

SENG321: Requirements Engineering

Dr. Daniela Damian and Zane Li

thesegalgroup.org

danielad@uvic.ca

Requirements Engineering

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

[Brooks S., 1995: No silver bullet]

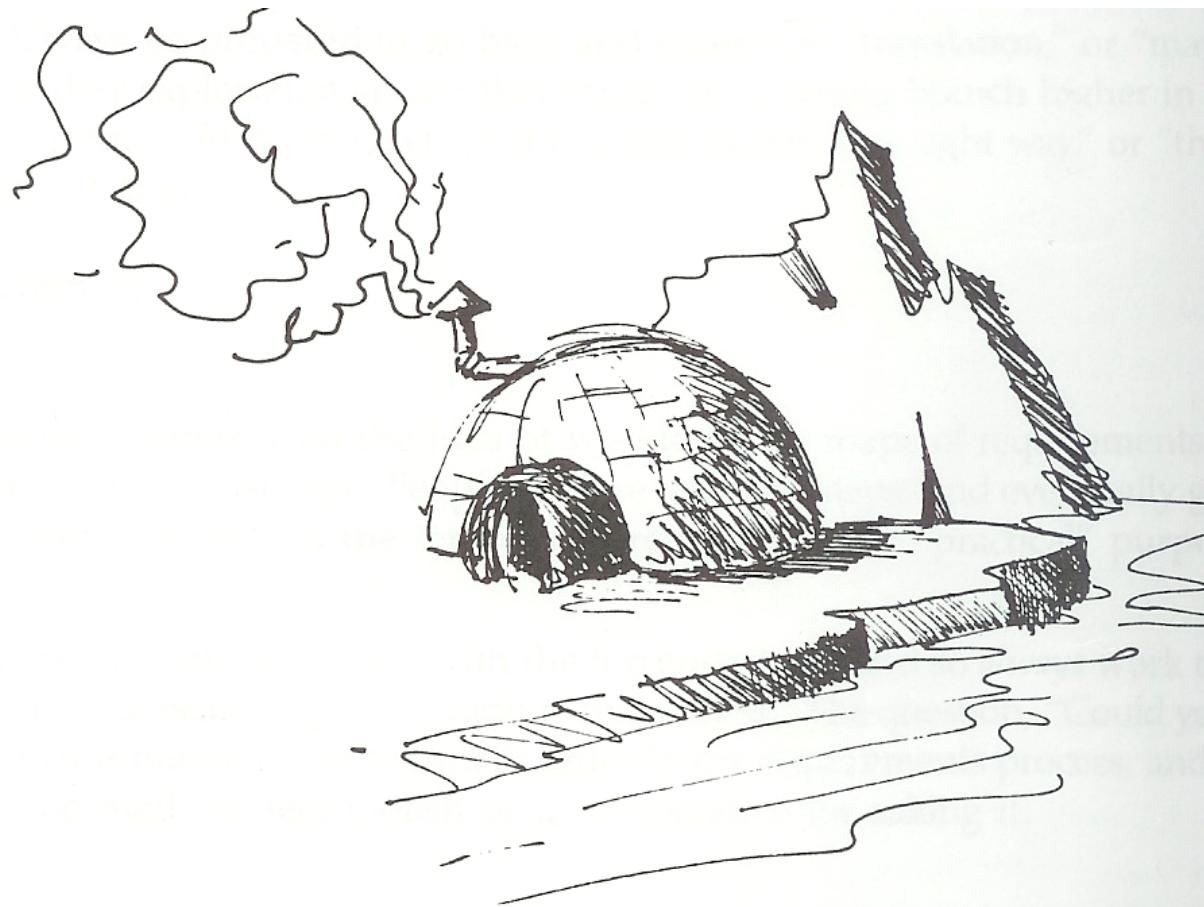
Leaving with ambiguity

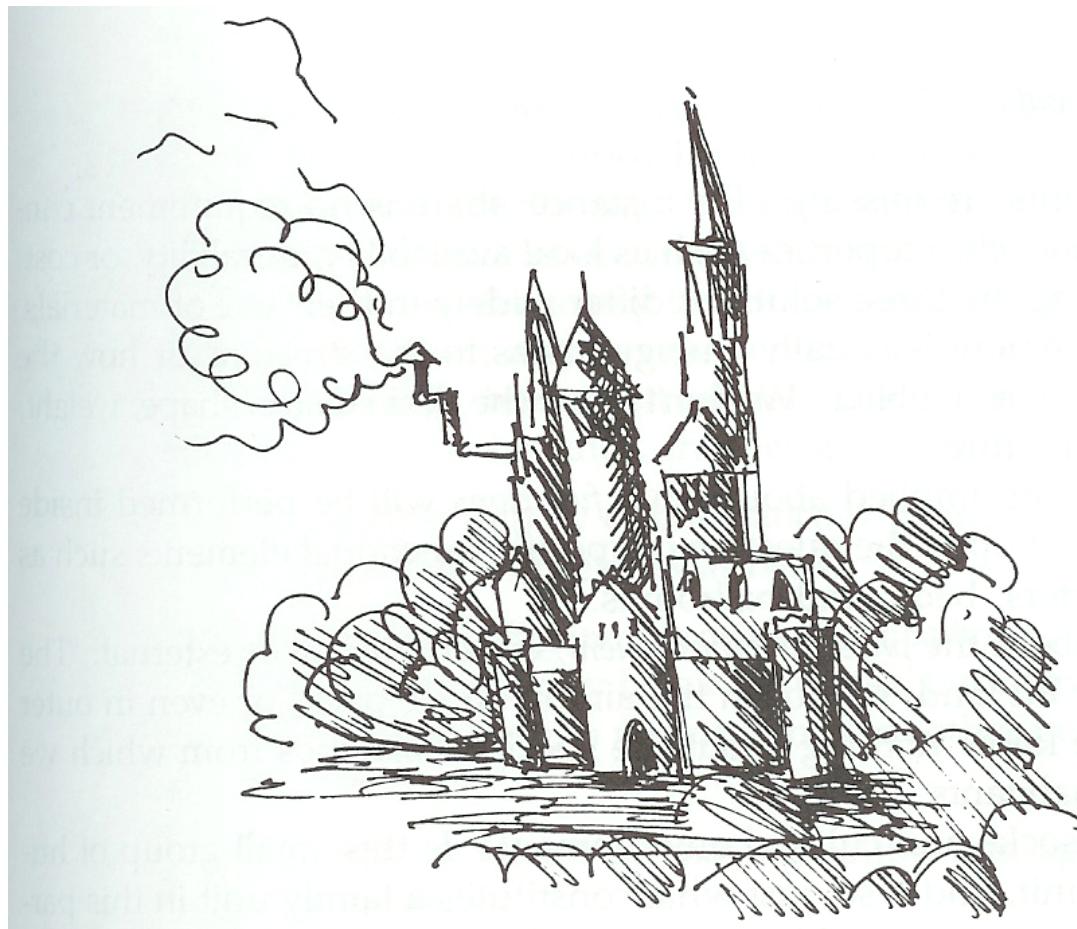
Requirement:

