MI0A112T - Ingénierie du logiciel avancé

Tableau de bord / Cours / S2 - MI0A112T

Modern Requirements and Business Analysis

1 - Introduction

Jean-Michel Bruel

UE 1002 – Spécification des exigences

https://bit.ly/jmbruel

@SmartModelTeam

https://github.com/smart-researchteam

UNIVERSITÉ TOULOUSE Jean Jaurès

SM@RT

IRIT

Official reference:

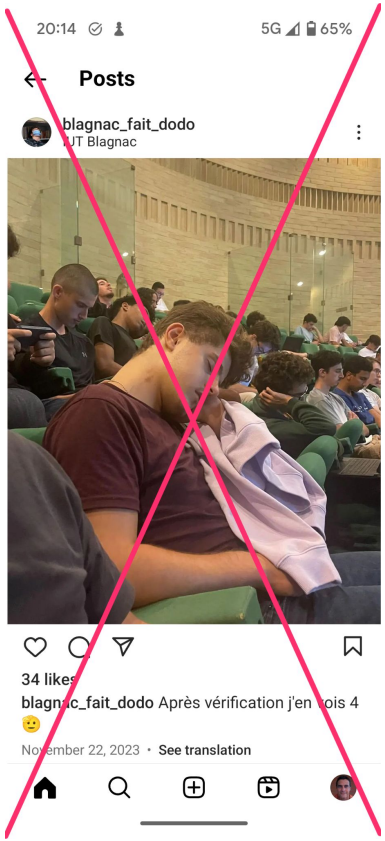
https://iris.univ-tlse2.fr/course/view.php?id=3645

How to cite:

Jean-Michel Bruel, “Cours Ingénierie des Exigences – Master ICE”. Toulouse, 2024.

Get my slides (pdf)
(not yet)


NO FOOD
OR DRINK


20:14
5G 65%
Posts
blagnac_fait_dodo
IUT Blagnac
34 likes
blagnac_fait_dodo Après vérification j'en vois 4
November 22, 2023 · See translation


C'est pas
du
Retard
...
{ C'EST DU SUSPENSE }

25’ of attention

- Today
- During practice
- Any time!

#P5	IS Nan Messe	IS/RE Thuy Nguyen	RE JMB 
Thèmes	Généralités sur l'IS Lien Software Engineering (V vs Agile) SysML	IS assistée par la modélisation & la simulation Stimulus	Ingénierie des exigences en lien avec l'IS Stimulus PEGS
22-26 avril			8h - Généralités - ISO XYZ - GORE (Kaos) - Justifications
29 avril - 3 mai	4 - Intro Système & IS - Intro Processus	4 - BASAALT - Iso	4+6 - Exigences formelles - Initiation à Stimulus (Cranbrook) - Tracabilité - Exigences et SysML ?
6-7 mai			8 - Projet PEGS
13-17	4+4 - Lien UML et SysML - Les diagrammes SysML avec Papyrus	4 - Stimulus sur un cas pratique	(4+2) - Projet PEGS
21-24		4 - Projet Stimulus	
Evaluation	50% d'IS Exam (contrôle dernière heure)	50% d'IS Projet	Projet 70% de RE Exam/QCM 30% de RE

Disclaimer (and assumptions)

Désolé pour l'utilisation de l'anglais 😅

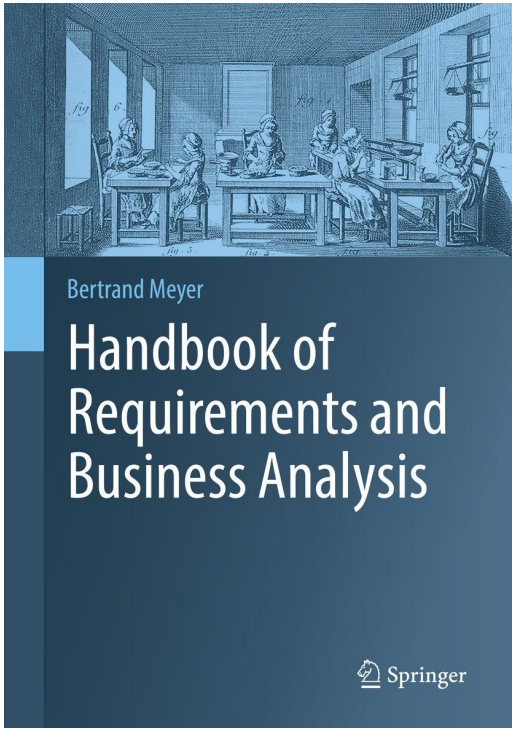
C'est la 1ère fois que je donne ce cours!!

=> vous allez être mis à contribution !

Why me?

- Professor at Toulouse University
  - Teaching modeling and DevOps
- Member of the CNRS-IRIT Laboratory
  - Model-Based Systems Engineering
- Airbus MBSE Chair of Toulouse
- Leader of the companion book (end of 2024)

