

Overview

CMPSC 488: Computer Science Project
Spring 2016
Penn State Harrisburg

Project Information

- Transform Your Knowledge into Practice
- Learn to work in teams
 - 3-4 person self-organized team
 - Reasonably complex software product
- Grading
 - Based on the deliverables from 8 project phases

Project Phases

- Phase 0 : Team Formation / Project Selection
- Phase 1: Requirements Elicitation
- Phase 2: Preliminary Design
- Phase 3: Detailed Design
- Phase 4: Project Milestones
- Phase 5: Coding / Testing
- Phase 6: Promotional Website
- Phase 7: Product Delivery

Project Phase 0

- Team Formation / Project Selection
 - Due: January 13, 2016
 - Description
 - 3-4 self-organized team
 - Self-selected software project
 - Project with reasonable complexity
 - Approval required
 - Deliverable
 - One-page project proposal

Software Engineering Activities

- Specification Phase
- Design and Implementation Phase
- Verification and Validation Phase

Software Engineering Activities

- Specification Phase
 - Understand software product
 - Activities
 - Identify ideas
 - Elicit requirements
 - Express requirements
 - Prioritize requirements
 - Manage requirements
 - Formulate potential approaches

Software Engineering Activities

• Design and Implementation Phase

– Activities

- Design the architecture
- Design the database
- Design the interfaces
- Write the code
- Integrate functionality
- Document

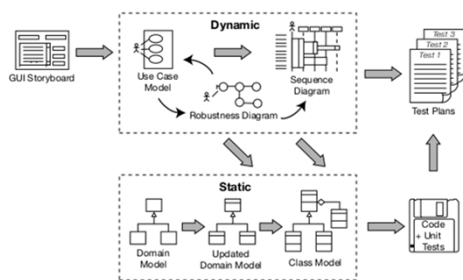
Software Engineering Activities

• Verification and Validation Phase

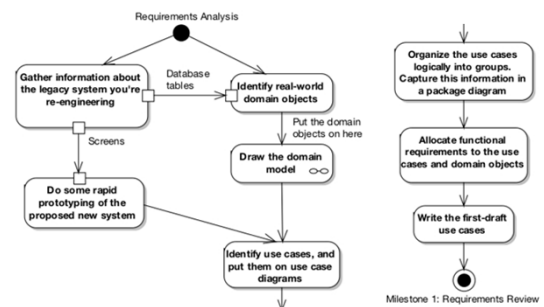
– Activities

- Develop test procedures
- Create and execute tests
- Conduct reviews
- Demonstrate to clients
- Conduct retrospectives

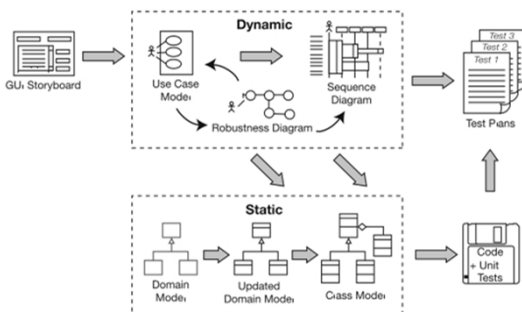
Software Process



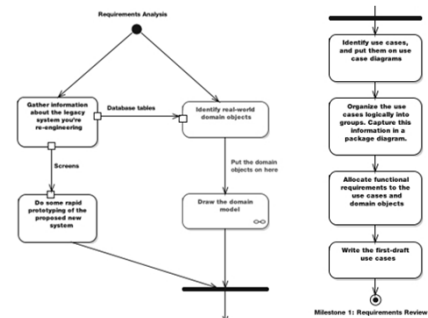
Software Process



Domain Modeling



Domain Modeling



Project Phase 1

- Requirements Elicitation
 - Due: January 20, 2016
 - Description
 - Elicit requirements from client
 - Express requirements in clear/unambiguous way
 - Create Domain Model and System Glossary
 - Deliverable
 - Functional Requirements, Domain Model, Glossary

