CHALMERS |  UNIVERSITY OF GOTHENBURG

# DIT045/DAT355
# Requirements and User Experience

## (Start at 10:15)

## Course Introduction

Jennifer Horkoff
jenho@chalmers.se

# Outline

- Part 1: Content Introduction
  - Motivating Examples
  - What is RE?
  - Why is RE important?
  - Learning Objectives
  - What is UX?
  - Why is UX important?
  - Learning Objectives
  - Relation to SEM
- Part 2: Practicalities
  - Online lectures
  - Schedule
  - Assignments
  - Exams
  - Grading Scheme
  - Group work
  - Literature
  - TAs
  - Canvas
  - Communication Policies
  - Course Eval
  - Course Reps

# What is Requirements Engineering?

- Requirements
  - As implied, things that people or organizations require
  - (but some requirements can also be optional)
- Engineering
  - Being systematic and following reliable processes
- Requirements Engineering (RE)
  - Gather peoples needs (requirements) in a systematic and reliable way
- But…

# What is Requirements Engineering?

- But…
  - But how do you know what people need?
  - Do people know what they need? Who are the people?
  - Do they ask for the things that will really solve their problem?
  - What if they forget something that turns out to be really important?
  - What if one person asks for something that conflicts with what another person asks for?
  - What if they ask for things that they can't afford? Can't be finished in time?
  - What if they ask for things that will be really slow? Not easily usable?
  - What if they ask for things that break regulations? Are illegal? Unsafe? Risky?
  - …

Welcome to RE :)

# Examples?

- When you design and code, how do you know what to code?
- How do you know you are designing and coding the right thing?
  - In university?
  - In a job?
    - What if you are a team leader?
    - Manager?
    - CTO?

# Requirements Engineering: Examples in the Small

# RE Example 1: Network Security

- Example: Secure Network
- Requirement R: “The network shall only be accessible by authorized personnel”
- Domain Properties D:
  - Authorized personnel have passwords
  - Passwords are never shared with non-authorized personnel
- Specification S:
  - Access to the network shall only be granted after the user types an authorized password

(Easterbrook & Campbell)

- Is the network secure?

# RE Example 1: Network Security


We forgot something important…

Passwords shall never be recorded in such a way that they are visible by others

# Example 2: Ask Bob

- A consultant works with a team to build a system
- They carefully talk to potential users, developers and work hard to create something that the team really likes
- They are ready to make a beta release, and then…
- Developer: “Oh, we should show this to Bob”
- Consultant: “Who is Bob?”
- Developer: “Oh, you know, Bob, the VP (Head of operations, some high up position).” …
- They show the new software to Bob. Bob does not like it, not what he has in mind at all.
- The project is cancelled, or significant changes must be made

We forgot someone important…

# Requirements Engineering: Examples in the Large

# RE Example 3: Pheonix


While thousands of public servants await proper payment for their government jobs, IBM has already made more than $140 million and counting on the Phoenix payroll system it was hired to design and implement.

http://www.cbc.ca/news/canada/ottawa/phoenix-payroll-problems-ibm-1.3770947

# RE Example 3: Phoenix (cont.)

"One of my favourite quotes is good IT is expensive and bad IT is even more expensive," said Alex Beraskow, a management consultant with 30 years experience, both implementing and reviewing huge, multi-million dollar IT projects for the Canadian government.

The company was tasked with creating a new PeopleSoft-based payroll system for the government's more than 100 departments and agencies.

Beraskow said there will be many lessons learned out of Phoenix.

"A large part of the take away is making sure that the government, the users, the buyers know exactly what they want, that the procurement processes work and that the process is competitive at all times so that industry can build up their capacity as well to deliver on these projects."

http://www.cbc.ca/news/canada/ottawa/phoenix-payroll-problems-ibm-1.3770947

# RE Example 4: Denver Airport

- "The Denver International Airport tried to build a very sophisticated version of such a system (Baggage Handling System) several years ago. The system used PCs, thousands of remote controlled carts, and a 21-mile-long track. Carts moved along the track carrying luggage from check-in counters to sorting areas and then straight to the flights waiting at airport gates. After spending $230 milling (USD) over 10 years the project was cancelled. Much of the failure can be attributed to requirements engineering mistakes."
- Issues:
  - Poor performance
  - Poor reliability
  - Poor understanding of complexity, novelty

- De Neufville, Richard. "The baggage system at Denver: prospects and lessons." *Journal of Air Transport Management* 1.4 (1994): 229-236.
- http://www.denverpost.com/2014/12/31/united-express-has-major-baggage-issues-at-denver-airport/

# And… More examples ☺

- https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects
- Examples range from 1980's to 2013

# More examples?

- Can you think of examples of system failures due to poor understanding of requirements/user needs/scope?

