SENG321: Requirements Engineering

INTRODUCTIONS

Dr. Daniela Damian and Zane Li

thesegalgroup.org

danielad@uvic.ca

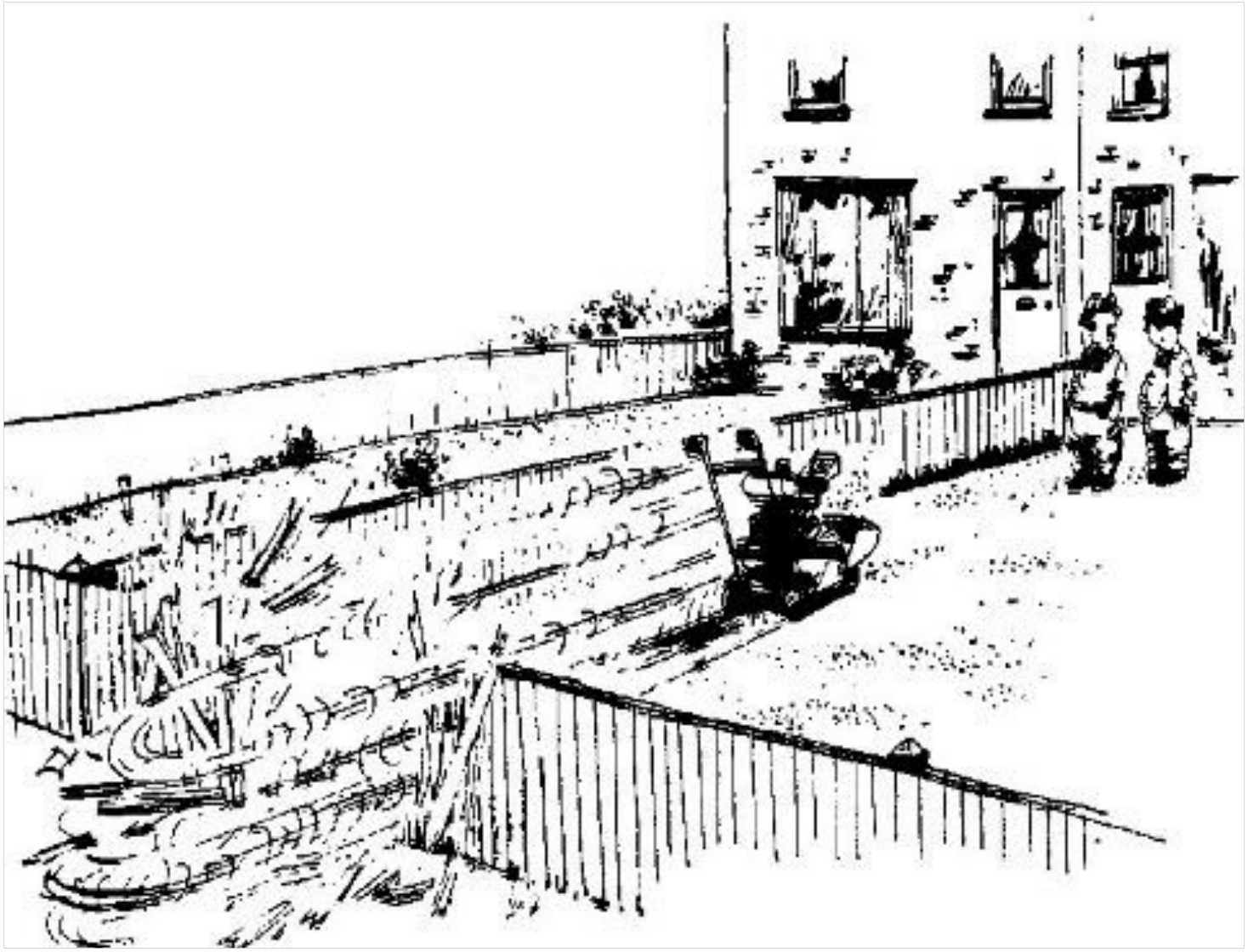
Outline of first class

What is Requirements Engineering (RE), importance and difficulty

Introductions

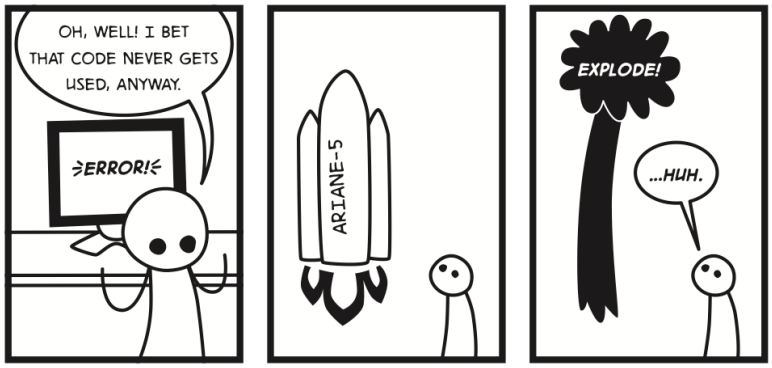
Why are we here: your opportunity

Administrative: course structure and Expectations



The problem is, it's been programmed to cut a longer lawn

. . . more seriously, Safety-critical systems



Other Examples

- Denver Airport's Automated Baggage System
- healthcare.gov initial launch (2013)
  - Poorly defined requirements
- AWS down during Prime Day for 63 minutes (2018)
  - Non-functional requirements

non-functional systems?