Requirements Elicitation

- Interact with clients
 - Clients are responsible for providing requirements
 - You are responsible for helping clients
 - Use mockups, prototypes, wireframes, storyboards
 - Create glossary for the product
- Interview end-users
- Conduct feasibility studies
- Watch end-user using the product

Requirements Elicitation

- Client Interaction
 - Find a correct balance
 - Not too passive, Not too aggressive
 - Stay focused on the goal / purpose of product
 - Stay as independent of technology as possible

Requirements Elicitation

- Good Questions to Ask
 - Ask good open-ended questions
 - Avoid simple yes/no questions
 - Constantly ask “Why?”
 - Why do you need the product?
 - Why would anyone use it?
 - Why do you want it that way?
 - etc

Requirements Elicitation

- Questions to Avoid
 - Simple yes/no questions
 - Too open-ended questions
 - What do you want?
 - What are your requirements?

Requirements Elicitation

- Good Questions to Ask
 - For Business Requirements
 - What problem are you trying to solve?
 - What is the motivation for solving this problem?
 - What would a highly successful solution do for you?
 - What is a successful solution worth?
 - Who could influence this project?
 - Who could be influenced by this project?

Requirements Elicitation

- Are there any related projects to this one?
- Which activities should be included in the scope?
- Could there be any unintended consequences of the new system?

Requirements Elicitation

- Good Questions to Ask
 - For Business Rules
 - What policies must the product conform to?

Requirements Elicitation

- Good Questions to Ask
 - For User Requirements
 - Who will use the system?
 - Will there be several types of users?
 - What is the skill level of each user?
 - What goals could this product help you accomplish?
 - What problems do you expect this product to solve?
 - What words would you use to describe the product?
 - What aspect of the product excites you?
 - What aspects are most/least valuable to the users?

Requirements Elicitation

- Good Questions to Ask
 - For Functional Requirements
 - What will the system do?
 - When will the system do it?
 - Are there several modes of operation?
 - What computations/data transformations must be performed?
 - What are the appropriate reactions to possible input?

Requirements Elicitation

- Good Questions to Ask
 - For Non-Functional Requirements
 - Are there constraints on execution speed, response time, or throughput?
 - What efficiency measures will apply to resource usage and response time?
 - How much data will flow through the system?
 - Should the access to the system/data be controlled?
 - Should user's data be isolated from others?
 - How often will the system be backed up?

Requirements Elicitation

- Good Questions to Ask
 - For External Interface Requirements
 - Is input coming from one or more other systems?
 - Is output going to one or more other systems?
 - Is there a prescribed medium that the data must use?
 - Is there a prescribed way in which input/output data must be formatted?

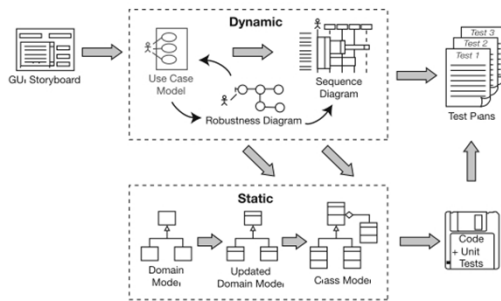
Requirements Elicitation

- Good Questions to Ask
 - For Physical Settings Requirements
 - Where is the product to be located?
 - Is there one location or several?
 - Are there any environmental restrictions?
 - Are there any constraints on the size of the system?
 - Are there any constraints on power, heating, or air conditioning?

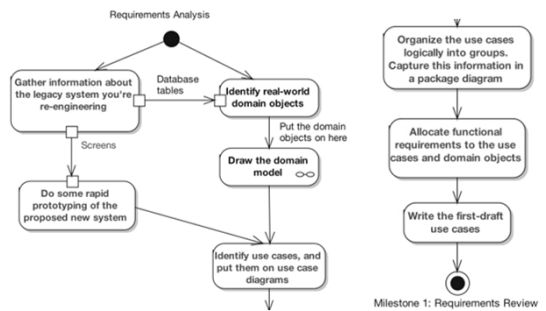
Requirements Elicitation

- Good Questions to Ask
 - For Development Constraint Requirements
 - Are there constraints on programming language because of existing software components?
 - What skills must the developers have?
 - What materials, personnel, or other resources are needed to build the system?
 - How much documentation is required?
 - Should it be on-line, in book format, or both?

