website: <https://www.tanli.org/cpts421>

- Class notices will be posted online.

Office Hours

- Time: Tuesdays 3-4 pm or by appointment
- Office: Dr. Tan's office, West 134L.

Resources

Textbooks

- (Required) Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1994. ISBN: 0201633612. (a.k.a. the “Gang of four” book)
- (Recommended) Bernd Bruegge and Allen H. Dutoit , Object-Oriented Software Engineering: using UML, Patterns, and Java. 2th edition, Prentice Hall. 2003. ISBN-10: 0130471100.

Resources

Online tutorials

These tutorials are required readings.

- **Java Enterprise Edition. Oracle.**
`https://docs.oracle.com/javasee/7/tutorial/`
- **HTML Tutorial.** `http://www.w3schools.com/html/`
- **CSS Tutorial** `http://www.w3schools.com/css/`
- **Javascript Tutorial** `http://www.w3schools.com/js/`

Outline

- 1 Overview
 - Senior design project in nutshell
- 2 **Class Policies**
- 3 Get Started
 - Homework
 - Workflow and Team Activities

Prerequisites

Prerequisites

Prerequisites for this course is,

- Cpt S 322: Software Engineering Principle I;
- Cpt S 323: Software Design.

You must take the above courses before taking CptS 421.

Copyright Assignment

Senior design project is a trainee/education opportunity.

- The copyright of design design project belongs to project sponsors.
 - You may be required to sign a copyright release form;
- there will be at least one project option that does not require the signing of a copyright release form.

Submission policies

- Homework assignment and project materials are submitted using SVN.
 - Each project will have a repository.
 - Each student will have his/her own SVN repository.
- Late penalty.
 - Advanced notice to the instructor is required for late submission.
 - 10% reduction of points for the late homework presentation for up to 1 week with prior request for extension.

Performance Metrics

Project Score

$$S = SM * (40\% * PT + 60\% * PI)$$

- SM: the maximal points decided by implemented requirements;
- PT: team performance.
- PI: Individual performance in team.

Performance Metrics

- Term report and presentation:
 - At the end of the semester your team will write a report to summarize project activities and deliverables,
 - For each deliverable, the report will include the percentage of contribution from each individual member.
 - Your team will give a presentation on the last lecture of the semester, and term report is due at the same time.
- Weekly status briefing:
 - You will fill in your status on your team website before weekly meeting with the instructor.
- You and your team performance is judged by the quality of finish product, and more importantly, by the quality of,
 - Software artifacts such as design and testing plans
 - Team and personal processes;

Policy on data management

You will produce many form of data, including computational data and design documents. You are responsible for safe store and management of these data.

Good practices

- Back up often, and back up early to prevent the damage.
- Maintain distributed copies of data on different location and media.

Policy on SVN Repositories

- Your personal repository can only be accessed by you and the instructor;
- Your team repository can only be accessed by you, your team members, and authorized developers.
- You are responsible for maintaining your repositories. If your repository is corrupted/destroyed, your performance will be evaluated based on whatever data has been left on the repository.
- **Do not modify the repositories directly on ELEC system under /home/svn. The repositories are not playground. It is a violation of academic integrity to modify (e.g. delete) the repositories on the ELEC system. Always use client-side command. If you are not clear about this policy, ask the instructor.**

General policies

Academic Integrity

- Will strictly follow the academic integrity standard per university policy. No exception.
- Use your common sense. If you have any doubt about this policy, please come to see me.
- Make good use of office hours. If you are unclear on homeworks/projects, ask the instructor!

Disability Accommodation

Please notify the instructor during the first week of class of any accommodations needed for the course.

General policies

Academic Integrity

- Will strictly follow the academic integrity standard per university policy. No exception.
- Use your common sense. If you have any doubt about this policy, please come to see me.
- Make good use of office hours. If you are unclear on homeworks/projects, ask the instructor!

Disability Accommodation

Please notify the instructor during the first week of class of any accommodations needed for the course.

Outline

- 1 Overview
 - Senior design project in nutshell
- 2 Class Policies
- 3 Get Started
 - Homework
 - Workflow and Team Activities

Personal and Team Activities

A (partial) list of activities that you and your team will undertake besides project development:

- Fill in weekly status and attend weekly status briefing/milestone meeting;
- Build and maintain team websites;
- Develop and document your team and personal processes;
- Fill peer-review forms, and write the project reports with your team;
- Attend the team meetings called by your teammates and/or the instructor;
- Research homework topics, and prepare class presentation;
- ...

Processes, Tools, and Standards

Emulate an industrial software process using open-source, or commercial (free of charge) tools.

- Use rational Unified Process (UP) as the underlying process model;
- Use SVN as version control system;
- Use a Joomla website as media exchange platform;
- Use Bugzilla to organize development and quality assurance activities;
- Use Eclipse-based development tool;
- Use Axure UI as a UI prototyping tool.

Design with UML

We will use UML in requirement analysis and software design.
Each design contains:

- Scenario-based element
 - E.g. use-case diagrams.
- Behavior-based element
 - E.g. state diagrams, activity diagrams, sequence diagramsetc.
- Class-based element
 - E.g. class diagrams.
- Architecture-oriented element:
 - E.g. deployment diagrams, component diagrams etc.

