McMaster University

What are (good) Requirements?

Dr. Sébastien Mosser
Lecture #02
CS/SE 3RA3 - Fall 2024

McSCert

1 What are “Requirements”?

2 Req. Quality & Validation

3 Let’s write some requirements!

Comments from industry

- "*We can train good people in our technology stack;* ***we can't as easily train them in sociotechnical skills like requirements engineering****, project management, presentations…*" — Chief Software Architect [Rolls-Royce]
- "*We hire people with great technical skills but where* ***they really struggle*** *is with the other things:* ***RE****, certification, critical path analysis…*" — Principal Engineer [IBM]
- "*A junior engineer will naturally evolve into a senior without difficulties. But when they reach this stage,* ***only a few will become architects****. Other options require a* ***deep understanding of our products and stakeholders****, and they're really not good at that.*" — Senior HR manager, [BigTech]
- "***Developers are easily replaceable by cheaper ones*** *if they only code and do not get involved in the project requirements & management*".— Tech Lead [BigTech]

Comments from Former Students

ENGINEERING
Computing
& Software

- "***I didn't like writing requirements specifications*** *in my degree, but now I'm working* ***I hate even more not having a requirements specification****.*"
- "*My co-op claims to do* ***DevOps*** *but in reality* ***it's just chaos****. There's no requirements spec.* ***No-one really knows what they're doing****. I do my best to* ***steer the mob in the right direction****.*"
- "*I'm working at [large bank] and the dev skills of the team are impressive. However,* ***without our project lead who actually understands RE****, we would have* ***crashed and burned a year ago****.*"
- "*My* ***technical skills are decent*** *and probably got me this job at [well known company]. But* ***my soft skills and RE skills got me the promotion*** *to tech lead.*"

Requirements Engineering

Is a necessary evil

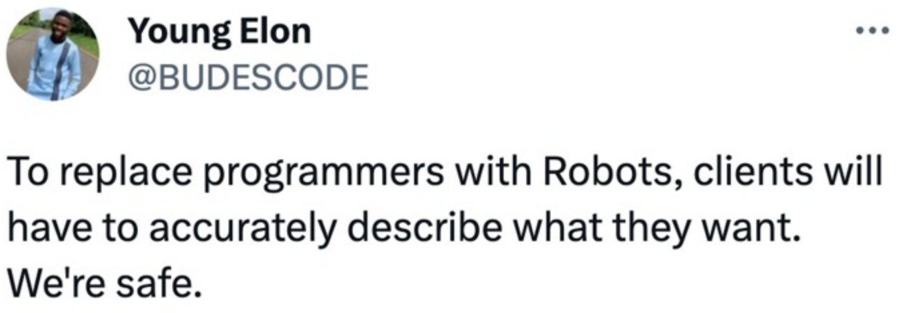
*For your projects, **and** for your career.*

What are “Requirements”?

Let’s define some important terms

1

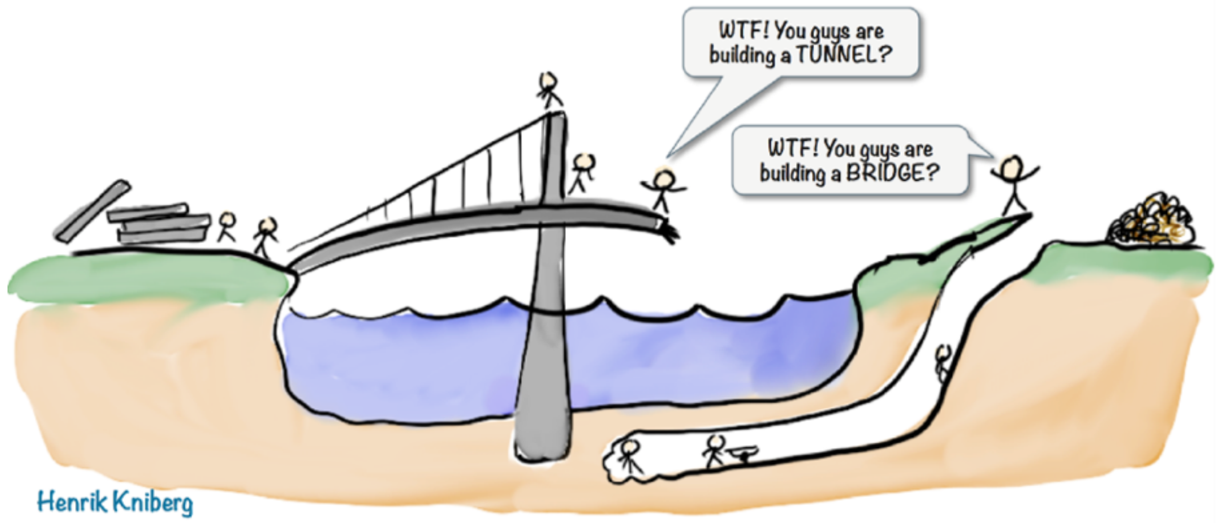
Functional & Non-functional Requirements


Young Elon
@BUDESCODE

To replace programmers with Robots, clients will have to accurately describe what they want.
We're safe.