Requirements impact all other artifacts and phases in the project

GIF image 600x213 pixels http://www.dilbertzone.com/comics/dilbert/archive/images/dilbert20

RE: a critical activity in software development

Causes:

incomplete requirements (13.1%)

lack of user involvement (12.4%)

lack of resources (10.6%)

unrealistic expectations (9.9%)

lack of executive support (9.3%)

changing requirements and specifications (8.7%)

lack of planning (8.1%)

system no longer needed (7.5%)

[Standish Group Study, 1995]

RE: a critical activity in software development

(traditional view…)

A poor requirements process can be expensive! The cost to find and fix a requirements-based problem during:

- … requirements definition is $1
- … design is $5
- … coding is $10
- …unit testing $20
- … and as much as $200 after the delivery of the system!

Can you get all the requirements right from the beginning?

Requirements Engineering course

Requirements Engineering: beyond (software) requirements, throughout development, and pervasive in our lives

Improvement in RE processes significant to industry

Training in RE:

- Critical industry need
- Offers competitive advantage

Introductions

Who is your teaching team?

Dr. Daniela Damian

Professor of Computer Science and PEng
ECS-CAPI Chair in Inclusive Science and Engineering
PhD in Requirements Engineering
Extensive work with industry partners (IBM, Siemens, Dell, Unisys, startups)

Zane Li (Co-Instructor and TA)

PhD candidate, Topic: Requirements Engineering

Kezia Devathasan (TA)

PhD student, Topic: Software Engineering

SENG321

Emphasis on requirements as the negotiated product of understanding the market and business needs

Dealing with ambiguity, uncertainty

Develops skills for:

- Analysis and Contract negotiation
- Solution seeking in context of customer problem
- Choosing the right requirements technique for particular projects, organizational setting and development process model
- Requirements validation for solution fit

SENG321

Teaching it is difficult, feels theoretical and abstract

In this course:

Emphasis on the **requirements *knowledge*** and ***its management,*** not as much on requirements specification

Experiential learning to apply Concepts/ Techniques from class

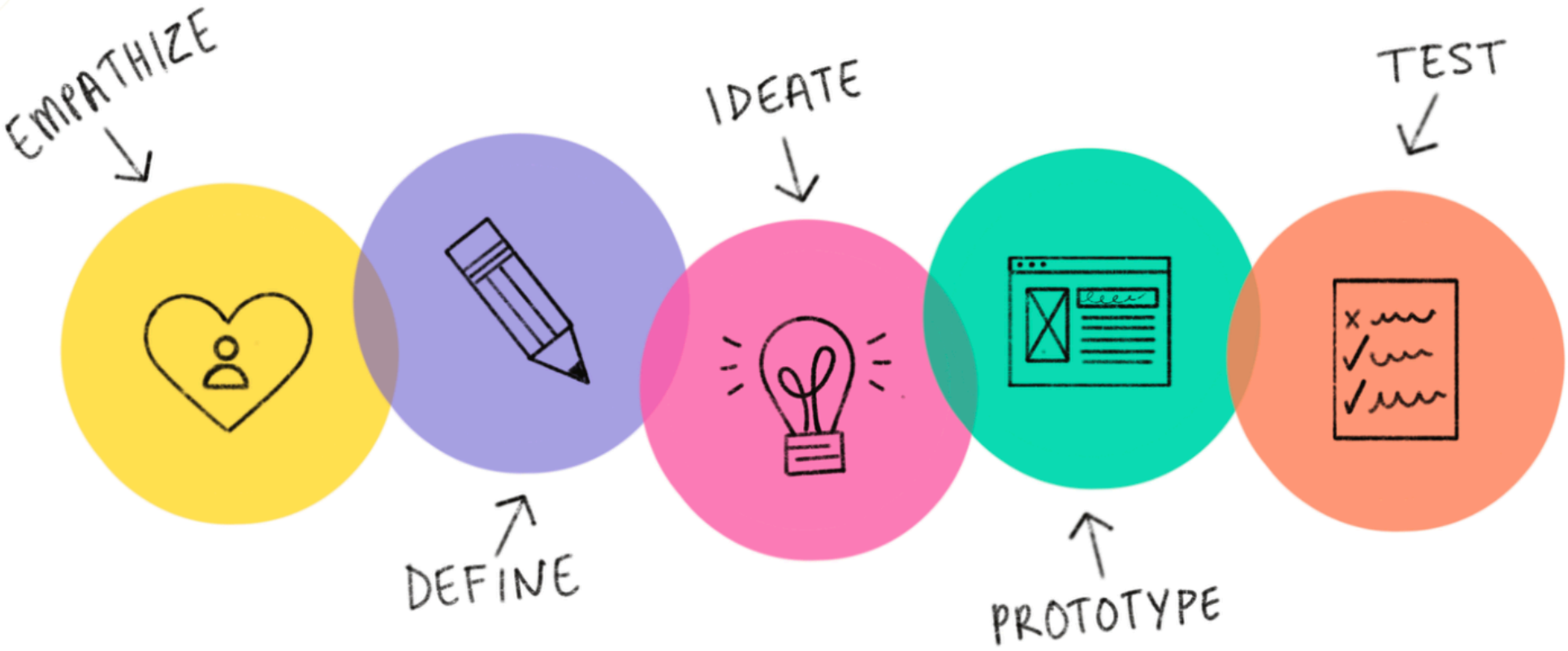
Term project – emphasis on user-centred approaches

