SE463
Software Requirements Specification & Analysis

Review

Scoping

Recognition

Terminology: project purpose, constraints, requirements
Difference between the Work and the System to be built
Different types of stakeholders

Comprehension

The knowledge and contributions that each stakeholder type brings to the table

Application

Stakeholder analysis for a particular problem
Draw a context diagram
Draw a use-case diagram

Elicitation

Recognition

Challenges of requirements elicitation
Different elicitation techniques
Lightweight modelling notation syntax and semantics
Quality requirements, fit criteria
Rationale behind user stories

Comprehension

Applicability of elicitation techniques to an elicitation problem

Application

Draw activity diagram
Draw process model
Write user stories
Devise quality requirements for a system

Requirements Analysis

Recognition

Strategies for resolving conflicts among requirements (or people)
Benefits, challenges of prioritizing requirements
Prioritization techniques
Estimates vs. targets vs. commitments
Software estimation techniques

Comprehension

Applicability of different resolution strategies
Applicability of different prioritization techniques
Influences on software size and cost

Application

Use resolution techniques to suggest a resolution to a conflict
Perform steps of a prioritization technique
Perform steps of a risk analysis
Perform steps of a software estimation

Requirements and Specifications

Recognition

Requirements vs. specification

The environment of a system vs. interface phenomena

Assumptions

Atomic requirements, conditions of satisfaction

Comprehension

The vocabulary used to express requirements vs. the vocabulary used to express specifications

Application

Devising atomic requirements or conditions of satisfaction

Deriving specifications from requirements

Identifying assumptions that need to hold in order for a system that meets the specification will also meet the requirements

Modelling and Documentation

Recognition

UML class diagrams syntax and semantics
State machine modelling syntax and semantics
Rationale behind EARS

Comprehension

Read and understand OCL expressions over domain models
Read and understand state-machine models

Application

Write scenarios, EARS requirements
Draw a requirements domain model that represents entities, relationships, attributes, etc. used in the *requirements* of the system
Draw a specification domain model that represents entities, relationships, attributes, etc. used in the *requirements* of the system
Create user story map
Draw a state machine model
Draw wireframe, navigation map

Validation and Verification

Recognition

Validation vs. verification

Walkthroughs, reviews, formal inspections

Testing of requirements

Comprehension

Reviews vs. testing of requirements

RE in Practice

- Colin Rhodes (Roadmunk)
- Dave Cliffe (PagerDuty)

How to Study

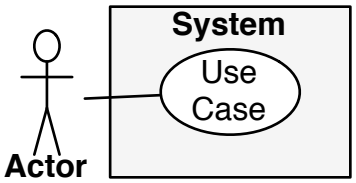
Final Exam:

Wednesday August 3
7:30-10:00
PAC 7

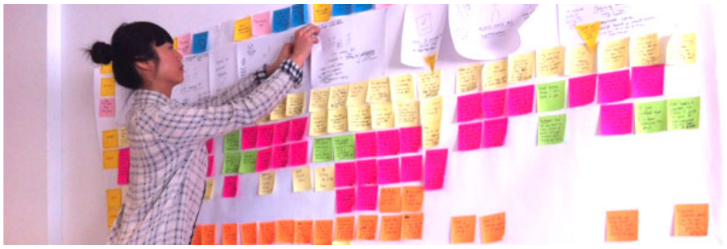
Application-style questions will be 75%-85% of the exam. So the best way to study is to do practice exams:

