Software Engineering Project Workshop (SENG202)

Matthias Galster

Phase 1 – tasks (part 1)

July 15, 2020

UC
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Deliverables

- Project setup checklist
- Design document
- Reflections and logs
- Presentation

Deliverables

- Project setup checklist
- Design document
- Reflections and logs
- Presentation

Format

- Must include cover page (followed by a table of contents)
  - Project title and project phase
  - Team number and team members
  - Submission date

- Numbering
  - Number all sections (except exec summary, references, appendix)
  - All pages, tables and figures must be numbered
  - Tables and figures must be labelled, and referenced in the text

- Appendix must show who worked on which sections

Hint

Documents are “linear” pieces of text. This may not reflect how software engineers work. Software engineering is done in a highly incremental and iterative way. Your documents will describe your project, analysis, design, etc. at certain points in time. This means that you may need to work on several issues (or report sections) in parallel (top-down as well as bottom-up). In general, your documents must be written so that reviewers can understand them, i.e., assumptions, concepts, notations (syntax and semantics), etc. must be documented and explained.

Discuss issues, decisions and implications with us (and your team) **before** investing significant time resources

Hint

Create a design document readable by an outsider (imagine a new member joins the team and is given just the document). In previous years the reports for the first phase ranged from 24 pages to 32 pages. This included everything (cover page, ToC, appendix, large figures and tables, project plan, list of contributions, etc.). Note that the length of the document does not necessarily positively correlate with its quality. However, a document of 5-10 pages most likely won't capture enough details and writing 50 pages probably means that you are over-specifying things at this point. Remember that you will add and refine things later. You should focus more on the contents of the report and not worry about the length as that will come naturally. You receive suggestions for content to fill the report, so if you want a really great grade you should keep working away at that list and put as much good work into the report as you can. You will not be marked on length (as quality is more important), but the more high quality work you give us the better your mark will be.

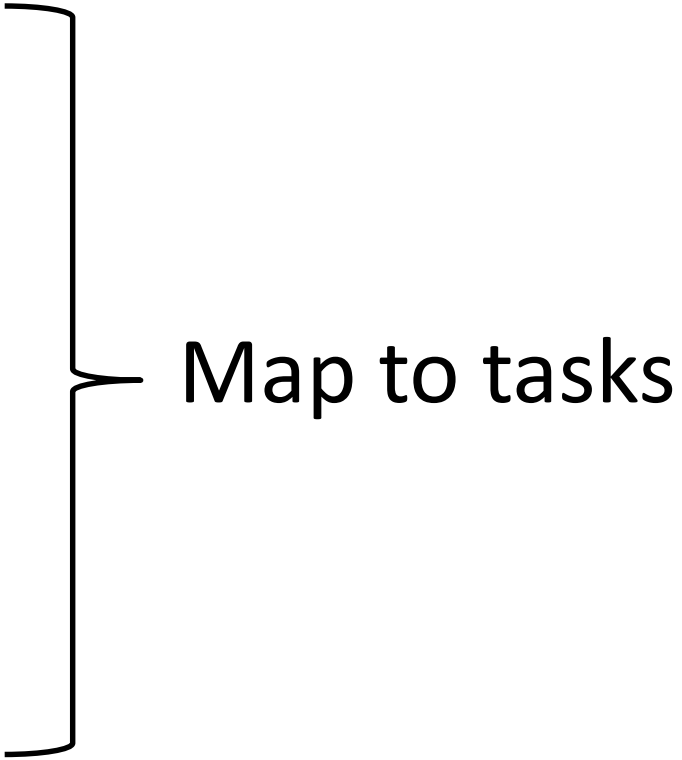
Content of design document

Executive summary

1. Business and system context
2. Stakeholders and requirements
3. Acceptance tests
4. GUI prototypes
5. Deployment model
6. Detailed UML class diagram
7. Risk assessment
8. Project plan

References

Appendix



Content of design document

Executive summary

1. Business and system context
2. Stakeholders and requirements
3. Acceptance tests
4. GUI prototypes
5. Deployment model
6. Detailed UML class diagram
7. Risk assessment
8. Project plan

References

Appendix

Executive summary

- Brief summary of document (0.5 page, ~1-3 paragraphs)
  - Brief overview of the main idea of document, background information, etc.
  - Typically written as one of the last sections

- Example content
  - What kind of system is described
  - What is target market
  - Who are main stakeholders (including some key concerns)
  - How product differentiates from other products
  - Key features
  - Most important quality requirements
  - Most important risks
  - Etc.