# The “Software Crisis”

- “Software crisis is a term used in the early days (1972!) of computing science for the difficulty of writing useful and efficient computer programs in the required time.”
- “The crisis manifested itself in several ways:
  - Projects running over-budget
  - Projects running over-time
  - Software was very **inefficient**
  - Software was of **low quality**
  - **Software often did not meet requirements**
  - Projects were unmanageable and code difficult to **maintain**
  - Software was never delivered

**RE focuses here**

- Various processes and methodologies have been developed over the last few decades to improve software quality management .... However software projects that are large, complicated, **poorly specified**, and **involve unfamiliar aspects**, are still vulnerable to large, unanticipated problems.”

https://en.wikipedia.org/wiki/Software_crisis

# System Failures

- Solution 1 (SE Methods):
  - Build and deploy system incrementally
  - Heavy user involvement
  - Prototype
  - (a more Agile method)
- Solution 2 (More RE):
  - Spent more time upfront understanding the problem, users, environment, existing systems…
  - Capture knowledge in a structured and understandable way
  - (a more traditional Requirements Engineering method)
- Both can be good or bad depending on the type of system, size of system, size of team, nature of domain…
- Best solution can be in between

# So What is RE?

- RE is a set of systematic techniques, models, structures to help you think about the problem you will solve with technology
- RE techniques can be very simple or very complex
- Make sure you understand what you are trying to accomplish BEFORE you spend a lot of time coding/designing/creating an architecture
- Why?
  - To avoid wasting time coding something that won't be successful
  - To avoid having to make many changes
  - To make the final product better
  - To anticipate the effects of your product/software
  - To avoid being sued

# (Some) RE Questions

- What is the problem we are trying to solve?
- Is there a new business case? Innovation? Market?
- What does the system have to do to solve the problem?
- How well does it have to do this?
- Who will use the system?
- Who pays for it?
- Do different users have different problems? Which ones will we solve? Which problems are out of scope?
- What other systems do we interact with?
- What laws are relevant?
- What assumptions do we make? What are we dependent on? What are the risks?

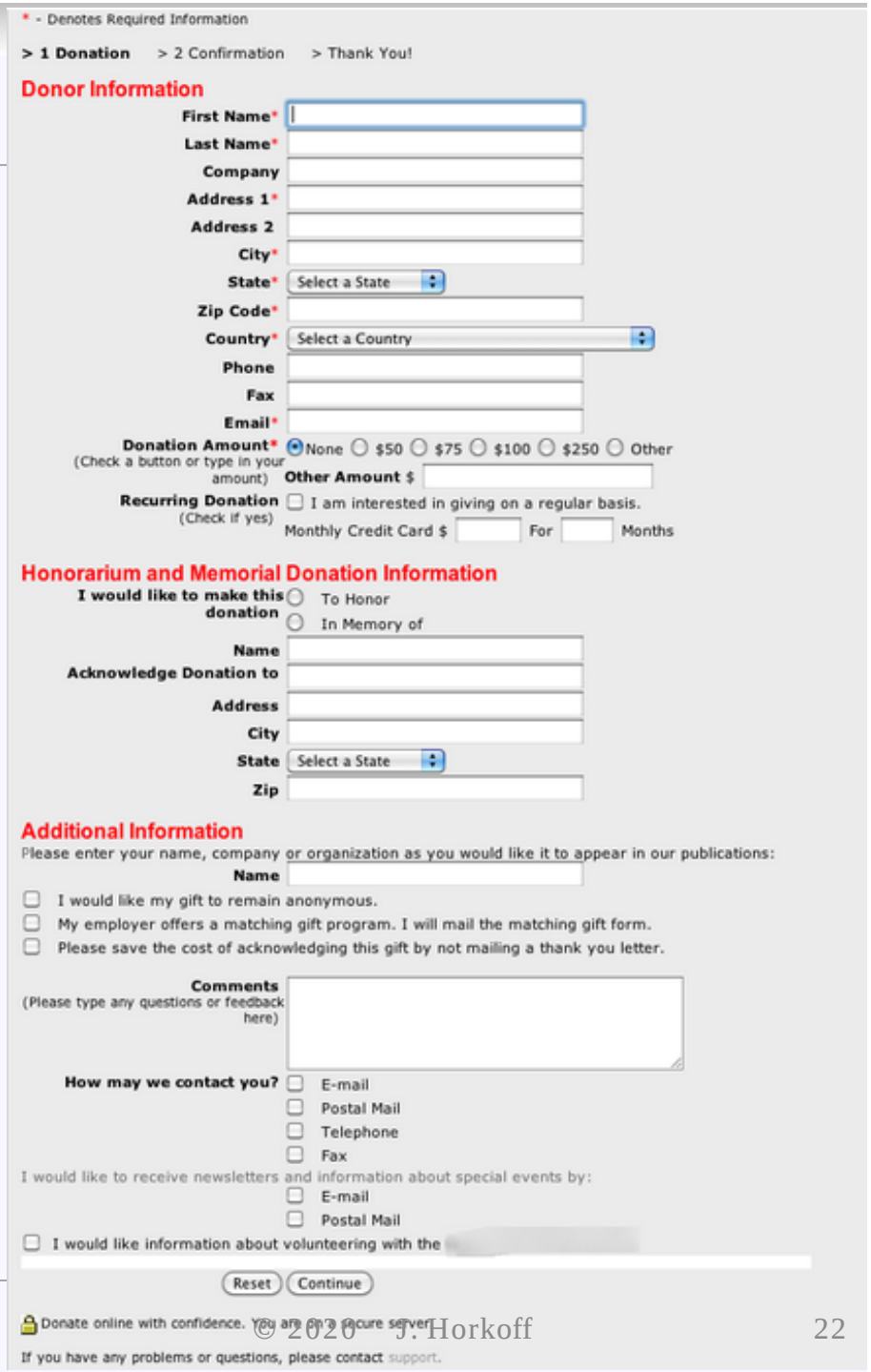
# Learning Objectives (RE subset)