Functional: WHAT does the system do?

Functional & Non-functional Requirements


WTF! You guys are building a TUNNEL?
WTF! You guys are building a BRIDGE?
Henrik Kniberg

Non-Functional: HOW does the system do it?

What is a **requirement?**

- **New Shorter Oxford English Dictionary**
  - *"**Something** called for or **demanded**, a **condition which must be complied with**."*
- **IEEE Standard 29148** (older version: IEEE 830)
  - *"A **condition** or capability **that must be met or possessed by a system** or system component to **satisfy a contract**, standard, specification, or other formally imposed document. The set of all requirements forms the **basis for subsequent development** of the system or system component."*
- **Handbook of Requirements and Business Analysis**:
  - *"A requirement is a **relevant statement** about a **project**, **environment**, **goal** or **system**"*

Categorizing **Requirements**

- **Goals**: *The desired results for the target organization*
  - ***Obstacle***: a property that needs to be overcome
- **Behaviour**: Property of the inspiration of the system
  - ***Functional***: outcome produced by the system
  - ***Non-functional***: property of how the system achieves the outcome
- **Constraints**: Property imposed by the environment
  - Business rule, Physical rule, Engineering decision
- **Other environmental elements**: Assumption, Effects, Invariants

What is **Requirements Engineering?**

- "Requirements engineering is the branch of software engineering concerned with the **real-world goals** for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to **precise specifications of software behaviour**, and to their **evolution over time** and **across software families**." — Pamela Zave (Bell Labs/AT&T)
- Notes:
  - **Real-world goals** focus on the '**why**' as well as the '**what**'
  - **Precision** paves the way for **analysis** and **validation**
  - **Requirements change** and are often re-used in later projects
  - But, Zave's focus is **limited to software engineering** and does not consider the wider context of **systems engineering**...

Another **definition**

- "A **requirements specification** should **provide individuals with everything they need to know** to satisfy the relevant stakeholders... **but nothing more**."
  — Parnas, modified by McDermid, modified by Vickers
- This definition is **not restricted to software** alone
- **Important**: **Requirements** is not **design**
  - *Design decisions belong to designers* (not necessarily ≠ people)
- **Example**: In your project, you will **identify some software components**
  - **Identifying** the components and their requirements ≠ **designing** the components ≠ **coding** the components (layered abstractions)

Yet Another Definition

- “*Requirements Engineering is the **task of developing requirements**. Developing includes not just **producing an initial version** of the requirements, but **updating** it regularly, and **managing** the (possibly complex) **set of requirements**.*” — Meyer
- **Note**:
  - This definition converges with the accepted definition of “**business analysis**.”
  - **Business analysis** is more focused on **business goals**
  - **Requirements Engineering** has a more **technical connotation**
- This definition emphasizes the “**set” dimension of requirements**

2

Reqs. Quality & Validation

Let’s write good requirements

First things first

- Meyer: "A requirement is a **relevant statement** about a **[…] system"**
- **Statement**:
  - A statement is a **human-readable expression** of a property
  - **Property**: A **boolean trait** of a project, environment, goal or system
- **Relevance**:
  - A **goal** is always relevant
  - Something that affects (or is affected by) stakeholders is relevant ***(project, system)***
  - Something that affects (or is affected by) the project is relevant ***(environment)***

Some Examples

- **Statement**:
  - "*All humans are mortals*". $\forall x : \text{Human} \mid \text{isMortal}(x)$
  - "*Tous les hommes sont mortels*" (**lost in translation**: Humans ⇝ hommes ≍ men)
- **Requirements**:
  - *The project shall produce a 1st release by October 31, 2023*
  - *All websites shall conform to PIPEDA*
  - *The Bridge Maintenance System shall limit bridge closure to no more than one night a month*
  - *After 5 failed login attempts, access shall be blocked for 30 minutes*

Remember the first lecture?

A good requirement states something that is necessary, verifiable, and attainable

- Use these **three pillars** as an immediate **go/no-go sieve** for requirements
  - Is the thing I am writing "**necessary**"?
  - Is the thing I am writing "**attainable**"?
    - Can you **achieve it** with a **reasonable amount of resources**?
  - Is the thing you are writing "**verifiable**"?
    - *Can you **prove to someone —not yourself!**— that you've **accomplished** the requirement?*