Content of design document

Executive summary

1. Business and system context
2. Stakeholders and requirements
3. Acceptance tests
4. GUI prototypes
5. Deployment model
6. Detailed UML class diagram
7. Risk assessment
8. Project plan

References

Appendix

Business and system context

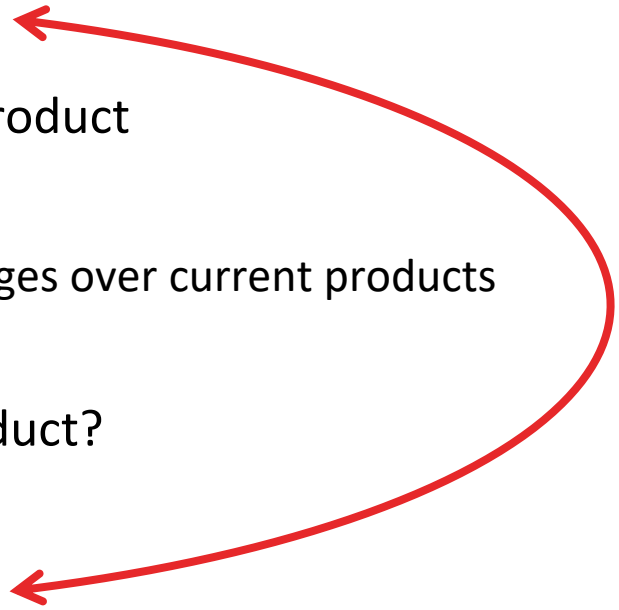
- Why think about business and system context?
  - Motivates system + clarifies (shared) product vision and rationale

1.1 Relevant business information

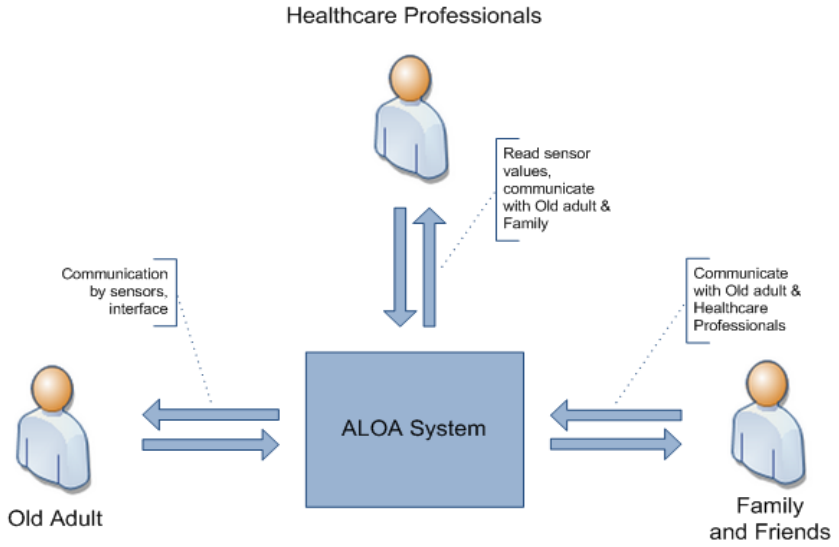
- High level description of services provided by product
- Business opportunities
  - Unique selling points + anticipated (dis-)advantages over current products
  - Target markets, potential customers
- Why is it worth developing (paying for) the product?