- *Knowledge and understanding*
  - describe the process of requirements elicitation, evaluation and prioritization,
  - documentation, validation and development of software requirements,
  - state techniques to acquire and model user demands,
- *Skills and abilities*
  - identify and specify requirements by means of, for instance, scenario-based techniques or goal-oriented techniques,
  - apply techniques to identify personas, scenarios and user stories,
- *Judgement and approach*
  - choose and motivate appropriate methods for involving users in the design process.

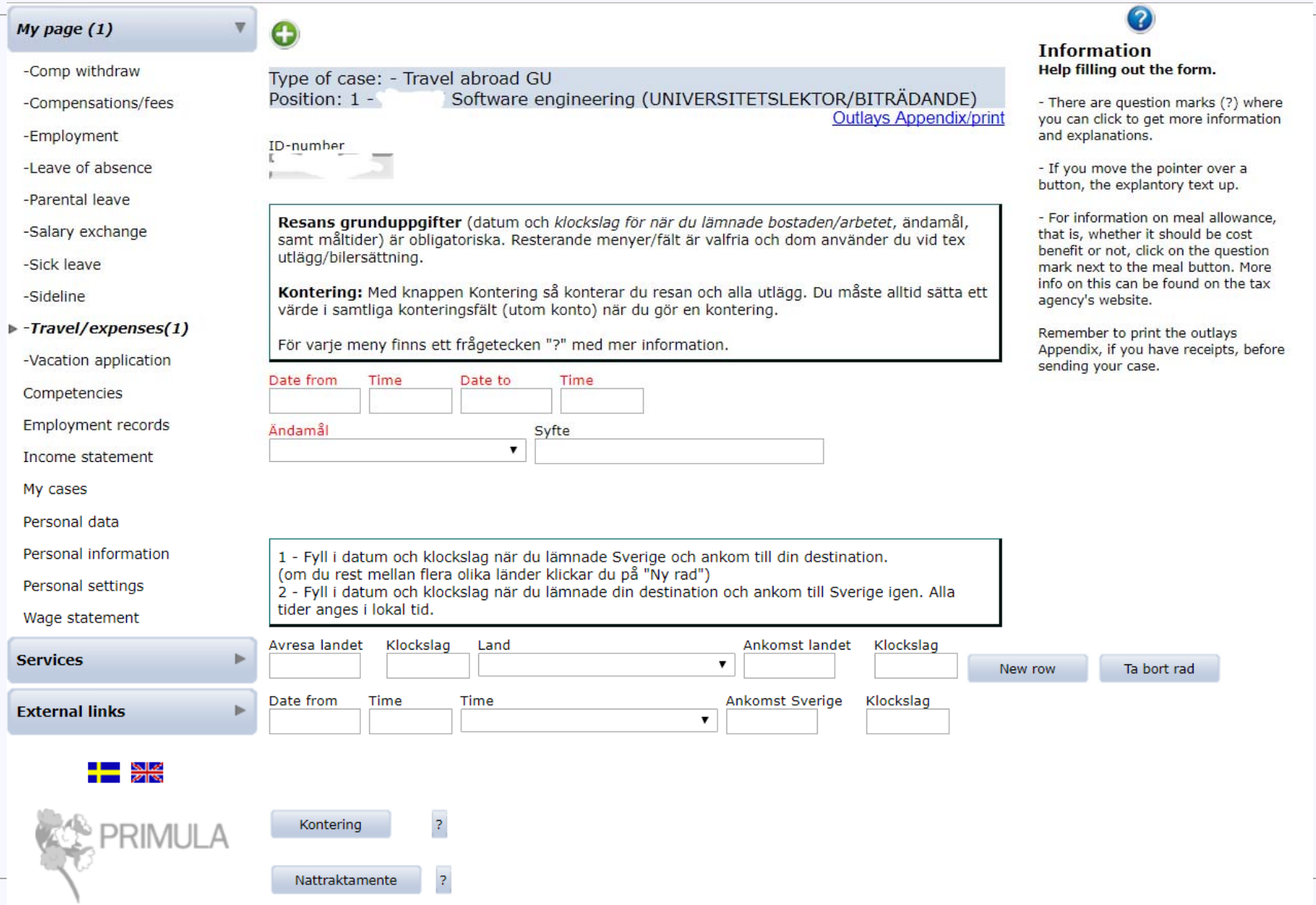
# User Experience:  Examples

# UX Example 1

https://thenextweb.com/dd/2015/09/29/6-examples-of-awful-ux-design/



# UX Example 2 ☹


My page (1)
-Comp withdraw
-Compensations/fees
-Employment
-Leave of absence
-Parental leave
-Salary exchange
-Sick leave
-Sideline
-Travel/expenses(1)
-Vacation application
Competencies
Employment records
Income statement
My cases
Personal data
Personal information
Personal settings
Wage statement
Services
External links
PRIMULA
Type of case: - Travel abroad GU
Position: 1 - Software engineering (UNIVERSITETSLEKTOR/BITRÄDANDE)
Outlays Appendix/print
ID-number
Resans grunduppgifter (datum och klockslag för när du lämnade bostaden/arbetet, ändamål, samt måltider) är obligatoriska. Resterande menyer/fält är valfria och dom använder du vid tex utlägg/bilersättning.
Kontering: Med knappen Kontering så konterar du resan och alla utlägg. Du måste alltid sätta ett värde i samtliga konteringsfält (utom konto) när du gör en kontering.
För varje meny finns ett frågetecken "?" med mer information.
Date from
Time
Date to
Time
Ändamål
Syfte
1 - Fyll i datum och klockslag när du lämnade Sverige och ankom till din destination.
(om du rest mellan flera olika länder klickar du på "Ny rad")
2 - Fyll i datum och klockslag när du lämnade din destination och ankom till Sverige igen. Alla tider anges i lokal tid.