Readings essential

Term Project

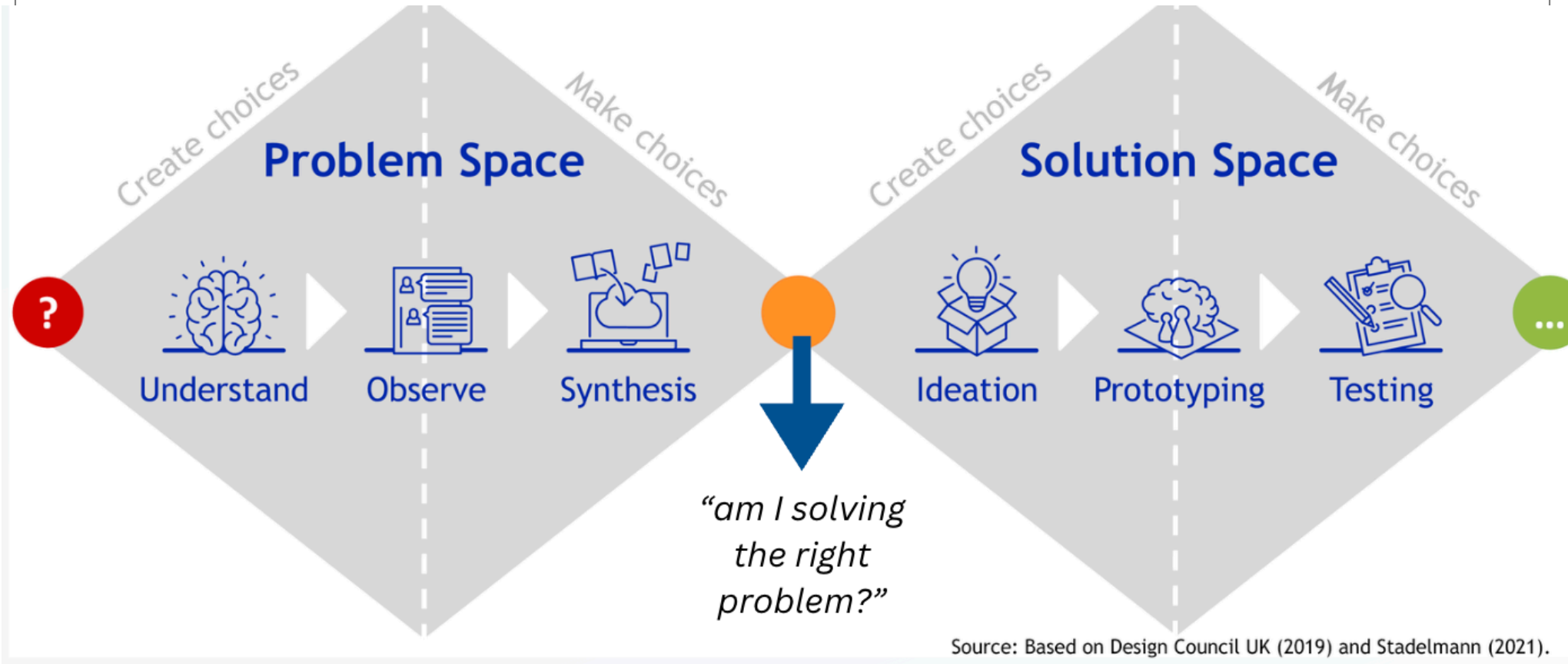
Design thinking – as the user-centered approach in our project

Design Thinking



Design Thinking

Design Thinking



Source: Based on Design Council UK (2019) and Stadelmann (2021).

What is DT?

-Human centred

-Highly Creative

-Collaborative

-Iterative

-Hands-on

-Show, Don't Tell

- A process that celebrates failure

Design Thinking Example

AN EXAMPLE OF PROBLEM SOLVING: THE ENCUMBERED VS. THE FRESH MIND

Some years ago, an incident occurred where a truck driver tried to pass under a low bridge. But he failed, and the truck was lodged firmly under the bridge. The driver was unable to continue driving through or reverse out.

