Requirements & Modeling

Chapter 2 p29—p31, Chapter 9, Chapter 11 p117--p120

Chapter 4.0—4.5

Chapter 8.1, 8.2.5, 8.4, 9.2, 9.3.1
Administrative stuff

- TA office hours
  - Monday 3:00—4:00 @ CSIL 4 (Yuxi) not every week
  - Tuesday 2:00—3:30 @ CSIL 1 (Lefan)
  - Wednesday 1:00—2:00 @ CSIL 1 (Yuxi) not every week

- Warm-up project
  - Due on Jan. 17th

- Project proposal
  - Due on Jan. 19th

- Quiz
Project Proposal

• You are required to work on this in a 2-person or 3-person group.
• You will brainstorm with your group members to propose a software project that a 8 person group, including you, will work on for the remainder of this quarter.

• What to submit:
• The whole group will submit one copy of the proposal document.
• This document needs to include the following items:
• 1. What programming language you plan to use;
• 2. What programming IDE you plan to use, if you plan to use any;
• 3. A brief overview of what you are proposing (be brief here)
• 4. A complete use-case diagram of the proposed project
• 5. Choose one of the following two options to describe every use case
• option 1: activity diagram, following the format we will discuss in lecture
• option 2: use case text description, following the format we will discuss in lecture
• 6. Optional: tell us whatever you think can help convince us to accept your proposal
A few project example

• Proposal examples
  • ...

• Repository examples
  • https://github.com/catherinemoresco/PDFProject
  • https://github.com/courageousillumination/deckr
  • https://github.com/dyxh/cs220
  • https://github.com/marlonliu/DivAssist
Course Project Grading

• Group performance
  • 75%

• Individual performance
  • Commit log
  • Self-evaluation + peer-evaluation
    • After milestone 3.b
    • After milestone 5
Outline

• Definitions
  • Requirement, requirement engineering

• Why?

• How to write the requirement document?

• How to find out and model the requirements?
How to find out & represent the requirements
System modeling
Use case diagrams

• A diagram includes
  • Actors
  • Use cases
  • Associations
  • System boundary
How to describe use case?
Use case text

• Use case name
• Main scenario
  • Steps
• Extensions
  • Extension condition; steps
Use case text

• Use case name
• Main scenario
  • Steps
    • Each step clearly shows who (actor/system) is carrying out what action
• Extensions
  • Extension condition; steps
    • Think about what could go wrong or go differently

• Specify what to do, not how to do
• Do not specify user interface
• Optional: priority, trigger, precondition, postcondition (guarantees), sub-usecase
Example template

[use case name]
Main success scenario:
1. X does s
2. X does t
3. ...
4. ...

Extensions:
2.a [extension condition]
.1 xxx
.2 xxx, return to MSS at step 4
3.a [extension condition]
.1 xxx
.2 X may xxx or cancel
Example

Game
Main success scenario:
1. User chooses a puzzle
2. User works on a puzzle
3. User submits the puzzle result
4. System grades the result
5. System displays the result

Extensions:
2.a User decides to give up on the puzzle
   .1 User aborts the current puzzle, return to MSS step 1
5.a Top 10 grade
   .1 System displays the result and a congratulation message
   .2 System updates the top result record
How to map it to eXtreme Programming?

• Use case or sub-usecase is similar with the user story
Activity diagrams

• An activity --- multiple actions
  • Can be used to describe a use case
  • Can represent parallel relationship
Activity diagram components

• Components
  • Start
  • Actions
  • Fork/Join
  • Decision/Merge
  • Flow
  • Final
Example

Pick a Q

Work on a Q

Abort [give up game]

Change Q [give up Q] [finish]

Submit A

Succeed [correct] [incorrect]

Fail
Example

Pick a Q

Count down

Work on a Q

Abort [give up game]

[give up]

Change Q [finish]

Submit A

Succeed [correct]

[incorrect]

Fail
To Finish in the Next Lecture
Definitions & Motivations
What are requirements?
What are requirements?

- The services the software should provide
- The constraints the software should follow
- Functional requirements
- Non-functional requirements
What is requirement engineering (RE)?

• The process of
  • Finding out
  • Analyzing
  • Documenting
  • Checking
these desired services and constraints
is Requirement Engineering
Who will read requirement document & why?
Who will read requirement document & why?

• Users
• Design team
• Developers
• Testing team
What to put in a requirement document?
What are the requirements?

• Functional requirement

• Non-functional requirement
What are the requirements?

• Functional requirement
  • Specify functionality
  • Input, output, ...

• Non-functional requirement (how to measure them quantitatively?)
  • Performance
    • Time complexity, space complexity, scalability, throughput, latency, space
  • Security
  • Usability
  • Power & energy
  • Legal, ethical
  • Dependability
    • Security
    • Availability = available time / (service available time + service down time)
    • Reliability = how likely the service will go town at time T
Non-functional requirements

- Product requirements
  - Efficiency requirements
  - Reliability requirements
  - Portability requirements
- Organizational requirements
  - Interoperability requirements
  - Ethical requirements
- External requirements
  - Usability requirements
  - Delivery requirements
  - Implementation requirements
  - Standards requirements
  - Legislative requirements
  - Performance requirements
  - Space requirements
  - Privacy requirements
  - Safety requirements
Non-functional requirements

• Try to use quantitatively measurable metrics to describe them
• Examples
Requirement document format

• IEEE standard
  • http://en.wikipedia.org/wiki/Software_requirements Specification