Avresa landet
Klockslag
Land
Ankomst landet
Klockslag
New row
Ta bort rad
Date from
Time
Time
Ankomst Sverige
Klockslag
Kontering
?
Nattraktamente
?
Information
Help filling out the form.
- There are question marks (?) where you can click to get more information and explanations.
- If you move the pointer over a button, the explantory text up.
- For information on meal allowance, that is, whether it should be cost benefit or not, click on the question mark next to the meal button. More info on this can be found on the tax agency's website.
Remember to print the outlays Appendix, if you have receipts, before sending your case.

# UX Example 3

- Your bad day…
  - Wake up to full sunlight, clock says 3:43 am… you have 10 minutes to get to school!
  - Turn on the coffee maker… no coffee ☹
  - Drive to school… car needs gas! Gas station pump takes credit cards, but won't take yours. Must wait in line at the cashier, takes forever!
  - Driving detour due to accident…
  - Late for school! And no coffee..

- What does this have to do with UX? It's just bad luck? (Garrett)

# UX Example 5 (cont.)

- What does this have to do with UX? It's just bad luck?
  - Accident: the driver took his eyes off the road to turn the radio down, it was impossible to identify the volume button from touch alone
  - Register: the line moved slowly because the cash register was complex and confusing. The clerk would make a mistake and have to start all over again
  - Pump: You turned the card the wrong way to swipe it, but nothing on the pump indicated this, and you didn't notice
  - Coffee: you didn't push the power button all the way! No lights to tell you whether or not it has been turned on.
  - Clock: your cat stepped on the clock in the middle of the night and reset the time, a slightly different button configuration would have made the alarm cat proof!

(Garrett)

# More examples?

- Can you think of examples of poor UX design?