The story goes that as the truck became stuck, it caused massive traffic problems, which resulted in emergency personnel, engineers, firefighters and truck drivers gathering to devise and negotiate various solutions for dislodging the trapped vehicle.

Emergency workers were debating whether to dismantle parts of the truck or chip away at parts of the bridge. Each spoke of a solution which fitted within his or her respective level of expertise.

A boy walking by and witnessing the intense debate looked at the truck, at the bridge, then looked at the road and said nonchalantly, "Why not just let the air out of the tires?" to the absolute amazement of all the specialists and experts trying to unpick the problem.

When the solution was tested, the truck was able to drive free with ease, having suffered only the damage caused by its initial attempt to pass underneath the bridge. The story symbolizes the struggles we face where oftentimes the most obvious solutions are the ones hardest to come by because of the self-imposed constraints we work within.

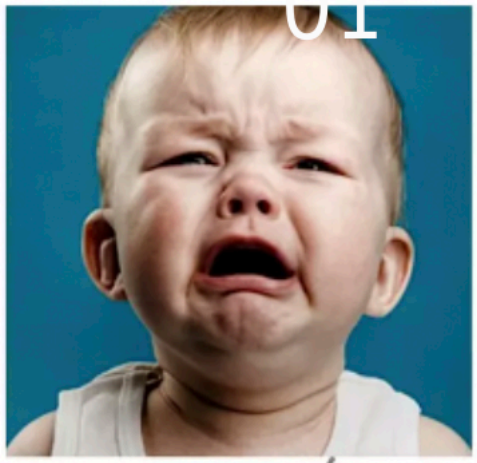
Design Thinking Example

SUCCESS STORY 2 – GE HEALTHCARE

MRI For Children

SOLUTION

01



MRI scans require a person not to move, but little kids cry and move around.

02

By immersing in the experience of a kid they learned that …

03

… for a kid an MRI room must be very stressful and a frightening experience.

04

Kid-friendly MRI. Simple commands to get the scan done accurately become part of an adventure.

Term Project (or how we use Design thinking)

Uses *agile methodology* and *iteration*-based process

Fast pace

Teamwork as *Clients* and as *Developers*

Significant interaction between developers and clients

Uncertainty is the only certainty is the Project

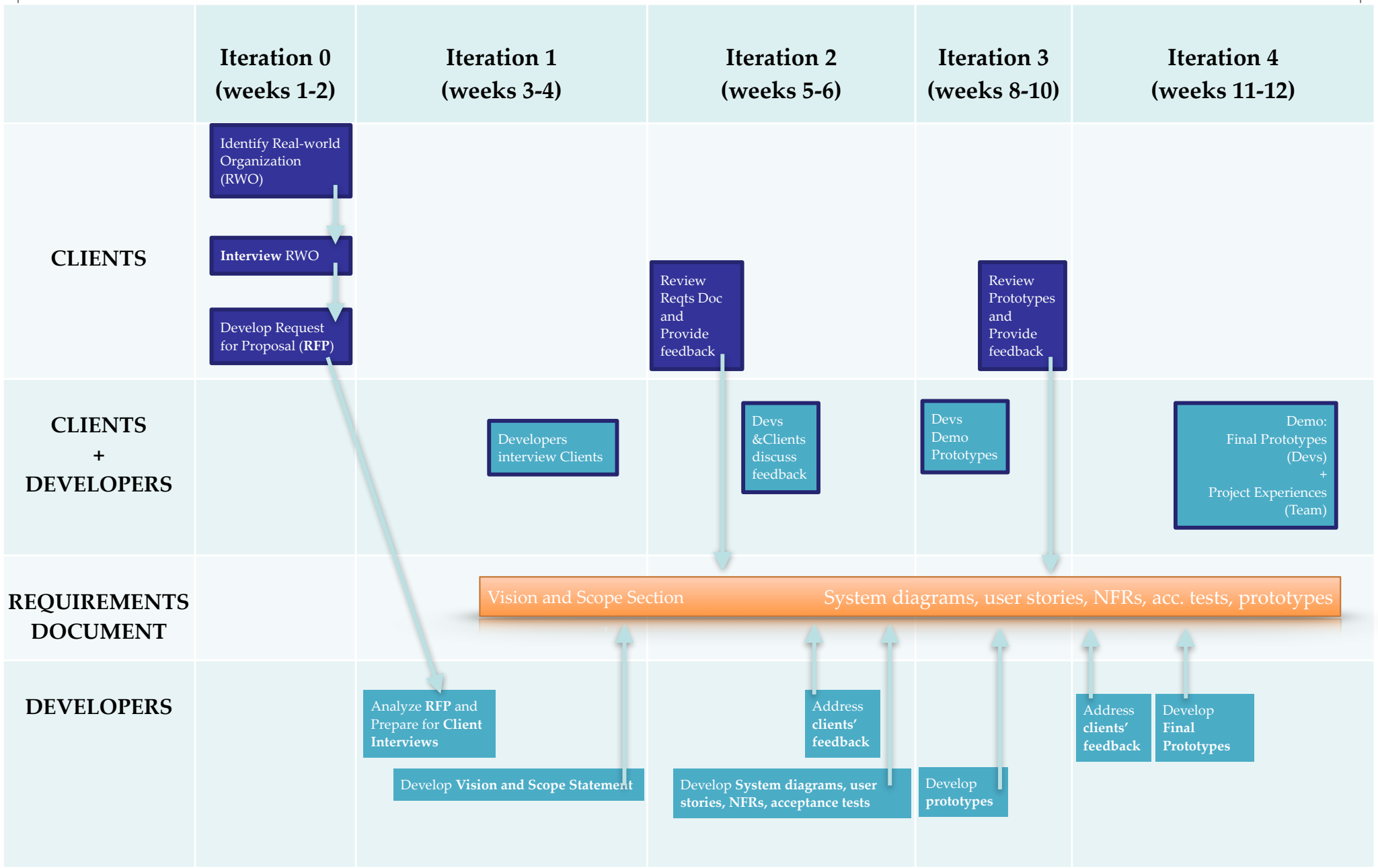
Need to keep up with **readings**, **lectures** and the **Project teamwork** *every week*