https://www.student.cs.uwaterloo.ca/~se463/examInfo.shtml

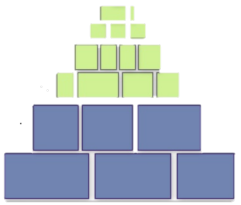
Consumer Products



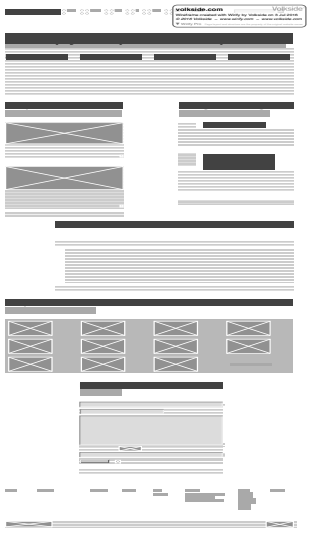
elicited functionality



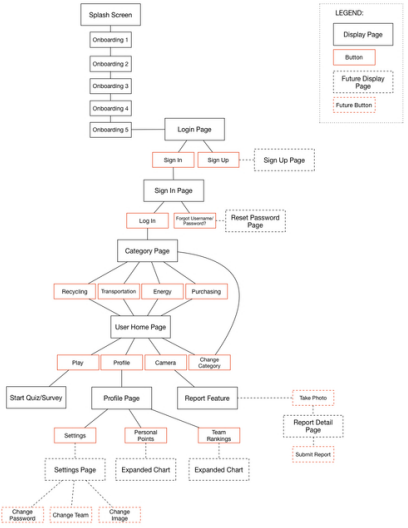
user stories, roadmap



prioritized backlog

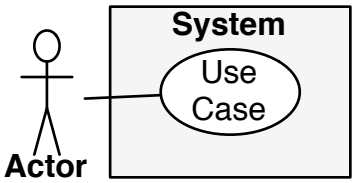


wireframes



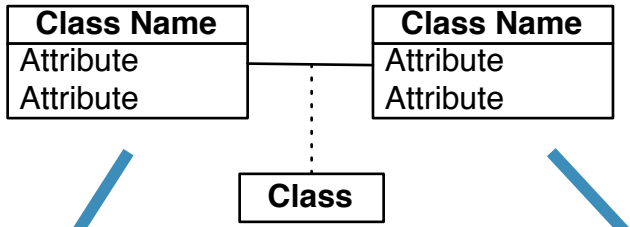
navigation map

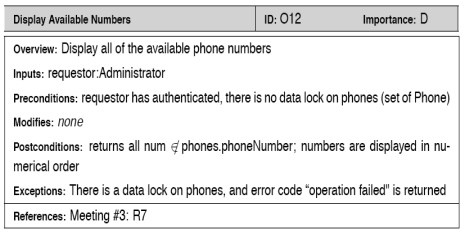
Enterprise Systems



elicited functionality

domains and relationships





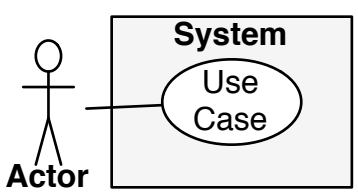
Display Available Numbers	ID: O12	Importance: D
Overview: Display all of the available phone numbers		
Inputs: requestor:Administrator		
Preconditions: requestor has authenticated, there is no data lock on phones (set of Phone)		
Modifies: *none*		
Postconditions: returns all num ∉ phones.phoneNumber; numbers are displayed in numerical order		
Exceptions: There is a data lock on phones, and error code "operation failed" is returned		
References: Meeting #3: R7		

system operations that modify domain

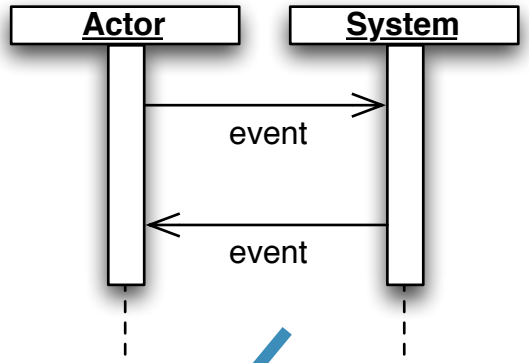
context Membership **inv**
self.transactions->collect(points)->sum = self.points

constraints on data values, relationships

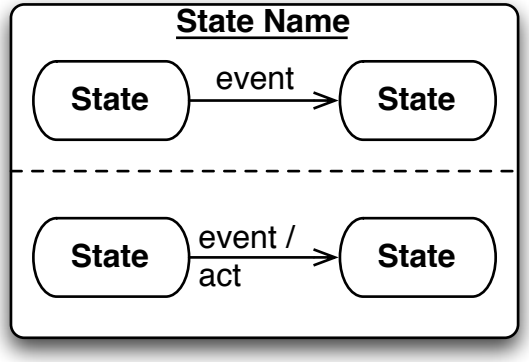
Reactive Systems



elicited functionality



dialog between system and environment



dialog between system and environment

$\Box(\neg(moving \wedge (door = open)))$

constraints on behaviour