# What is User Experience?

- "Know they users, for they are not you!"
- Step 1: figure out what users are trying to accomplish
  - E.g., no one wants to fill out a form for the sake of filling a form
- Step 2: design, make familiar by using patterns

(Tidwell)

- User experience: the experience the product creates for the people who use it in the real world
- "UX is not about the inner workings of a product or service. User experience is about how it works on the outside, where a person comes into contact with it. When someone asks what it's like to use a product or service, they're asking about the user experience. Is it hard to do simple things? Is it easy to figure out? How does it feel to interact with the product?"

(Garrett)

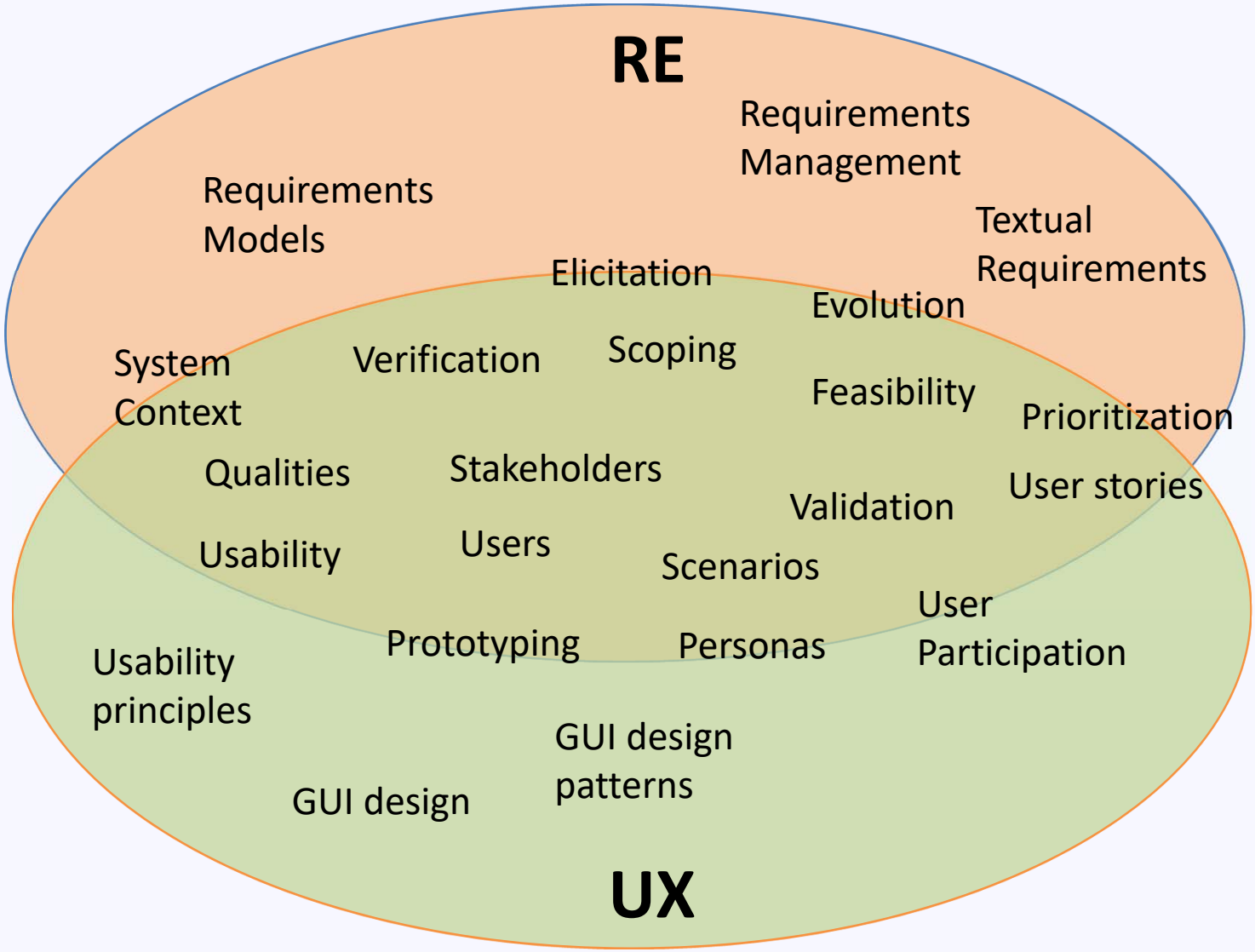
# Why is UX Important?