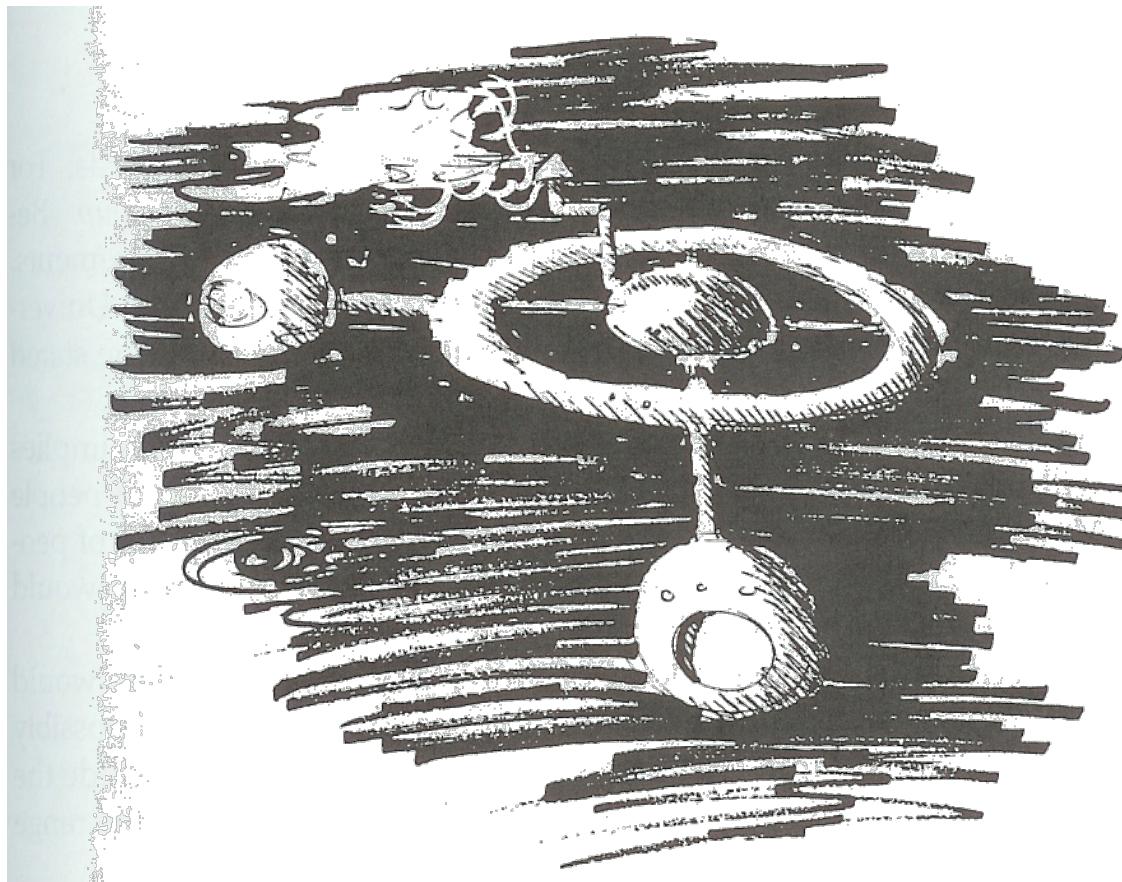
"create a means for protecting a small group of human beings from the hostile elements of their environment"

Possible solutions...

