CECS 343:
Introduction
to
Software Engineering

Instructor: Vatanak Vong

Course Overview

Objectives

- Introduction to the software development discipline
- Experience delivering multiple milestones as a software team
- Application of modeling a system using UML

Resources

- Textbook
  - Software Engineering: A practitioner's approach 8th Edition by Pressman & Maxim
  - Chapters: 1-12, 17 & parts of 29
- Reference Materials
  - https://github.com/v-vong3/csulb/tree/master/cecs_343

Important

- This slide deck is only for highlighting main topics. Students still need to utilize all other materials to fully comprehend the subject matter.

Semester Project

Teams

- Teams of five students max

- All team assignments need to have the following information on the front
  - Team Name
  - Team Leader
  - Team members
  - Date

- Each team will compete against each other to provide the a solution that best satisfies the client’s need.

- The instructor will portray the client that will choose the winning solution.

Milestones

- **Milestone 1**
  - Business Requirements Document (BRD)

- **Milestone 2**
  - Project Plan

- **Milestone 3**
  - Design Document
  - Test Plan

- **Presentation**
  - Request For Proposal (RFP) presentation

Software

Application Domains

- **System Software**
  - Programs used by other programs (e.g. compilers for IDEs, file utilities for OS)

- **Application Software**
  - Programs that addresses a specific business need (e.g. Auto email marketer, low disk space network scanner)

- **Scientific Software**
  - Number crunching programs that intake data and perform calculations/evaluations (e.g. Space Flight simulator, molecular breeding)

- **Embedded Software**
  - Programs controlling low level functions of a broader system (e.g. firmware, drivers, car dashboard)

Application Domains

- **Product-line Software**
  - Mass-marketed programs that provide specific capabilities (e.g. Microsoft Office, Visual Studio). These programs are considered “base-models”; they are not meant to be tailored.

- **Mobile Software**
  - Programs for tablets or smart phones that may or may not be network connected (e.g. iMessage app, iTunes Store app)

- **Web Software**
  - Browser-based programs that mainly relies on an active connection (e.g. **google.com**, **facebook.com**)

- **AI Software**
  - Software that utilizes heuristics to makes decisions or provide deep insight that would normally be difficult using straight forward analysis (e.g. IBM Watson, Azure ML)

Jobs

- **Web**
  - Front-end Developer (HTML, JavaScript, CSS)
  - Back-end Developer (Java, .NET, PHP, JavaScript for node)
  - Full-stack Developer (Both front-end & back-end)

- **Mobile**
  - Mobile Front-end Developer (Swift, Kotlin, Objective-C, Java)
  - Mobile Full-stack Developer (Mobile Front-end & Web back-end)

- **IT Administrator**
  - Windows (Powershell, Batch, VBScript, Python)
  - Max\Linux (Bash, Python)

- **Gaming**
  - Graphics programmer (Unity, Unreal Engine, OpenGL)
  - Gameplay/Level programmer (Lua scripting\Python\custom)

- **Data**
  - Data Scientist (SQL, Python, R)
  - SQL Developer (SQL, T-SQL, PL/SQL)

- **Other**
  - Desktop Application Developer (Java/.NET)
  - Embedded Systems Engineer (C/C++/Assembly/Python)

SDLC

Phases

- **Requirements**
- **Planning**
- **Design**
- **Construction**
- **Testing**
- **Release**

Phases

- **Value Identification**
- **Requirements**
- **Planning**
- **Design**
- **Construction**
- **Testing**
- **Release**

Value Identification

Purpose

- **Answer these questions**
  - Is there a market for the product?
  - Is there a positive, achievable outcome (e.g. profit, productivity, etc.)?
  - If the client knows what they want, why are these features useful to them?

- **Market Research**
  - Mainly for determining if a demand for a new solution exist
  - Survey or poll a random population asking if a solution would be desirable
  - Can be used for determining new enhancements for an existing product

- **Client/user pain points**
  - Mainly for determining new enhancements
  - Survey or poll users can send feedback for enhancements that they would like to see and why
  - Inquire users on what issues they are having the most trouble with and offer a solution to it
  - Can be used for determining market for a new product

Scenario

John needs a calculator.

- **Identify John's pain points**

Scenario

John needs a calculator
to estimate monthly mortgage payments.