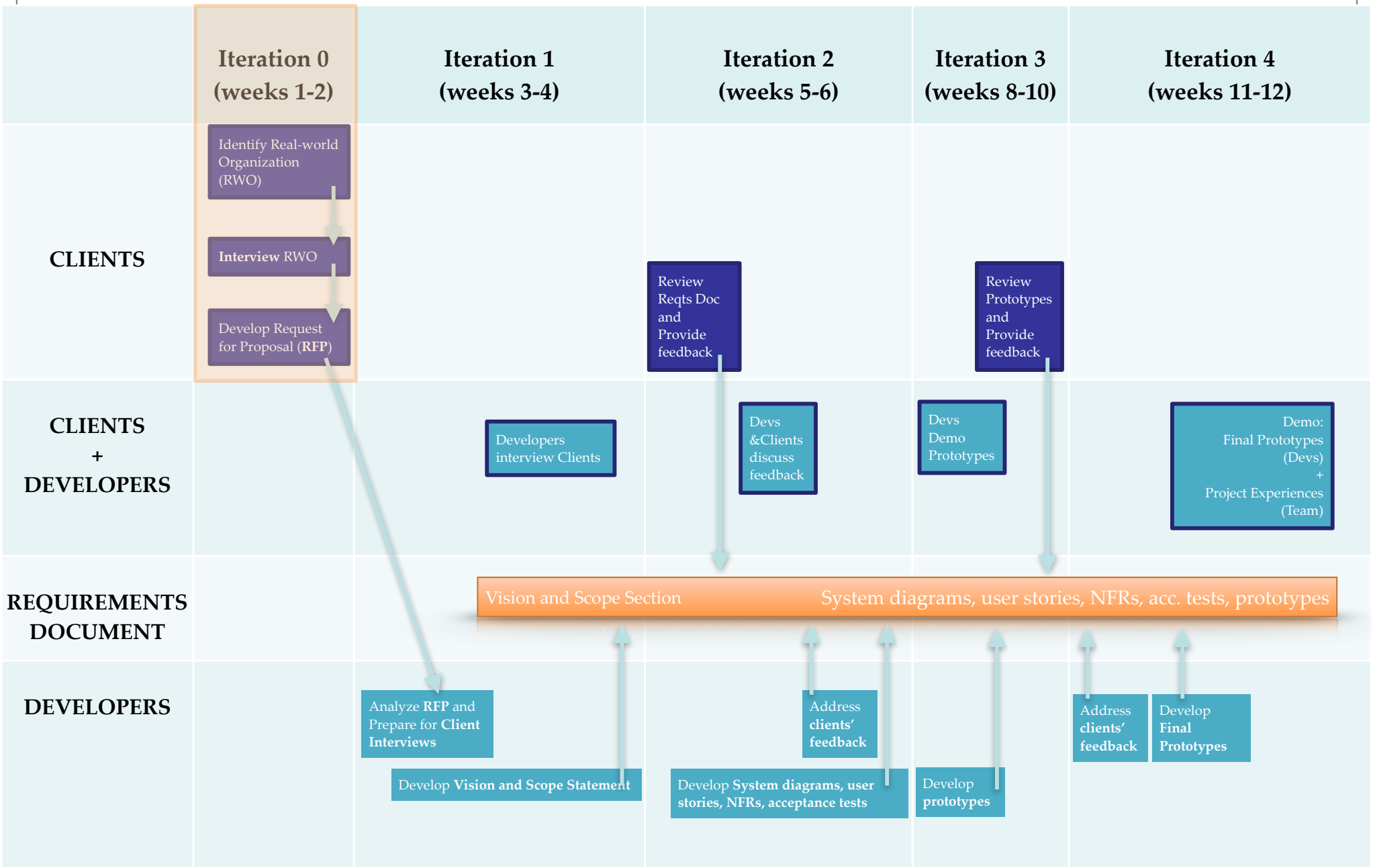
Major Project Component

Goal: apply RE techniques in a realistic setting and in a professional client-designer relationship