[Gause and Weinberg, 1989: Exploring requirements:
quality before design]









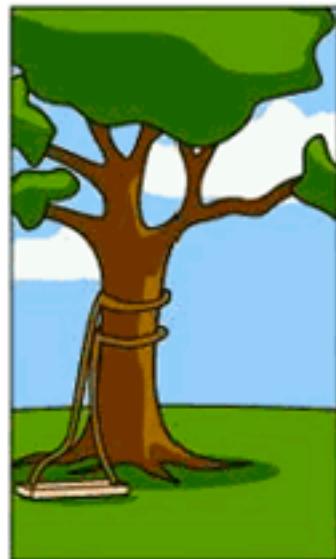
How the customer
explained it



How the project leader
understood it



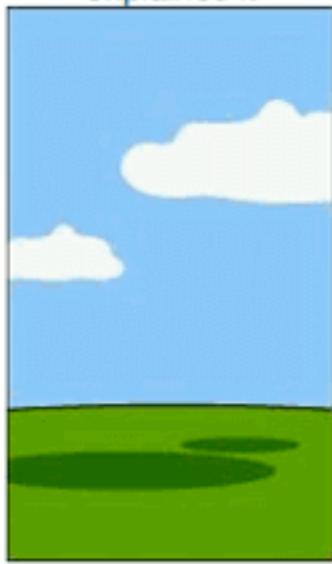
How the engineer
designed it



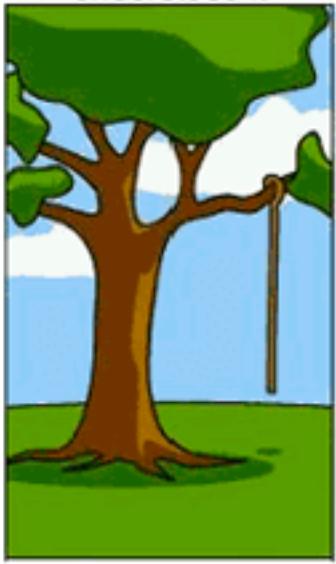
How the programmer
wrote it



How the sales
executive described it



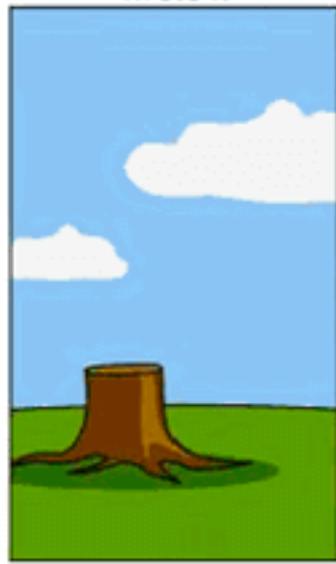
How the project was
documented



What operations
installed



How the customer
was billed



How the helpdesk
supported it



What the customer
really needed

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



Coordination and
Communication in SE



Software Ecosystems



Software Engineering
Education



Applications of SE

Welcome to SEGAL



SEGAL is a research facility in the Computer Science Department of University of Victoria, BC. We carry out research to improve the collaboration of geographically distributed software development teams. Global software development is increasingly becoming common practice in the software industry. The ability to develop software at remote sites in projects allows organizations to ignore the geographic distance and benefit from access to a qualified resource pool and reduced development costs. However, large software projects in such large

organizations or IT ecosystems involve complex interactions across organizational, functional as well as national, cultural and socio-economic boundaries, making their study important but difficult. In our [research projects](#) we employ a synergy of empirical methods, data mining and social network analysis techniques to understand these complex interactions as well as develop methods, processes and tools to improve the effectiveness of communication and coordination in large, distributed software projects. Check out a 2014 report on our [Research Programme](#) and the list of our [recent publications](#) for results of our work.

[How to Reach Us](#)



Recent Research News

16

MAY

MSc and PhD positions in
SEGAL!

Our lab is seeking applications for MSc and Doctoral research positions in the area of software ecosystems.

The research pertains to... [Read](#)

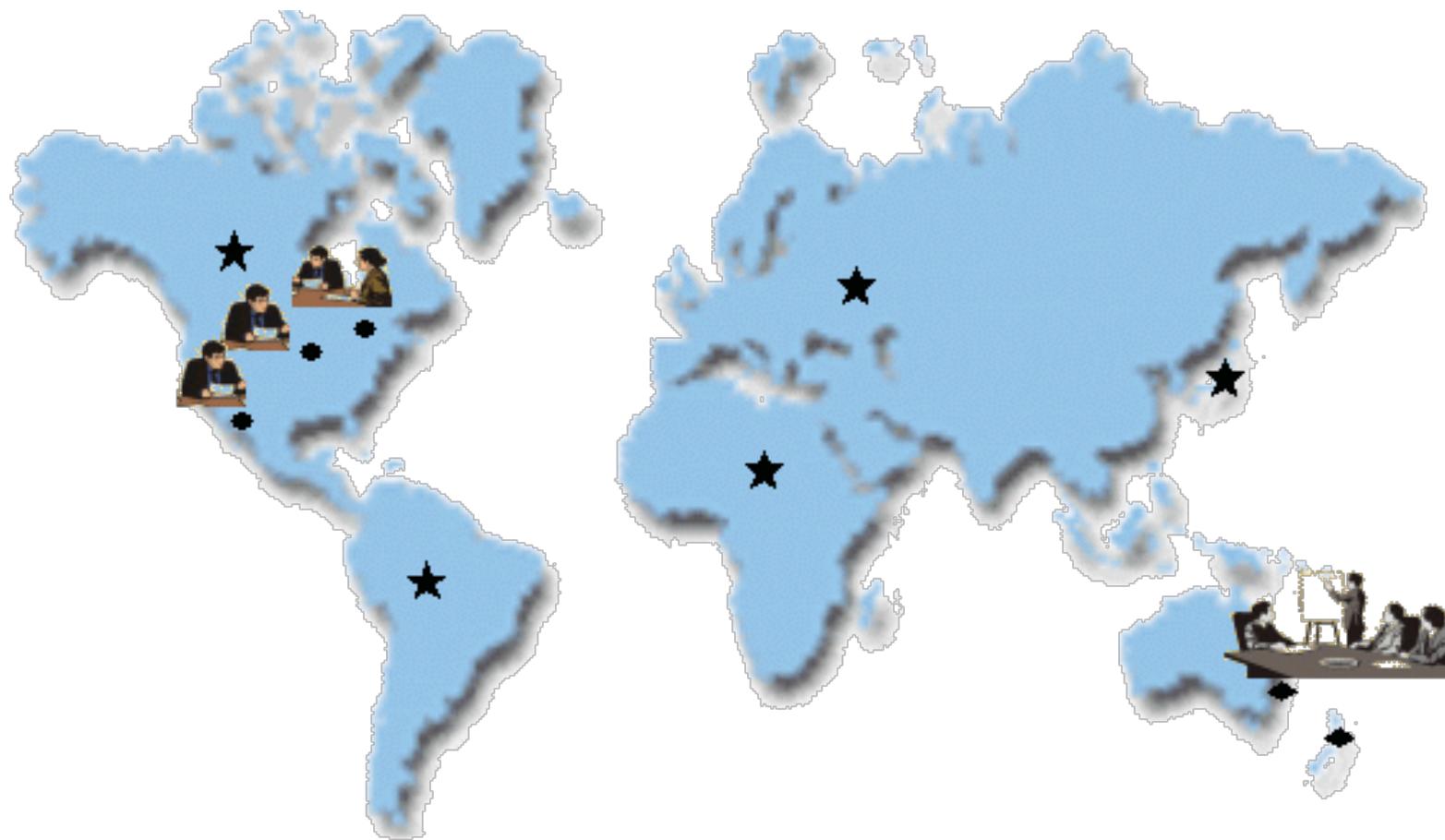
[More →](#)