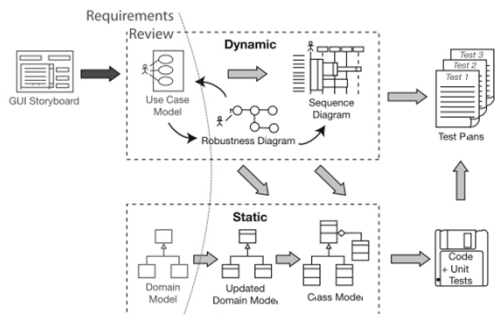
Use Case Modeling



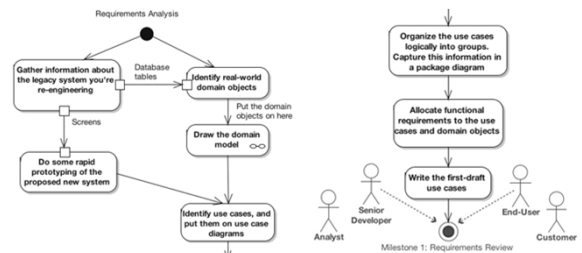
Use Case Modeling



Requirements Review



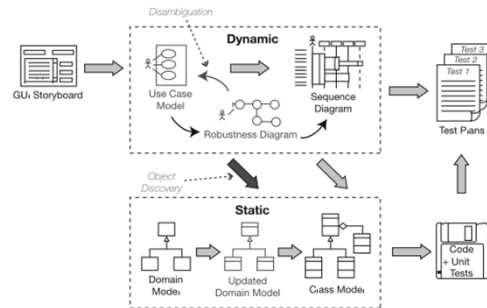
Requirements Review



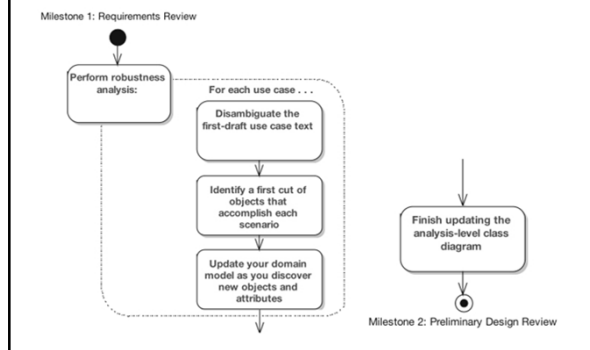
Project Phase 2

- Preliminary Design
 - Due: January 27, 2016
 - Description: Expressing user requirements
 - 2 paragraph rule, active voice, noun-verb-noun structure
 - Do not forget the alternate courses
 - Specify the actor
 - Use the object names in the system glossary
 - Deliverable
 - Use Case Diagram, Use Cases
 - Optional: GUI Storyboard/Wireframes