- Efficiency, Time
- User satisfaction
- Sales, attracting and keeping customers
- Safety
- Sanity

# Learning Objectives (UX subset)

- *Knowledge and understanding*
  - explain key techniques to account for usability in software products,
- *Skills and abilities*
  - apply techniques to identify personas, scenarios and user stories,
  - design and implement graphical user interfaces according to usability principles,
- *Judgement and approach*
  - choose an appropriate technique to evaluate the usability of a software product,
  - choose and motivate appropriate methods for involving users in the design process.

# How do UX and RE relate?



# RE + UX as part of SEM Bachelor

- We divide SE subjects into courses for easier teaching and learning
- But almost everything is connected and crosscutting
- Elements of RE + UX are almost everywhere in your degree
- (If you are a CS or other student, you can make similar mappings)

| Year 1 | | | |
|---|---|---|---|
| **Term 1**: Team Programming | | **Term 2**: Systems Development | |
| **Study period 1** | **Study period 2** | **Study period 3** | **Study period 4** |
| Object-Oriented Programming 7.5 credits<br>Someone has given you requirements | Requirements and User Experience 7.5 credits<br>You are here. Hi. | Data Management 7.5 credits<br>Requirements and UX design determine needed data | Software Analysis and Design 7.5 credits<br>How to translate requirements and UX design to working code |
| Mathematical Foundations for Software Engineering 7.5 credits<br>Some RE + UX methods (e.g., formal analysis, statistical analysis) need this | Mini Project: Team Programming 7.5 credits<br>Will do some RE and UX here in parallel | Data Structures and Algorithms 7.5 credits<br>Requirements will help select these, especially quality requirements | Mini Project: Systems Development 7.5 credits<br>All projects: should start with some RE, continuous, UX may happen |

| Year 2 | | | |
|---|---|---|---|
| **Term 3**: Distributed Systems Development | | **Term 4**: Cyber Physical Systems and Systems of Systems | |
| **Study period 1** | **Study period 2** | **Study period 3** | **Study period 4** |
| Fundamentals of Software Architecture, 7.5 credits<br>Requirements and architecture tightly linked (twin peaks) | Software Development Methodologies 7.5 credits<br>RE is part of all methods (even if not named), UX usually design | Development of Embedded and Real-Time Systems 7.5 credits<br>Requirements as input, some UX | Project: Cyber Physical Systems and Systems of Systems 15 credits<br>All projects: should start with some RE, continuous, UX may happen |
| Mobile and Web Development 7.5 credits<br>Should start with RE, lots of UX used | Mini Project: Distributed Systems Development 7.5 credits<br>All projects: should start with some RE, continuous, UX may happen | Software Quality and Testing 7.5 credits<br>Software quality could be part of requirements, software tested against requirements | |

| Year 3 | | | | |
|---|---|---|---|---|
| **Term 5**: Global and Enterprise Software Development | | **Term 6**: Software Engineering Research and Practice | | |
| **Study period 1** | **Study period 2** | **Study period 3** | **Study period 4** | |
| Optional course<br>Learn about creativity and innovation in RE | Optional course<br>RE relates strongly to change, requirements change | Research Methods in Software Engineering 7.5 credits | Bachelor Thesis in Software Engineering and Management 15 credits**<br>Can pick RE or UX topic | |
| Optional course<br>GSD involves distributed RE and UX | Optional course<br>RE and UX at enterprise level | | Startups and Industrial Software Product Management, 7.5 credits*<br>RE and UX in startups (not always called this) | |

- Project: Software Innovation, 15 credits
- Change Management in Software Development Organizations, 7.5 credits
- Global Software Development, 7.5 credits
- Project: Enterprise Software Development, 15 credits

# Summary

- Many example of system failures, in the small and in the large
- Failures can be avoided/mitigated by:
  - Incremental development, improved SE methods and/or
  - Attention to requirements engineering
- Many examples of poor UX
- RE and UX have quite a lot of overlap
- RE and UX are used in/relate to many things

# Questions?

# Part 2: Course Practicalites

# Outline

- Part 1: Content Introduction
  - Motivating Examples
  - What is RE?
  - Why is RE important?
  - Learning Objectives
  - What is UX?
  - Why is UX important?
  - Learning Objectives
  - Relation to SEM
- Part 2: Practicalities
  - Online lectures
  - Schedule
  - Assignments
  - Exams
  - Grading Scheme
  - Group work
  - Literature
  - TAs
  - Canvas
  - Communication Policies
  - Course Eval
  - Course Reps

# Online Lectures

- All course elements online
- Lectures, Exercises, and Office hours in Zoom
- Zoom lectures will be recorded and put on Canvas
  - Do not record the lectures yourself, you do not have permission
  - Extra example videos will also be posted on Canvas
- Announcements, materials, videos, readings (papers), assignments in Canvas
- Discussions and Questions in Canvas
- Exam online, open book, in Canvas
- All course activities in CET (Swedish) time zone