13

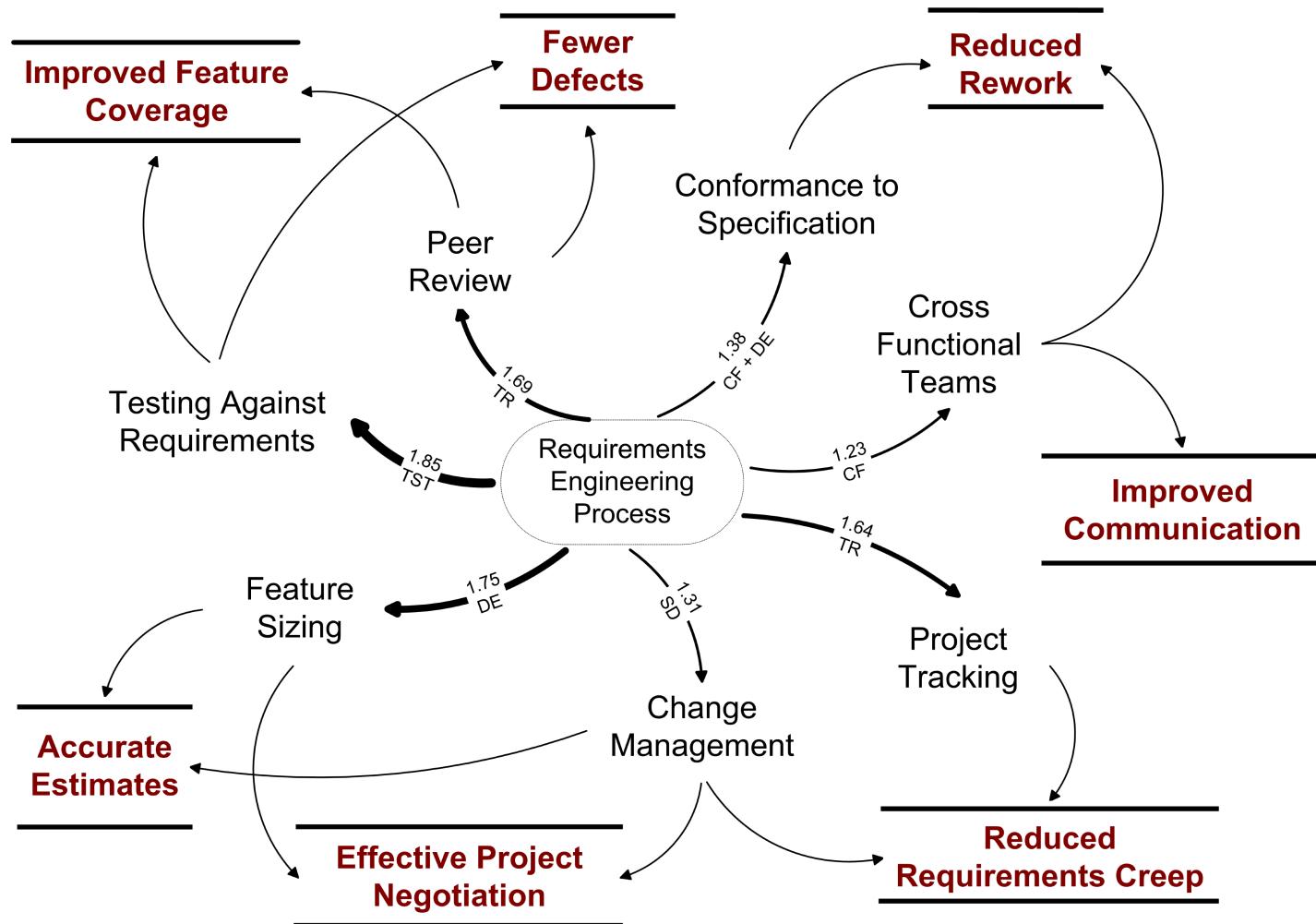
APR

Supporting the Adaptation of
Contextual Requirements at
Runtime

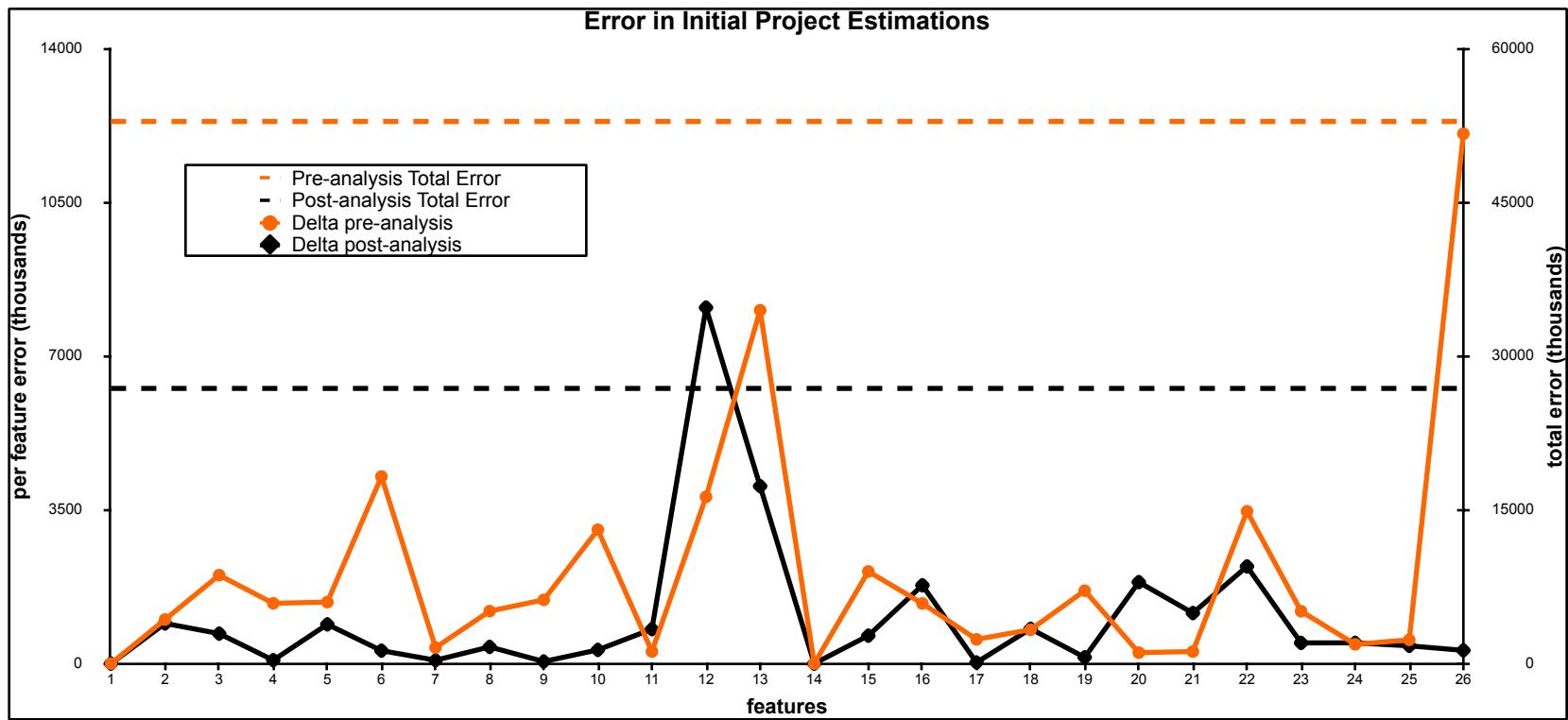
Typical client-developer relationship: trouble in software projects