- **Identify John's pain points**

Requirements Phase

Purpose

- Verify
  - Ensures feature exist
  - Login button exist

- Validate
  - Ensures feature is working as expected
  - When clicked, does login button logs me in as me or another user?

Activities

- **Requirements Gathering**
  - Communicating with client to determine their pain points and goals
  - Identifying software features (Functional Requirements)
    * Add function that displays the sum of operands
  - Quantifying constraints/behaviors of software features (Non functional requirements)
    * Operands must be integer values
    * Display must be a legible san serif font (15pt to 20pt size)

- **Requirements Analysis**
  - Understanding the problem
  - High level modeling
    * User stories
    * UML diagrams (high-level ones)

Artifacts

- Business Requirements Document (BRD)
  - Enumerates the features and capabilities that the client/system requires

- Includes
  - Project value
  - Functional Requirements
  - Non-Functional Requirements

- Project Road Map

User Stories

- **Template**
  - <Subject> <Action> <Outcome/Reason>

- **Examples**

- As a student, users can register for classes on the school's portal.

- Students can register for classes using the school's portal.

- Bob, a student, logs into the school's portal in order to register for classes.

UML Diagrams

- **High level modeling**
  - Workflow/Activity Diagram
  - State Diagram
  - Swim-lane Diagram
  - Use Case Diagram
- **Advantages**
  - Easier to comprehend the system holistically
  - Helps with scoping & estimations
- **Disadvantages**
  - Changes to “core” requirements can invalidate entire diagram
  - Unable to determine specifics of system

Considerations

- Make sure client approves of final BRD draft or statement of work (SOW) before proceeding to implementation
- Make sure entire development team understands project value and direction

Assignment

Jane is trying to study for two finals, but she wasn't very organized during the semester. She has to search for study materials from her hand written notes, emails and class handouts. One subject will only cover the last 2 months on the final while the other final will cover everything in the semester.

- Identify Jane's pain points
- Provide the functional and non-functional requirements of a solution that would address her pain points

Planning Phase

Purpose

- Determining
  - workload
  - estimations
  - the start of the project
  - the deadline
  - assignment of work

Main Goals of Projects

1. Deliver within budget
2. Deliver on time
3. Deliver to specification

In real life

1. Deliver within budget
2. Deliver on time
3. Deliver to specification

Can only achieve ***two*** out of the three.

Planning Considerations

1. Scope
    - What are you going to deliver?

2. Resources
    - Staffing needs, budget

3. Time
    - Internal vs External, Estimations

4. Priorities
    - Original work items vs changes

5. Risks
    - Worst case scenario

Artifacts

- Project Plan
  - Provides project overview and details timelines for completing the project

- Includes
  - Project scope
  - Resources (staff, budget, cost estimations)
  - Project timeline (milestones, deadlines)
  - Risk management

- Project Road Map (if not defined in BRD)

Estimation

- **Estimation by Guessing**
  - Using random numbers
  - Do NOT use this approach

- **Estimation by User Stories (Story Points)**
  1. Decide on a point scale
     - By complexity
       - 1-2: Easy (4-16 hours)
       - 3-4: Medium (16-24 hours)
       - 5: Hard (24-36+)
     - By hours of effort (prime numbers)
  2. Assign points to stories

Estimation

- Estimation by Previous Experience
  1. Determine the scope of the current task
  2. Recall a past work item with similar scope
  3. Assign actual effort of past work item to current task

- Estimation by Math

$$\frac{(\text{Best}) + (\text{Medium} * 2) + (\text{Worst} * 3)}{6}$$

Assignment

John was assigned two projects: A & B. Project A requires 200 hours of effort while project B requires 300 hours of effort. Project B has higher priority than project A. John has a team of three developers including himself that utilizes the Scrum methodology with 3-week sprints. Each developer can handle roughly 40 hours of work a week.

- Provide a project plan for delivering both projects.