# Schedule

- Monday: Lecture: 10:00-11:45 Zoom
  - Office Hours: 9 to 10 Zoom Jennifer
- Wednesday: Lecture 10:00-11:45 Zoom
  - Supervision/Exercise: 13:15 to 15 Zoom Jennifer + TAs
  - Supervision/Exercise: 15 to 16 Zoom Extra time + TAs
- See Canvas for Course Schedule
- https://chalmers.instructure.com/courses/10966

# Assignments

- "**Assignments** *(Inlämningsuppgifter)*, 3 higher education credits Grading scale: Pass (G) and Fail (U)"
- Three assignments
- The assignment part of the course is worth 3 credits
- You will be given a grade on each assignment in %
- You must get at least 40% on each assignment
- Will average the grade for all three assignments, each weighted equally
- Passing grade is 60%!
- Final 3-credit assignment course is pass/fail only ☹

# Failing Assignments

- You can fail an assignment and pass the assignment part of a course if the overall assignment average is high enough
  - (e.g., 65%, 48%, 70% = 61%, pass, >60%)
  - (e.g., 65%, 38%, 90% = 64%, fail, one <40%)
  - (e.g., 55%, 70%, 50% = 58.3% fail, < 60%)
- If you fail an individual assignment (e.g., A1, A2, A3), you **CANNOT** hand it in to be regraded in the course
- If you fail the assignments overall, you can redo the assignments (all three of them) with a new case and hand them in again after the course is complete
- Assignment resubmissions dates in Canvas

# Exam

- **"Written exam**, 4.5 higher education credits Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)"
- Exam: Jan 13th, 2020 (morning)
  - Note: check student portal for updates and details on exam
- If you fail the exam, you can retake it
  - April, August - check dates with exam page
- See the following for any updates:
  - https://studentportal.gu.se/english/my-studies/cse/Examination
- Online exam via Canvas, so open book
- No communication with each other!
- To mitigate cheating, will vary exam questions on individual exams

# Grading Scheme

- For assignments:

| % Grade | GU Grading Scale |
|---|---|
| 0-59.9% | Fail (U) |
| 60-100% | Pass (G) |

- For exams:

| % Grade | GU Grading Scale |
|---|---|
| 0-49.9% | Fail (U) |
| 50-75.9% | Pass (G) |
| 75-100% | Pass with Distinction (VG) |

  - (if you are in Chalmers, exams grades like this:)

| % Grade | Chalmers Grading Scale |
|---|---|
| 0-49.9% | Fail |
| 50-64.9% | 3 |
| 65-79.9% | 4 |
| 80-100% | 5 |

# Group Work

- Assignments must be done in groups of 4-5
- Should be the same group throughout the course.
- You can make your own groups
- Remember: diverse teams can often perform better
  - https://www.hcamag.com/hr-news/do-diverse-teams-perform-better-245514.aspx
  - https://hbr.org/2016/11/why-diverse-teams-are-smarter
- You cannot work in a group of less than 4, unless someone in the group drops out of the course later in the course.
  - All groups are graded the same way, regardless of size
- Can or cannot be related to the group in parallel course.
  - Maybe a good opportunity to work with new people?

# Group Work

- For each assignment, fill in Peer Review.
- Found on Canvas under quizzes.
- Due the same day as the assignment.
- EVERYONE fills this out individually.
- Fill this out in order to get an individual grade for each assignment.
- Should take ~5 minutes, if all went well, can write very little.
- I may ask to meet with you.
- You are graded as individuals (!) so I can change the grade of individuals on an assignment based on reported contribution.

# Group Work

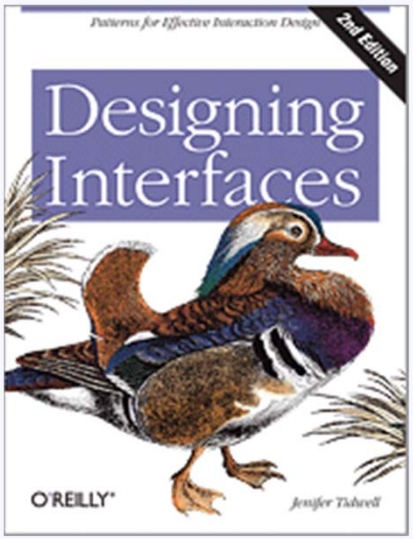
- If you don't have a group, we'll assign you groups on the Wed, supervision session, Nov. 4, 13:15.