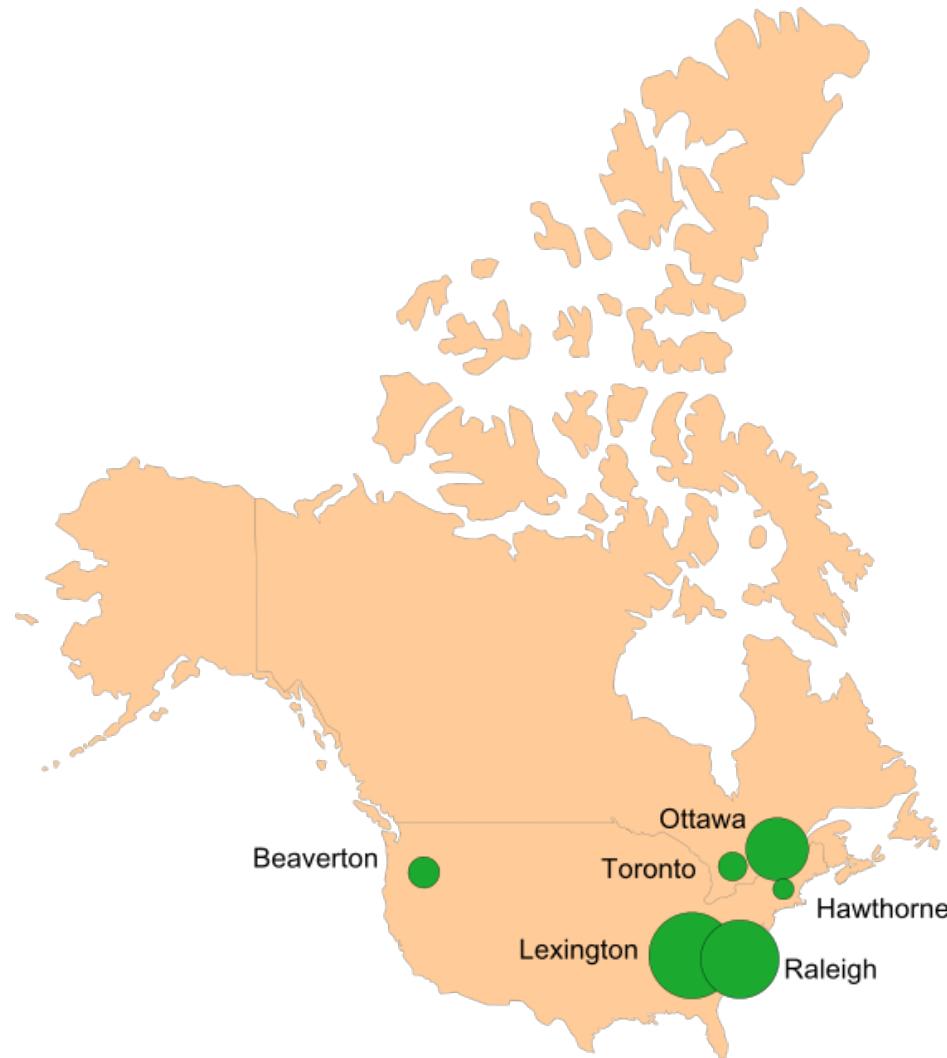
Rich interaction between the REP and other development processes contributed to gains in **productivity** (communication, rework), **quality** (defects) and **risk management** (estimations, feature coverage, negotiation, requirements creep).



Success stories: Significant improvements in estimation ability due to good RE processes



Case studies of developer communications in large projects



IBM Jazz Development Team

Legend:
Send 771 Requirements
Team Size 10 20 30 40 Engineering



Story 51890



Summary: * Test cases for different sync root types



Done



Overview

Done Criteria

Links

Approvals

History

Details

Description

Discussion (25 comments)

[Collapse All](#) | [Expand All](#)

1. Maneesh Mehra Apr 21, 2008 12:53 PM

Extracted from work item 49030.

2. Maneesh Mehra Apr 21, 2008 12:55 PM

Geoff/Cunxia: [Can you please look at the use cases and update](#) the ones which have questions next to them ? I was not sure what the intended outcome was for those cases.

3. cunxia sun Apr 23, 2008 11:05 AM

Note: in case 2: folder as sync root

14. Remove the sync folder on CC side and verify the folder is also removed from Jazz. [\(Will the sync root also be removed ?\)](#)

Actual behavior:

Remove the sync folder on CC side, re-sync; folder still exist in Jazz. sync will be removed.

4. cunxia sun Apr 23, 2008 12:09 PM

similar with step 15:

15. Remove the sync folder on Jazz side and verify the folder is removed from CC. [\(Will the sync root also be removed ?\)](#)

Remove the sync folder on jazz side, re-sync, folder is not removed in CC.

5. Geoffrey Clemm Apr 24, 2008 1:56 AM

It is the parent of a file that knows whether a file has been renamed, so if the parent is not a sync root, then neither the deletion of the file or the renaming of a file will be replayed in the other repository. This story should be updated to indicate this. (Possibly split into two stories, one where a parent (or some ancestor) of the file is also a sync root, and another where it is not).

Look at user stories
Can you please look at the use case and update...

Analyze discussion
(Will the sync root also be removed?)

Story 51890

Summary: * Test cases for different sync root types

Done

Overview Done Criteria Links Approvals History

Details

Description

Discussion (25 comments)

Collapse All | Expand All

1. Maneesh Mehra Apr 21, 2008 12:53 PM

Extracted from work item 49030.

2. Maneesh Mehra Apr 21, 2008 12:55 PM

Geoff/Cunxia: Can you please look at the use cases a
what the intended outcome was for those cases.

3. cunxia sun Apr 23, 2008 11:05 AM

Note: in case 2: folder as sync root

14. Remove the sync folder on CC side and verify the

Actual behavior:

Remove the sync folder on CC side, re-sync; folder st

4. cunxia sun Apr 23, 2008 12:09 PM

similar with step 15:

15. Remove the sync folder on Jazz side and verify the

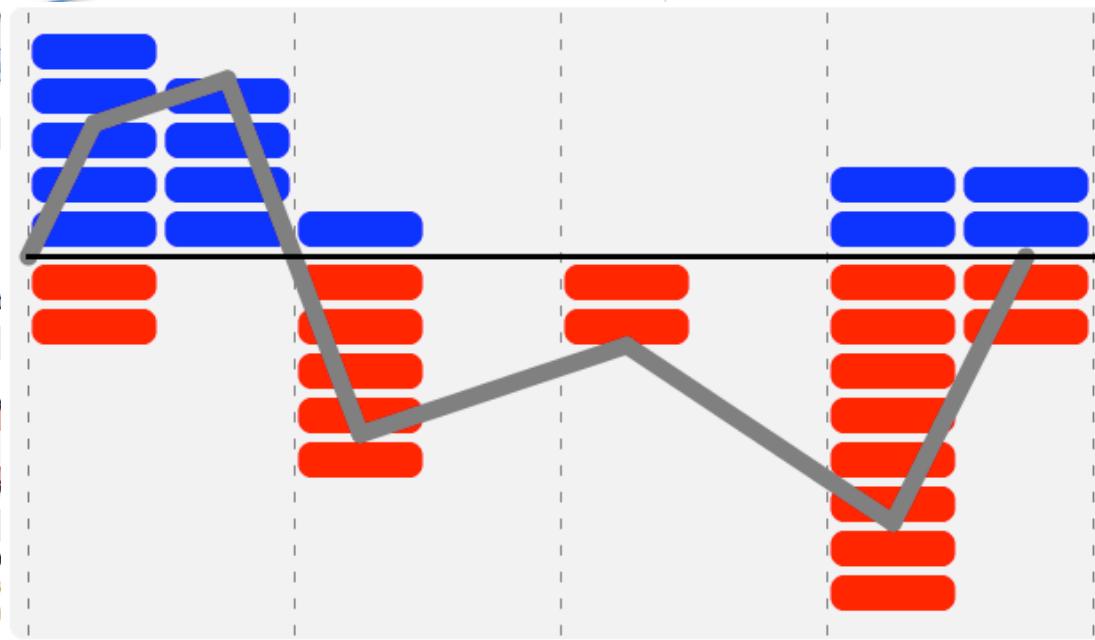
Remove the sync folder on jazz side, re-sync, folder is

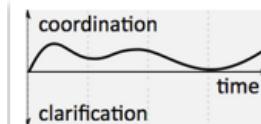
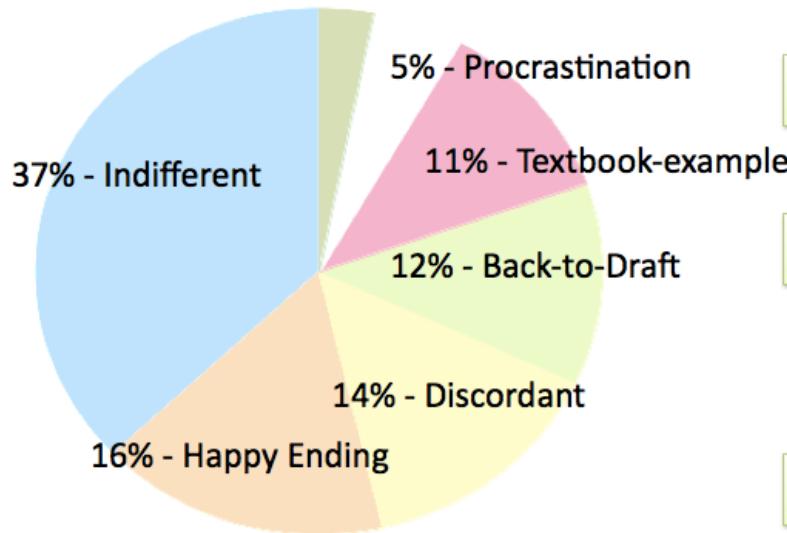
5. Geoffrey Clemm Apr 24, 2008 1:56 AM

It is the parent of a file that knows whether a file has been
deletion of the file or the renaming of a file will be repl
this. (Possibly split into two stories, one where a pare
is not).

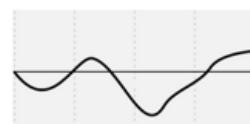
Can you please look at the use cases and update...