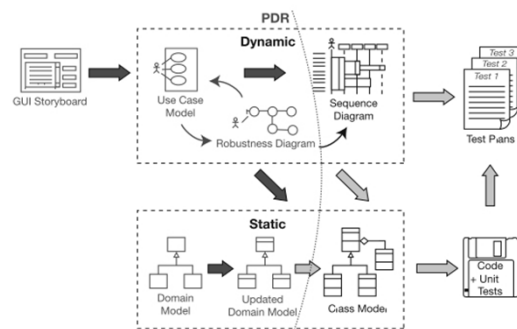
Robustness Analysis



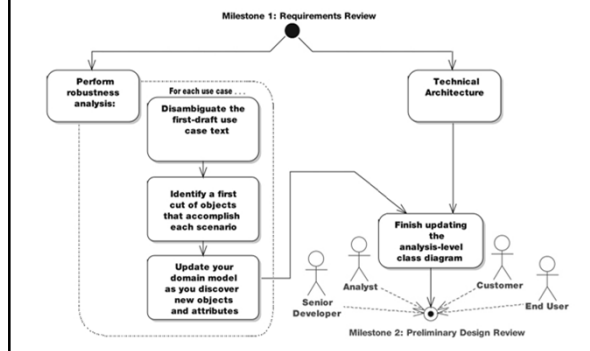
Robustness Analysis



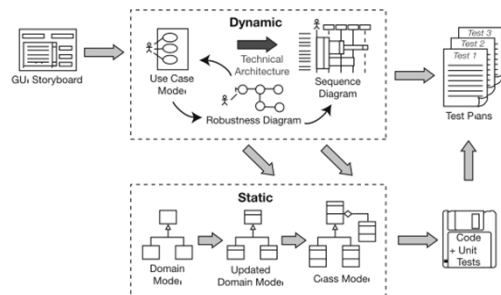
Preliminary Design Review



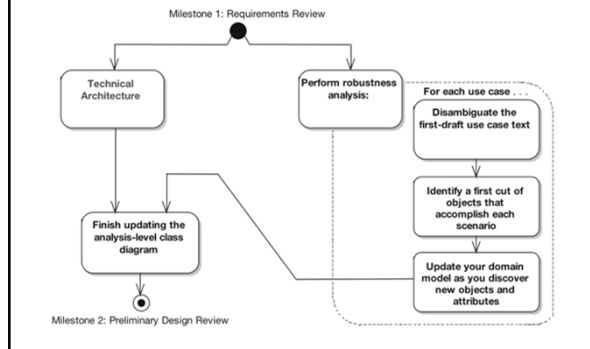
Preliminary Design Review



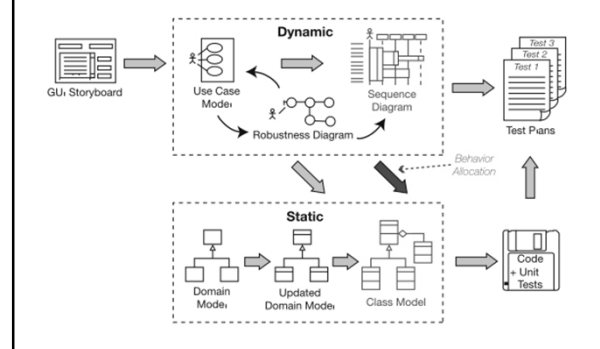
Technical Architecture



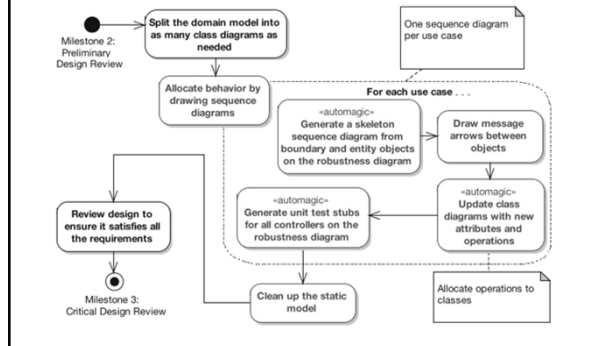
Technical Architecture



Sequential Diagram



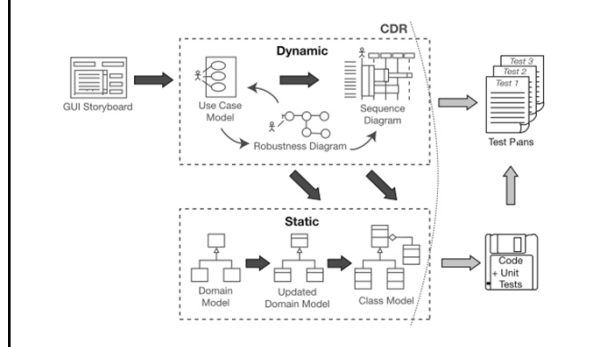
Sequential Diagram



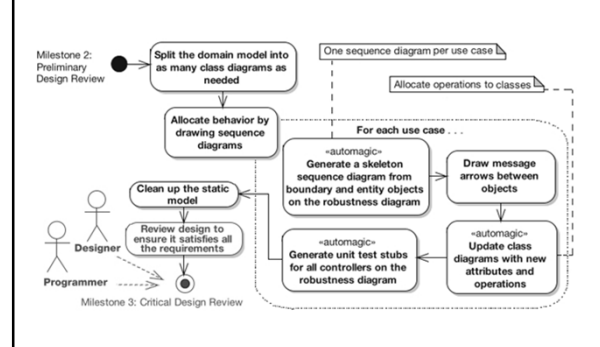
Project Phase 3