- Fill out a form to report to me your groups.
  - Group name: pick an adjective and animal from the list provided
  - Group members: name, email
  - Form can be found on Canvas

- Do this by the end of Friday, Nov. 6th, A0 on Canvas
  - (This is not for marks, but I can't give you marks for the other assignments unless you are in a group of 4-5)

# Literature

- RE: Various papers/handouts – see Canvas for list




- UX:
- Course book: Jenny Tidwell's “Designing Interfaces: Patterns for Effective Interaction Design. 2nd Edition”

# TAs

- We have 12, will mark assignments/exams
- Will help with some exercises
  - Leith Hobson
  - Altug Altetmek
  - Maximilien Uddgren
  - Negin Hashmati
  - Fayona Cowperthwaite
  - Joakim Deak
  - Hartmut Fischer
  - Ali Karkhaneh
  - Hannah Maltkvist
  - Krasen Parvanov
  - Sandra Smoler Eisenberg
  - Victoria Vu

# Canvas

- Find Lecture Notes
- Find Assignment Descriptions
- Hand in assignments
- I will create groups for you on Canvas (don't make your own)
- I will use announcements quite a bit
- Use the Discussion board features
  - You ask me questions about the assignment, lectures, exams, I answer so all can read the answer.

# Communication Policies

- Contact via email, jenho@chalmer.se. Put course name or code in subject (DIT045/DAT355, RE&UX)
- Note: if you are asking a general question that concerns the lectures, assignments, exam, etc. If you send me an email with a general question, I will tell you to post it to the discussion forum
- Blackout policy: I make no guarantee to answer questions about assignments in the 24 hours before they are due.
- I do not guarantee to answer emails outside of office hours.
  - E.g., assignment is due Friday at 23:59. I stop answering questions Thursday at 5 pm.
  - Be prepared!

# Course Evaluation

- The university rules have use using the standard course evaluation on Canvas
  - Please fill this out at the end of the course

- During the last lecture you'll be provided with a 1-page evaluation form
- Please take some time to fill it out
- It will be collected by the course representatives and handed to Richard (program manager)
  - It is anonymous, I will not see the filled out versions

# Course Representatives

- Anyone is super welcome to talk to me in person or via email with issues about the course
  - Sometimes it's too late to fix something this year, but I can fix it next year
- You can also talk to the course representatives
- I need volunteers!
  - 2-3 from Chalmers
  - 2-3 from GU
  - Email jenho@chalmers.se I will pick the first 2-3
- We'll meet mid-way through the course (end of November)

# Questions?

# Lecture Sources

- IREB (International Requirements Engineering Board)
  - https://www.ireb.org/en/downloads/
- Requirements Engineering (CSC340) S. Easterbrook, J. Campbell
  - http://www.cs.toronto.edu/~sme/CSC340F/
- Requirements Engineering for Software and Systems, Second Edition, By Phillip A. Laplante Kilicay-Ergin, Nil, and Phillip A. Laplante. "An online graduate requirements engineering course." *IEEE Transactions on Education* 56.2 (2013): 208-216.
- Dick, Jeremy, Elizabeth Hull, and Ken Jackson. *Requirements engineering*. Springer, 2017.
- "What is Requirements Engineering?" the draft chapter 1 and "What are Requirements?" the draft chapter 2 of Fundamentals of Requirements Engineering (FoRE), S. Easterbrook, 2004.
- Bray, Ian K. *An introduction to requirements engineering*. Pearson Education, 2002.

# Lecture Sources


- Wilson, Tom. "Software failure: management failure. Amazing stories and cautionary tales: S. Flowers Wiley, Chichester, New York,(1996) 197 pp£ 19.99 ISBN 0 171-95113-7." *International Journal of Information Management* 17.5 (1997): 387.
- Macaulay, Linda A. *Requirements engineering*. Springer Science & Business Media, 2012.
- Pohl K (1993) The three dimensions of requirements engineering. CAiSE'93, Paris, France
- Jackson, Michael, and Pamela Zave. "Deriving specifications from requirements: an example." *Software Engineering, 1995. ICSE 1995. 17th International Conference on*. IEEE, 1995.

# Lecture Sources


- Nuseibeh, Bashar, and Steve Easterbrook. "Requirements engineering: a roadmap." *Proceedings of the Conference on the Future of Software Engineering*. ACM, 2000.
- Cheng, Betty HC, and Joanne M. Atlee. "Research directions in requirements engineering." *2007 Future of Software Engineering*. IEEE Computer Society, 2007.
- Garrett, Jesse James. *Elements of user experience, the: user-centered design for the web and beyond*. Pearson Education, 2010.