Quality Attributes for Requirements

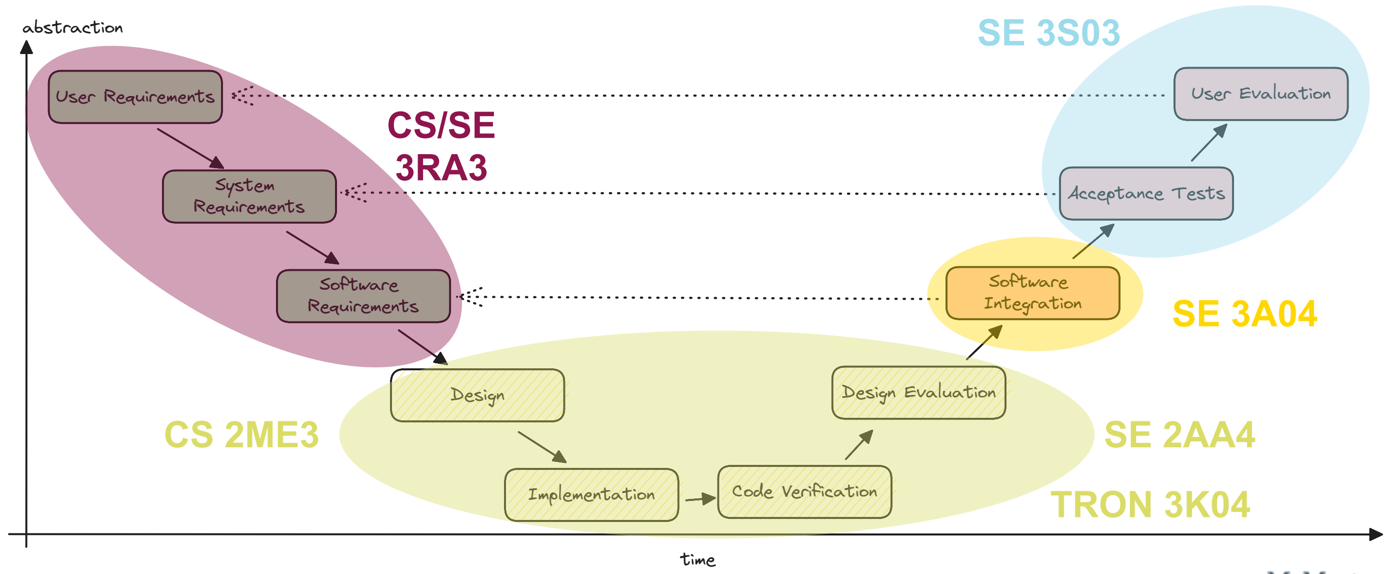
Attribute	Applies to
Correct	*Environment & System*
Justified	*Project & System*
Complete	*Everything*
Consistent	*Everything*
Unambiguous	*Everything*
Feasible	*Project & System*
Abstract	*System*

Attribute	Applies to
Traceable	*Everything*
Delimited	*Everything*
Readable	*Everything*
Modifiable	*Everything*
Verifiable	*Project & System*
Prioritized	*System*
Endorsed	*Everything*

Necessary, **Attainable**, **Verifiable**

BRIGHTER WORLD | **Sébastien Mosser**, *Computing and Software (CAS), Faculty of Engineering* *(Adapted from Handbook — Chapter 4)* *24 August 2024* | *18*

Where are we in the development cycle?



Pedagogical Template used in 3RA3

The four books of requirements	
Project (P) P.1 Roles and personnel P.2 Imposed technical choices P.3 Schedule and milestones P.4 Tasks and deliverables P.5 Required technology elements P.6 Risk and mitigation analysis P.7 Requirements process and report	**Goals (G)** G.1 Context and overall objective G.2 Current situation G.3 Expected benefits G.4 Functionality overview G.5 High-level usage scenarios G.6 Limitations and exclusions G.7 Stakeholders and requirements sources
Environment (E) E.1 Glossary E.2 Components E.3 Constraints E.4 Assumptions E.5 Effects E.6 Invariants	**System (S)** S.1 Components S.2 Functionality S.3 Interfaces S.4 Detailed usage scenarios S.5 Prioritization S.6 Verification and acceptance criteria

https://github.com/ace-lectures/cas-handbook-req-template

(Handbook — Chapter 3)

Let's write some Reqs.!

Fail early. Fail Fast. Recover.

3

The MacNav app!

- We've received a multi-million dollar investment to build the next GoogleMaps!
  - (It's actually a pretty stupid idea, we'll work on that soon with goals/pitch)
- What would be the requirements of MacNav?

Click on the link the instructor just sent on MS Teams and collaboratively provide some requirements for MacNav

Conclusions

TL;DR: Takeaway Messages

Takeaway Message

- This lecture could have been summarized into the following:
  - **Boring Definitions** and **CommonSense 101**
- A lot of Requirement Engineering is about **definitions** and **common sense**
  - We'll use ***tools***, ***methods*** and ***frameworks*** to ensure thoroughness
  - The **critical challenge** is to ensure **consistency**
- **No requirement document is perfect**
  - "***The best is the enemy of the good***" — Voltaire
  - You'll have to **make trade-off decisions**. All. The. Time. **#DealWithIt**

Next Lecture (Friday)

Is your first workshop

(No laptops. Please bring pens and paper)

McMaster University

ENGINEERING

Computing & Software

McSCert

Dr. Sébastien Mosser
Associate Professor

mossers@mcmaster.ca
https://mosser.github.io

ITB 131 (appointment only)