- requirements gathering and analysis
- problem understanding
- requirements validation through rapid solution prototyping
- interaction with clients in realistic situations

Detailed Project Specification in Brightspace/Github


Iteration 0 (weeks 1-2)
Iteration 1 (weeks 3-4)
Iteration 2 (weeks 5-6)
Iteration 3 (weeks 8-10)
Iteration 4 (weeks 11-12)
CLIENTS
CLIENTS + DEVELOPERS
REQUIREMENTS DOCUMENT
DEVELOPERS
Identify Real-world Organization (RWO)
Interview RWO
Develop Request for Proposal (RFP)
Review Reqts Doc and Provide feedback
Review Prototypes and Provide feedback
Developers interview Clients
Devs &Clients discuss feedback
Devs Demo Prototypes
Demo: Final Prototypes (Devs) + Project Experiences (Team)
Vision and Scope Section
System diagrams, user stories, NFRs, acc. tests, prototypes
Analyze RFP and Prepare for Client Interviews
Develop Vision and Scope Statement
Address clients' feedback
Develop System diagrams, user stories, NFRs, acceptance tests
Develop prototypes
Address clients' feedback
Develop Final Prototypes


Iteration 0 (weeks 1-2)
Iteration 1 (weeks 3-4)
Iteration 2 (weeks 5-6)
Iteration 3 (weeks 8-10)
Iteration 4 (weeks 11-12)
CLIENTS
Identify Real-world Organization (RWO)
Interview RWO
Develop Request for Proposal (RFP)
Review Reqts Doc and Provide feedback
Review Prototypes and Provide feedback
CLIENTS + DEVELOPERS
Developers interview Clients
Devs &Clients discuss feedback
Devs Demo Prototypes
Demo: Final Prototypes (Devs) + Project Experiences (Team)
REQUIREMENTS DOCUMENT
Vision and Scope Section
System diagrams, user stories, NFRs, acc. tests, prototypes
DEVELOPERS
Analyze RFP and Prepare for Client Interviews
Develop Vision and Scope Statement
Address clients' feedback
Develop System diagrams, user stories, NFRs, acceptance tests
Develop prototypes
Address clients' feedback
Develop Final Prototypes

SENG 321 Requirements Engineering Project Specification

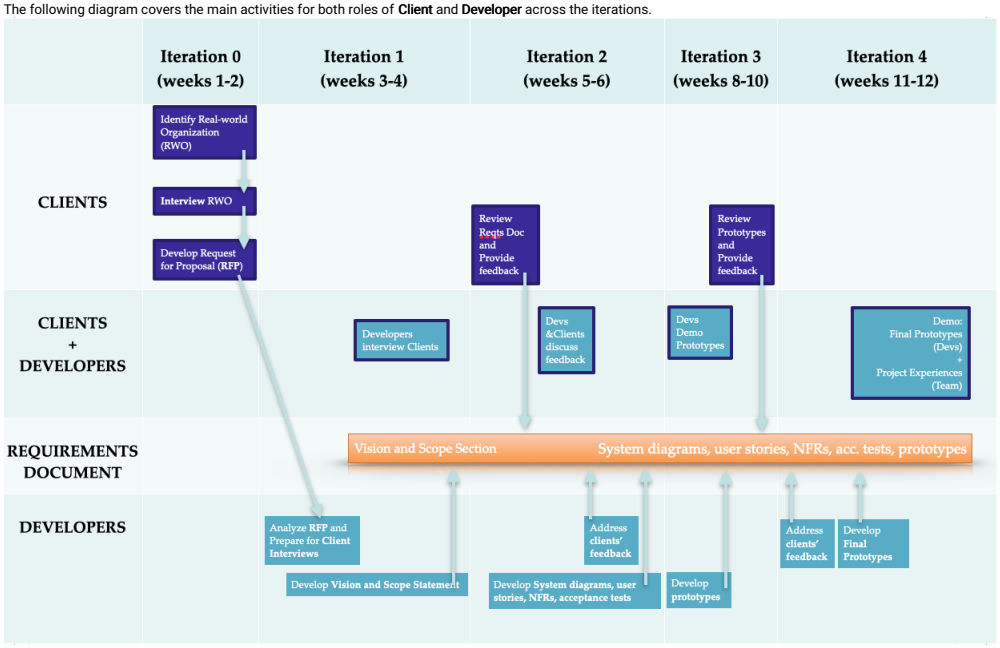
Objectives

1. Apply the RE techniques learned in the course in a team project focused on realistic problems.
2. Gain experience with developing and maintaining a software requirements specification as a living document in a software project.
3. Gain experience with the activities and expectations of the roles as clients as well as developers in a software project.
4. Promote student motivation, professionalism, and self-organization by interacting with real stakeholders.
5. Practice interviewing and stakeholder management skills with real stakeholders and other student groups.
6. Improve project presentation skills through in-class demos and updates.
7. Develop strong prototyping skills.