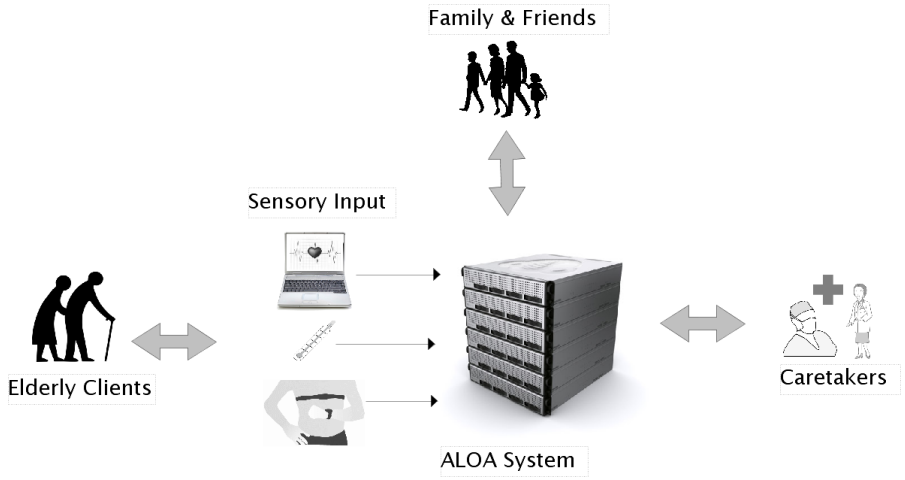
1.2 System context

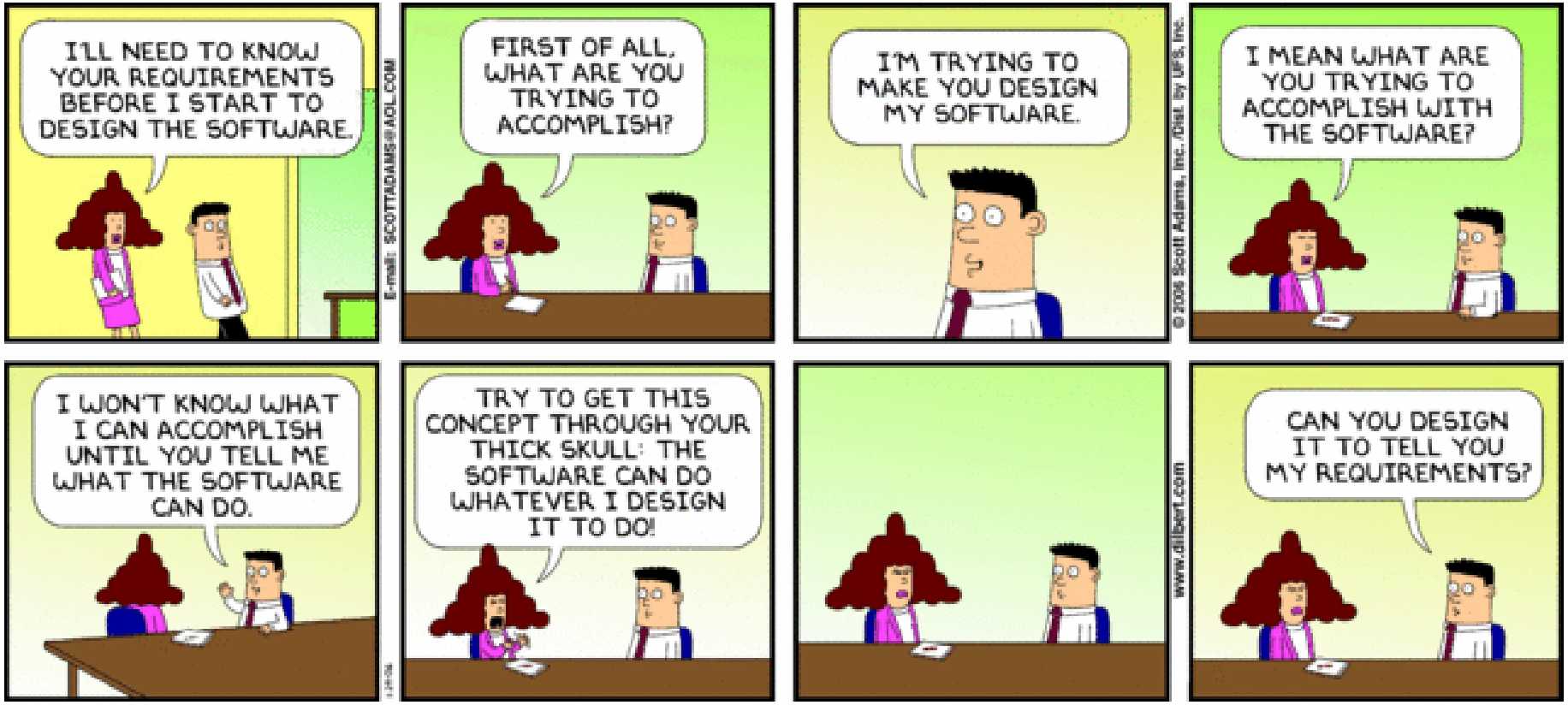
- Envisioned system in its context
- How will system be used, by whom, why, how?



System context – example






I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?
E-mail: SCOTTADAMS@AOL.COM
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
www.dilbert.com

Content of design document

Executive summary

1. Business and system context
2. Stakeholders and requirements
3. Acceptance tests
4. GUI prototypes
5. Deployment model
6. Detailed UML class diagram
7. Risk assessment
8. Project plan

References

Appendix

2. Stakeholders and requirements

2.1 Stakeholders

- Table with stakeholders

2.2 Requirements

- Use cases
  - Use case diagrams
  - Textual description of use cases
- Functional requirements
- Quality requirements

Hint

On a big project we have to start somewhere. That somewhere is figuring out some initial requirements. In this project we will not go into the details of a defined process for capturing and analysing requirements. Instead, we will provide an initial set of product features that can be used as a starting point to define requirements for the project. Carefully read the project description. We discuss the project description in the lab (note that developing project requirements with stakeholders is common practice in industry and it is surprisingly hard to do well).

Stakeholders and concerns: It is usually difficult to develop a system and to identify its requirements without knowing its environment, the context of the system under development and any relevant business information. Thus, you should first think about the domain of your system, including potential stakeholders and their concerns. In short, stakeholders are people, roles, organizations, etc. with “an interest in your system”. This includes end users but also other entities that may affect the software product as well as the software product development process.

Stakeholders

- Who will be using your product?
- Who will be affected or who will impact
  - Product
  - Development process
- People, roles, organizations, etc.
- Need to be prioritized

Table with stakeholders

ID	Stakeholder (+ description) and concerns	Priority

Stakeholders – examples

- Development team, course staff
- Professional athletes, casual exercisers (including their relatives)
- Runners, cyclists, swimmers
- Medical specialists
- Data analysts
- Personal trainers
- Sport teams
- Sporting goods/fitness equipment manufacturers
- Retailers
- Social media
- UC

Concerns – examples

- A successful product will produce good grades
- To provide adequate education
- Ability to track fitness progress
- Collect and analyze data to provide better health treatment
- Responsive and reliable application, intuitive interface
- Relevant analyses, informative data visualization
- Track client progress
- Consumer research, better marketing
- Increased sales of complementary fitness goods
- Increased user retention
- Reputation