- Detailed Design
 - Due: February 3, 2016
 - Description
 - Translate each use case into sequence diagram
 - Assign behavior to objects
 - Refine class diagram
 - Deliverables
 - Sequence Diagram,
 - Static Class Diagram

Critical Design Review



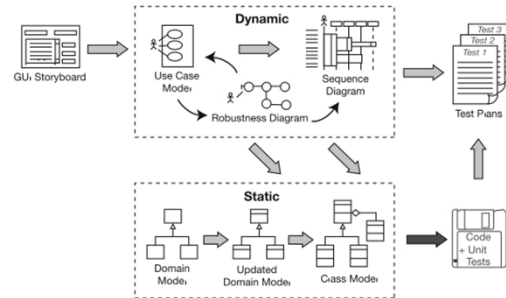
Critical Design Review



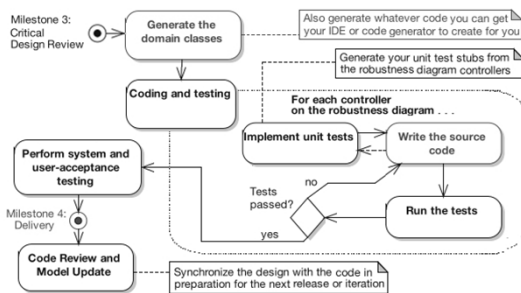
Project Phase 4

- Project Milestones
 - Due: February 8, 2016
 - Description
 - Create project schedule
 - Should include weekly milestones
 - Deliverables
 - Software process schedule