Context and Management

The requirements documentation in a project is a most powerful tool to document the knowledge, expectations and features of a software system. When managed as a living document, it records the understanding of the needs, the business and user requirements, features and non-functional requirements. It is also a record of decisions made through negotiations between clients and developers of the system. For this project you will be working in a 6-8 student project team in both roles of a **Client** and a **Developer** in **two different projects**. The requirements documentation is developed incrementally and iteratively thoughout the term project and it is the ultimate project deliverable is the course. Clarification: Your team is the same (but play different roles) in the two projects.

The following diagram covers the main activities for both roles of **Client** and **Developer** across the iterations.



The project follows an agile methodology and iteratively develops a number of artifacts such as a **Request for Proposal (RFP)**, **Requirements Documemnt (RD)**, Prototypes and in-class Presentations. **All your project work, must be uploaded to your GitHub repo by the due dates given. See below for instructions on each deliverable, due dates, and instructions on how to use GitHub.**

Working as a Client

In order to create realistic requirements for a developer team, you, as a client team, must first write an **RFP (Request For Proposal)**. This involves finding a real world organization (RWO) in our community, and gathering a set of requirements based on a problem that they need solved. For example, perhaps a pharmacy in your neighbourhood needs a better system to acquire and manage information about medications to be dispensed. Specific guidelines on how to choose a RWO, and create the **RFP** can be found below in Deliverables.

To ensure you find realistic requirements, your first task is to find a RWO, introduce yourself, and identify a problem in the organization that can benefit from a software solution. You will conduct as many interviews as needed to become familiar with the organizationâ€™s system and problem. Subsequently, you will act as representatives of this organization for the duration of this project. Thus, your first deliverable is the **request for proposal (RFP)**. Be sure you know enough about the software RWO and its operational context, as well as objectives for improvement in order to write the **RFP**. Once you hand in your **RFP**, a **different** team will be assigned to be the corresponding **Developer team** for the problem that is defined in your **RFP**. As representatives of the problem that can benefit from a software solution, you will act as clients to the Developer team for future deliverables. This involves defining in more detail the requirements and constraints for a solution system, as well as other expectations that your organiztion has of the Developer team (i.e. project timing, interaction mode, etc.) The Developer team will then read and understand **your RFP** and throughout the project will interview your organization, "the Client", on several occasions and as needed (see Schedule). During these interviews, you will provide information regarding your organization and system. Furthermore, you will provide feedback for the prototypes built by the designer team.

Project Deliverables and Evaluation

Main Deliverables

- **Requirements Document**
- **Reflection** on the Experience
- In-Class **Presentations**
- **Assessment** of Group member ***individual*** **performance**

Evaluation criteria:

Quality criteria for documentation

Teamwork, initiative, commitment, professionalism

No single way of doing it…

Course Schedule

See Google doc

A rich resource that shows the calendar of Lectures, Labs and Project activities, Deliverables and Due dates.

To be bookmarked!

Brightspace and GitHub

Course GitHub repo for Project info and work

Course Schedule (Lectures, Labs, Project activities and Deliverables) in Google Doc

Course marks

7 quizzes (tentative dates published – will keep 6/7)	30%
Project	60%
Class Participation	10%

You must PASS EACH Quiz component and Project to pass the course. The project grade accounts for individual contribution and does not rely solely on your group performance. Individual contribution is assessed monthly (see Contribution to Team Assessment)

Text and Readings

Required text, see **Textbook** in Readings (PDF in Bright)

"Software Requirements, 3rd Ed." by K.Wiegers and J. Beaty

Readings required BEFORE class, for effective class discussions and for the quizzes!
Material essential for Quizzes (30%) and Class Participation (10%)

Lots of readings, provide details and background, examples to complement the lectures.

Front loaded in the course, most in January/February!

Marks Breakdown

SENG 321 Requirements Engineering

Assignment	% Worth	Who's responsible?
INDIVIDUAL (total: 40%)		
6 quizzes	30%	Individual
Class Participation	10%	Individual
PROJECT (total: 60%)		
Client Role	**13%**	**Group**
Request for Proposals (RFP)	6%	Group
Reviews of Requirements Document (incl. Prototypes)	5%	Group
Professionalism	2%	Group
Developer Role	**35%**	**Group**
Requirements Document	28%	Group
Final Presentation	5%	Group
Professionalism	2%	Group
Group Dynamics	**12%**	**Individual**
Teamwork	7%	Individual
Team Assessment reports	**5%**	Individual
Total	**100%**	

SENG 321 Requirements Engineering Contributions to project Teamwork

To be completed in a survey (link will be provided prior to deadline)

These reports are not optional.

Each student is required to create several reports documenting the assessment of each team member of your group, including yourself. There will be an assessment **after each project iteration.**

Each report should document who did what during the iteration for each person, and your assessment of that person, again, including yourself.