Design Phase

Purpose

- Determining
  - how features will be built
  - third party integration points
  - structure of data
  - integration of technology
  - research points
  - estimations

High vs Low

- High Level Design
  - 10,000 foot view
  - Provides only main components of a system
  - Easier to digest a system/component

- Low Level Design
  - 1 foot view
  - Provides all the details of a component
  - Complete understanding of a component

UML Diagrams

- **Low level modeling**
  - Class Diagram
  - Entity Relationship Diagram
  - Sequence Diagram
- **Advantages**
  - Provides details needed to fully comprehend a system
  - Helps with development of system
- **Disadvantages**
  - Changes to “core” and feature requirements may invalidate entire diagram
  - Difficult to determine role/purpose of feature or other integration points

Artifacts

- Design Document
  - Provides implementation details for satisfying requirements

- Includes
  - UML Diagrams
  - Enumerates patterns and architectures

- Technology Specification (Tech Specs)
- Test Plan

Considerations

- Architectural Design
  - Holistic guidelines/constraints of system

- Design Patterns
  - Industry accepted coding constructs that solve common problems

- Document design decisions

Assignment

Model the following domain into objects

- Spearfishing
- Bank/Pond fishing

Construction Phase

Purpose

- To create the software
  - Construction is the only phase that needs to occur in order for software to exist

Artifacts

- Software
  - The entire implementation “code” base

- Includes
  - Source code
  - Unit tests
  - Assets (e.g. images, XML files, scripts)
  - Third party packages

Testing Phase

Purpose

- Verify
  - Ensures feature exist

- Validate
  - Ensures feature is working as expected

- Fix bugs before Release

Jobs

- Quality Assurance (QA) / Tester
  - Manual testers

- Software Developer in Test
  - Units tests
  - Automation testing

Artifacts

- Test Plan
  - Details how the system will be verified and validated

- Includes
  - Testing scope
  - Test approach (e.g. regression, performance, etc)
  - Test data
  - Success test cases
  - Failure test cases
  - Pre/Post conditions for test cases
  - Timeline

Considerations

- Bug Triaging
  - Organize and prioritize bugs by severity metrics
  - Assign bugs to resource to address

- Bug Severity
  - High dictates non-working crucial feature
  - Medium dictates non-working feature
  - Low dictates cosmetic defects

- Determine software “stable” criteria

Assignment

John is building a simple calculator app for iOS. He wants the application to be able to add, divide, subtract and multiply. The operands will be only decimal numbers. The final calculation should be displayed as decimal numbers.

- Provide at least 5 test cases

Release Phase

Purpose

- Initial deployment
  - General availability

- Maintenance
  - Bug fixes, patch
  - Applying requirement changes

- Enhancement
  - Adding additional functionalities or improvements

Artifacts

- Release Notes
  - Documents all changes to existing system or new features available

- Includes
  - High level change for end-users
  - Low level change for developers
  - Version
  - Date
  - Build Number

Versioning

- Random Number Versioning
  - Do NOT use this approach

- Year Versioning
  - Typically only used for marketing purposes (Visual Studio 2017)

- Generation Versioning
  - A higher number every time system is re-written (API v2)

- Semantic Versioning
  - Major.Minor.Patch (e.g. iOS 10.3.3, Visual Studio 15.5.4)
  - Major dictates breaking changes
  - Minor dictates non-breaking changes
    - ❖ Deprecating features or APIs
    - ❖ Enhancements
  - Patch dictates

Software Methodologies

Waterfall Model

Process

- Linear flow for developing software

1. Requirements
2. Planning
3. Design
4. Construction
5. Testing
6. Release

Advantages

- Excels in projects with strict, non-changing (or very minor changes in) requirements such as in small projects
- Easy to adopt

Disadvantages

- Unable to account for changes in non-minor requirements
- Requires more resources if organization is structured by job function
- Low confidence of client satisfaction

Spiral Model

Process

- Iterative approach where SDLC phases are done at the feature level and repeated until all features are addressed

Advantages

- Able to address changes in requirements per iteration
- Easy to adopt
- Have a working system sooner

Disadvantages

- Timelines per iterations can vary drastically depending on complexity of feature
- Requires more resources if organization is structured by job function
- Sometimes knowing more requirements ahead of time will result in a better designed system

Agile

Main Principles

- Improve customer satisfaction through constant communication throughout project

- Emphasize working system instead of process

- Frequent, iterative releases to obtain customer feedback

Scrum

Agile != Scrum

- Extreme Programming
- Team Software Process

Process

- Scrum Master
- Backlog Grooming
- Sprint
- Sprint Planning
- Team Velocity
- Burn-down / Burn-up Chart
- Kanban Board
- Daily Stand-up
- Sprint Retrospective

Advantages

- Able to address changes in requirements per sprint
- Have a working system every sprint
- High confidence in client satisfaction

Disadvantages

- Timelines per iterations can vary drastically depending on complexity of feature
- Requires more resources if organization is structured by job function