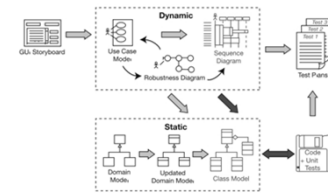
Implementation



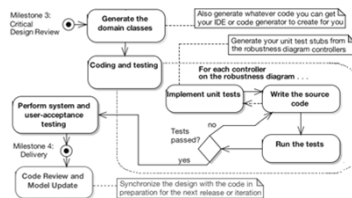
Implementation



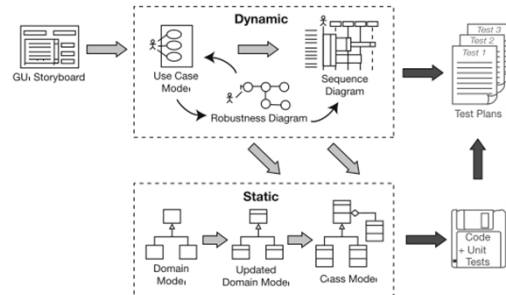
Code Review and Model Update



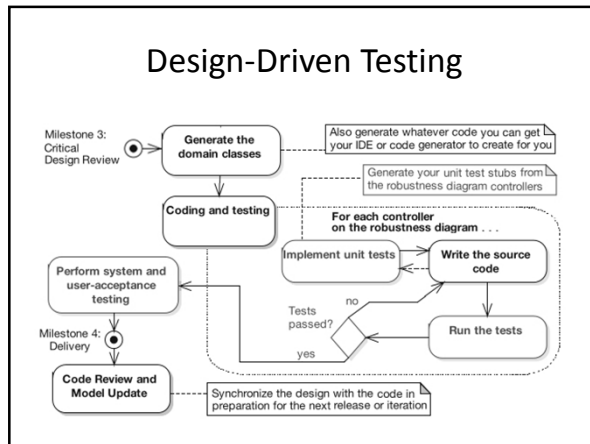
Code Review and Model Update



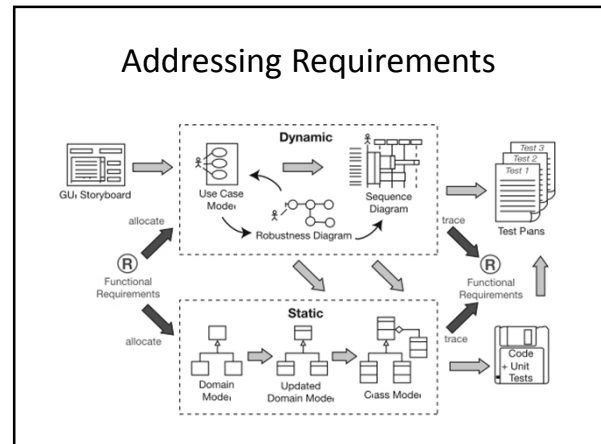
Design-Driven Testing



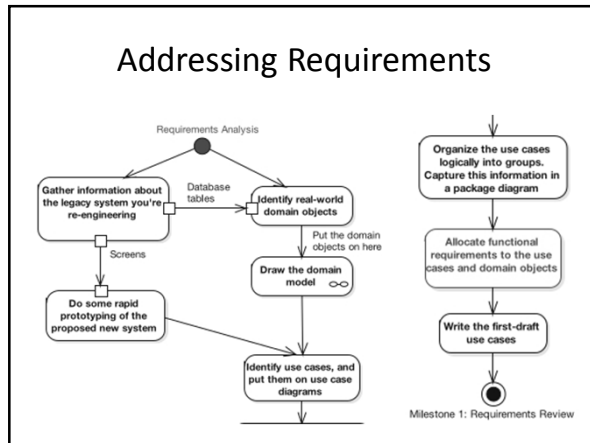
Design-Driven Testing



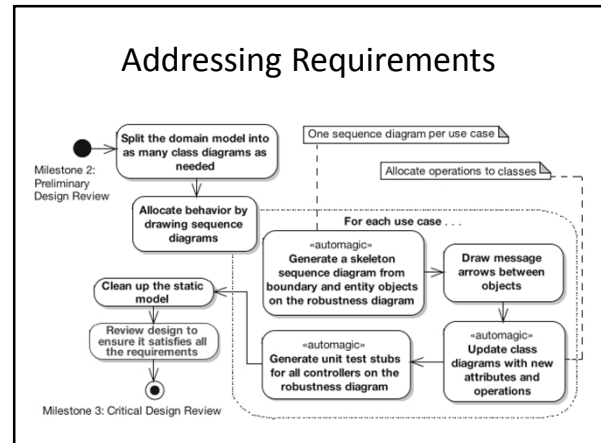
Addressing Requirements



Addressing Requirements



Addressing Requirements



Project Phase 5

- Coding and Testing
 - Due
 - February 10, 17, 24, March 2, 16, 23, 30, April 6, 13, 20
 - Description
 - Develop the product based on the scheduled milestones
 - Must be test-driven development
 - Document everything you do
 - Consistent coding style
 - Deliverables
 - Working prototypes of the product
 - Unit tests

Project Phase 6

- Promotional Website
 - Due: April 27, 2016
 - Description
 - Main page and About page
 - Download / Try-me page
 - Support page for users
 - System documentation page
 - Deliverables
 - Project website

Project Phase 7

- Complete System Delivery
 - Due: February 24, Final period, May 6, 2016
 - Description
 - Launch party
 - Capstone design conference
 - Deliverables
 - Project description (Feb. 24, 2006, Capstone Conference)
 - Complete system demonstration
 - Presentation / Exhibition