Look at user
stories

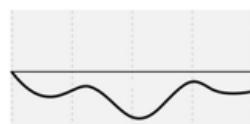




(a) Indifferent



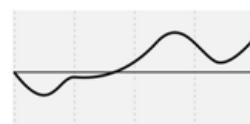
(b) Happy-ending



(c) Discordant



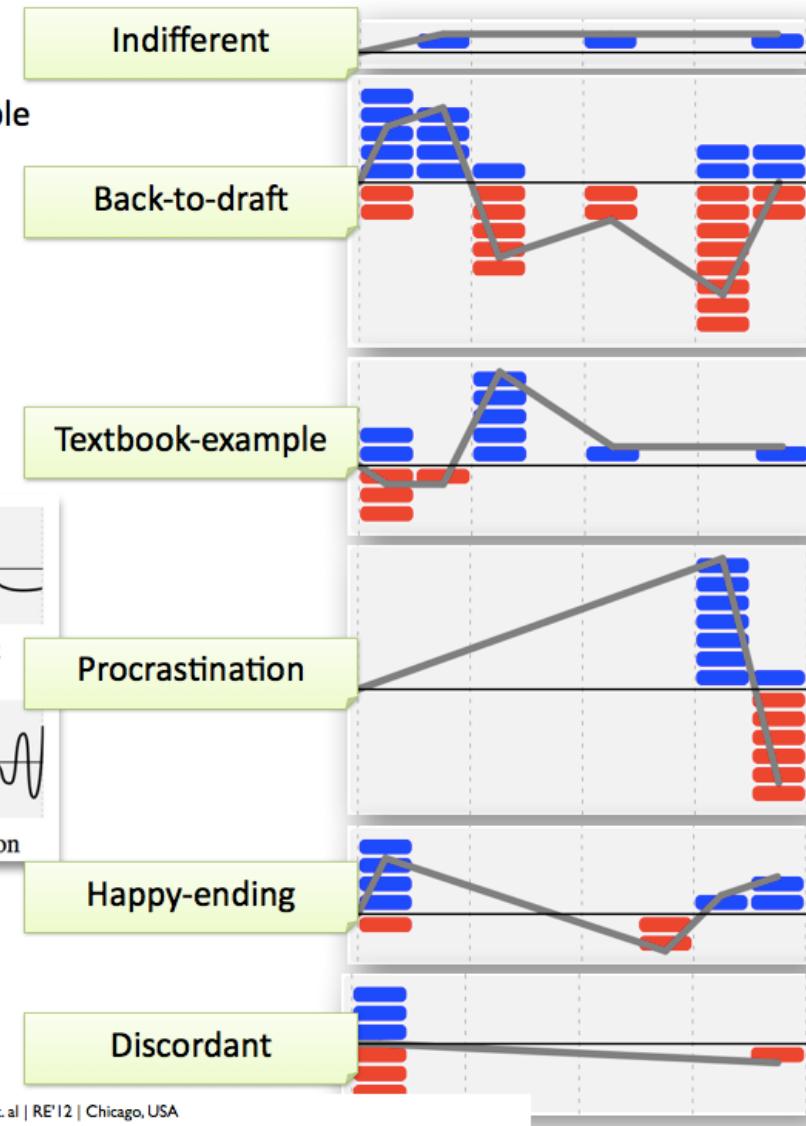
(d) Back-to-draft



(e) Textbook-example



(f) Procrastination



Requirements Engineering course

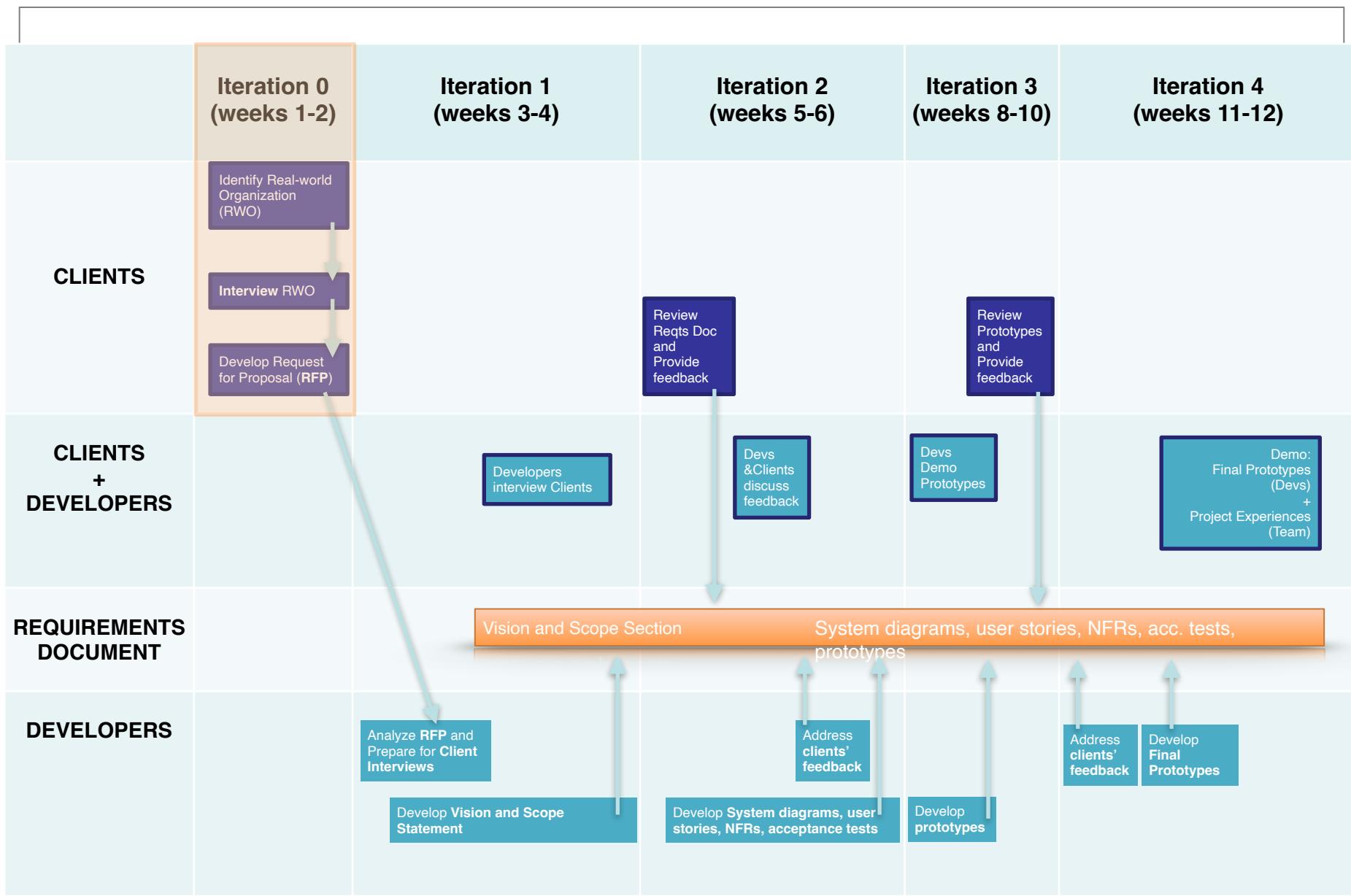
Requirements Engineering: beyond (software) requirements, throughout development, and pervasive in our lives

Improvement in RE processes significant to industry

Training in RE:

Critical industry need – how's AI going to change the landscape?

Offers competitive advantage



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%	

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 performance 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 in 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. Show initiative, commitment, autonomy.

Communication lines:

- Bright/GH
- your team mates
- your TA
- your instructor

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!!
- Develop 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 do well?
 - Finalizing RSD 2.0
 - Team communication
- What did we struggle with?
 - Consistency throughout entirety of RSD 1.0
 - Ambiguity in RSD 1.0
- 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)