Design with UML

We will use UML in requirement analysis and software design.

Each design contains:

- Scenario-based element
 - E.g. use-case diagrams.
- Behavior-based element
 - E.g. state diagrams, activity diagrams, sequence diagram etc.
- Class-based element
 - E.g. class diagrams.
- Architecture-oriented element:
 - E.g. deployment diagrams, component diagrams etc.

Design with UML

We will use UML in requirement analysis and software design.

Each design contains:

- Scenario-based element
 - E.g. use-case diagrams.
- Behavior-based element
 - E.g. state diagrams, activity diagrams, sequence diagramsetc.
- Class-based element
 - E.g. class diagrams.
- Architecture-oriented element:
 - E.g. deployment diagrams, component diagrams etc.

Design with UML

We will use UML in requirement analysis and software design.

Each design contains:

- Scenario-based element
 - E.g. use-case diagrams.
- Behavior-based element
 - E.g. state diagrams, activity diagrams, sequence diagramsetc.
- Class-based element
 - E.g. class diagrams.
- Architecture-oriented element:
 - E.g. deployment diagrams, component diagrams etc.

Design with UML

We will use UML in requirement analysis and software design.

Each design contains:

- Scenario-based element
 - E.g. use-case diagrams.
- Behavior-based element
 - E.g. state diagrams, activity diagrams, sequence diagramsetc.
- Class-based element
 - E.g. class diagrams.
- Architecture-oriented element:
 - E.g. deployment diagrams, component diagrams etc.

Design with Patterns

We use projects to practice design patterns:

- Every student will take turn to present design patterns from the textbook.
- During project, you are required to think actively about applying patterns to your design.
 - As part of your final report, you will document your efforts on applying design patterns.

Design with Patterns

We use projects to practice design patterns:

- Every student will take turn to present design patterns from the textbook.
- During project, you are required to think actively about applying patterns to your design.
 - As part of your final report, you will document your efforts on applying design patterns.

Unified Processes

Unified Process

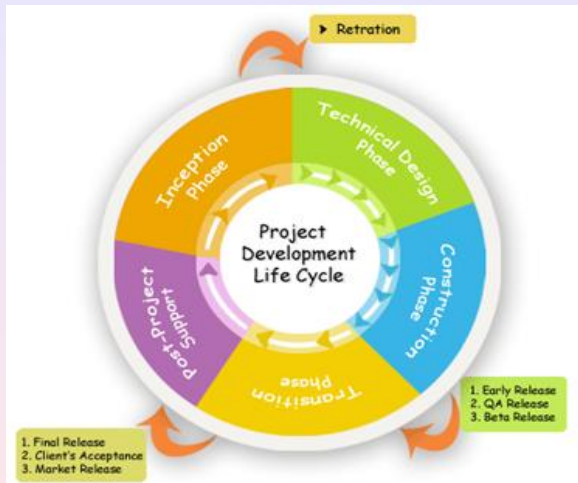
A “use-case driven, architecture-centric, iterative, and incremental” software process closely aligned with the Unified Modeling Language.

Unified Processes

Key Features

- Software processes incorporate OO analysis and design using UML.
 - We use IBM rational software architect for modeling in UML.
- Iterative development;
 - Each iteration will produce some software artifacts
 - E.g. Use-cases, architecture design, class-diagram.
- Early iterations seek feedback from the customer.
 - Risk is managed through early feedback.
- Risk focus
 - focus on addressing the most critical risks early in the project life cycle

Workflow and Team Activities



- Every iteration - identify use cases, create a design, implement the design
- Every iteration is a complete development process

Unified Process: Phases

Inception Phase

- Usually the smallest phase in the project.
- Establish a justification or business case for the project
- Establish the project scope and boundary conditions
- Outline use cases and key requirements that will drive the design tradeoffs
- Outline one or more candidate architectures
- Identify risks
- Prepare a preliminary project schedule and cost estimate

Unified Process: Phases

Elaboration Phase

- Capture a healthy majority of the system requirements.
- Address known risk factors
- Establish and validate the system architecture.

Construction Phase

- Implement design.
- Each iteration results in an executable release of the software.

Unified Process: Phases

Transition Phase

- Deliver to customers
- Refinement based on feedback

Unified Process: Work Products

Inception Phase

- Vision and requirement documents;
- Initial user-case model;
- Initial risk assessment;
- Project plan;
- Business model (if applicable);

Unified Process: Work Products

Elaboration Phase

- Use-case model;
- Supplemental requirements (any non-functional requirements);
- Analysis models;
- Software architecture design;
- Executable architecture designs (state diagrams, sequence diagrams, etc);
- Preliminary object design model;
- Elaborated project plans;
- Preliminary user manuals.

Unified Process: Work Products

Construction Phase

- Design models;
- Software components and code base;
- Test plan and procedures;
- Unit test cases;
- Support documentation (user manual etc).

Transition Phase

- Delivered software increment;
- Release management;
- Beta test report and documented user feedback.