Suggested criteria for assessing contribution to the group (client and description) work include:

- communication style
- completion of assigned tasks
- participation in meetings
- contribution to discussion
- reliability of contribution to group work
- attendance in general

Guidance for constructing your feedback:

- include group number, email, full name, and student number
- each assessment should be one paragraph (roughly) per person in your team, **including yourself**.
- You will be marked on how perceptive, insightful, and reasonable your comment are.
- Provide a **letter grade** per person, alongside your paragraph-long justification.
- Do not give all A+s (or any other grade) for every team member-- this is not appropriate, and is obvious to the teaching team that a lack of effort went into your assessment.
- **Do NOT share your report with anyone. It is confidential**

  Grading: the quality of writing in these reports is assessed as 5% of the overall course grade. Furthermore, the content of these reports provide information about each team member's individual contribution to the project and will be used to adjust the overall course grade for an underperforming or overperforming team member, to reflect the marking of individual perofrmance as defined in the course outline.

Class format and participation

Most classes include both Theory (lectures) and In-class project work!

Attendance in project-related activities is mandatory, participation in class and projects part of the course mark (**10% participation mark**, mere attendance to classes does not guarantee any portion of this grade).

Communication is key to success. We will use **Teams** for project communication.

SENG 321 Requirements Engineering Assessment of Class Participation

Your participation grade is ongoing and begins in Week 1. You will receive a final participation grade at the end of the course. Feedback on your own class participation will be provided course midterm upon request. You will be graded on your participation in class activities, including showing up on time, attending all classes, completing class activities, reports and presentations, and discussions. Class participation includes not only being present in class, but fully contributing by engaging in discussions, activities and readings in class and outside of class. Your contributions should make explicit links to course readings and concepts. Note: mere attendance to classes does not guarantee any portion of the class participation grade.

Effective participation does not mean "speaking more or less" but rather focusing on the balance and quality (i.e. thoughtfulness and critical thinking) of contributions. Engaged students will be mindful of not monopolizing class discussions and group work. Reserved or quiet students will be mindful of taking risks and increasing their engagement in the class and in group work. The participation grade evaluates your participation; it does not evaluate you as a person. Students are not graded on their personality but rather on their ability to engage in the course is a comprehensive and professional manner. Your participation grade will be assessed based on the following 2 criteria:

Criteria 1: Critical Thinking demonstrates your ability to integrate your own experiences and perspectives into the course content (readings, lectures, discussions, guest speakers, etc.). This involves providing thoughtful questions and comments that demonstrate that you have been doing the readings, and thinking critically about how it relates to your course project. The critical thinking criteria assesses your ability to engage with new ideas and material, and to integrate readings and course discussions into your work. It also includes your capacity to integrate constructive feedback into your learning.

Criteria 2: Effective communication and Involvement means being fully 'present' in class, paying attention to class schedules, participating in activities and discussions, coming and leaving on time, attending all classes unless you have a valid excuse, and completing readings in an effective and timely manner. Effective involvement also includes your capacity to demonstrate readiness for professional engagement in the field. Professionalism means acting and dressing appropriately for professional meetings and work in agencies and with clients; and listening and responding respectfully to diverse viewpoints (seeking first to paraphrase other viewpoints before expressing your opinion, in a way that encourages respectful and open engagement). Communication with instructor, students and your software project clients is respectful and appropriate to the content and context.

Course expectations

Act as Professionals

This is an intense and fast pace course, so learn to manage your time – another hardest skill!

DO NOT BE LATE TO CLASS! It is Disrespectful to your instructor and work group!

Coming and Participating! to lectures and Project work

Teams

- Meet regularly
- Self-organize: Decide on team lead roles and reflect on how that matches personalities/ strengths
- Run as a real project

Expected weekly **workload is heavy**: 3 hrs lectures plus 8-10 hours outside class reading/project work (**reconsider taking this course if you have a busy term**)

Late comers (missed the first class) should familiarize themselves with all this information on Project Deliverables and Expectations. Get in touch with your project Team ASAP.

What previous students said…

Advice for future classes

Our advice to future students taking this course

- Plan, plan, plan!!
- Development a roadmap that outlines major milestone and task deadlines to keep everyone on the same page.
- Just because it's a project course, doesn't mean it's easy.
- Do not underestimate the course load.
- Ask lots of questions to avoid confusion

What previous students said…

Looking back from the end

- What could we have done differently?
  - Meet with clients more frequently
  - More in depth elicitation questions
  - Same people conducting interviews
- What did we struggle with?
  - Consistency throughout entirety of RSD 1.0
  - Ambiguity in RSD 1.0
- What did we do well?
  - Finalizing RSD 2.0
  - Team communication
- What did we learn?
  - Consistency is hard
  - Unambiguity is hard

Reminder: FIRST (THIS) Week!

Get to know your project team (same for Client and Developer Roles)

1. **Project Teams** information in Bright
2. Identify **Real-world Organization** and Schedule/conduct Interviews
3. (re-) Familiarize yourself with **Github** (see GH repo)
4. Please send Zane your Github Id
5. **Quiz 1**: Thu class
6. Required Readings by Thu class:

   Chapter 1 (and Chapter 7)