https:/bit.ly/jmbruel

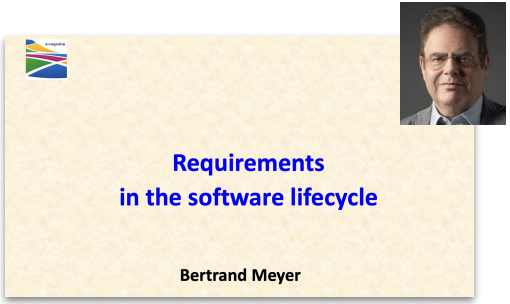


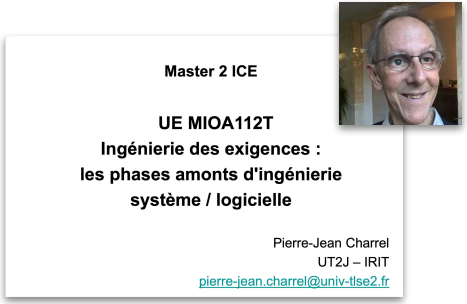
https://se.inf.ethz.ch/requirements/

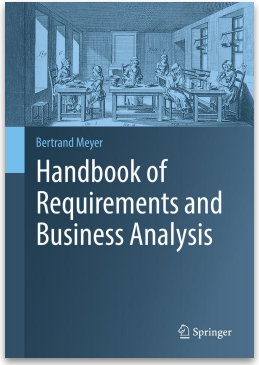
Joint effort...

- Innopolis University
  - Alexandr 
  - Manuel 
- Constructor Institute
  - Bertrand 
  - Li 
- IRIT/SM@RT team
  - Florian 
  - Sophie 
  - JMB 
  - Maria 
  - Pierre-Jean 
- CoCoVaD
  - Imen 
  - Thuy 
- McMaster University
  - Sébastien 

Sources for this course







https://bit.ly/jmb-teaching

https://requirements.university

Outline

Don't trust this!

- (22/04) 4h - Introduction to RE — Overview/ Teasing
- (24/04) 4h - PEGS approach / Reqs taxonomy
- (29/04) 4h - Practical aspects — Practice
- (02/05) 4h - Introduction to STIMULUS — Practice
- (02/05) 2h - Tooling / Practice — Practice
- (06/05) 2h - PEGS approach / Reqs taxonomy
- (07/05) 6h - Project #phase1
- (15/05) 4h - Project #phase2 — Active Participation
- (17/05) 3h - Project presentation / Feedback

33H

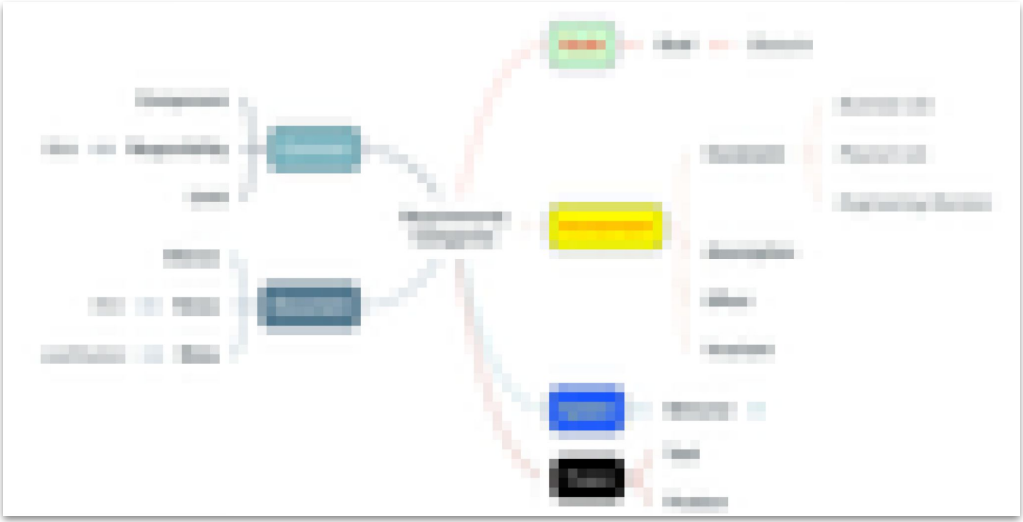

practice

What about **you**?

What about **requirements**?

Practice 1: Collaborative experience

1. Open collaborative mind map
2. Express categories you know/expect/manipulate
3. Organize and group
4. Discuss



Get my slides (pdf)

https://bit.ly/jmb-teaching

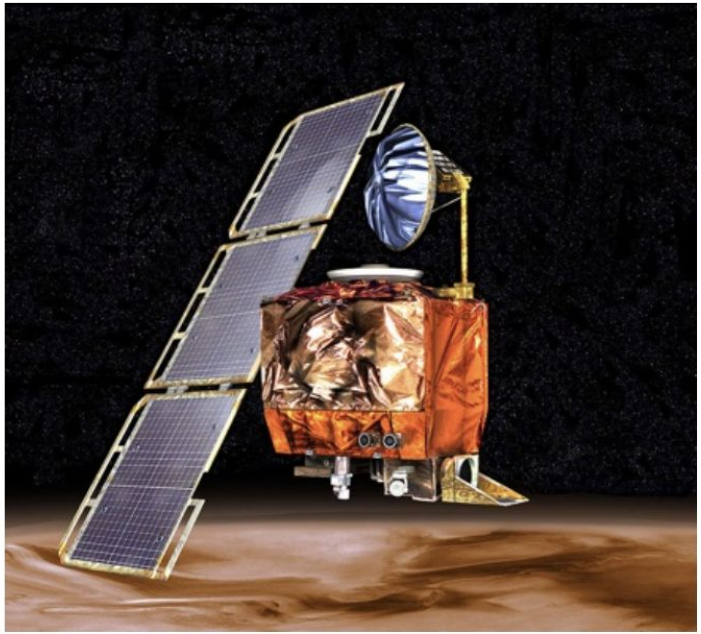
Context: why should you care?

Concrete example of Requirement problem…

https://www.youtube.com/watch?v=_47utWAoupo

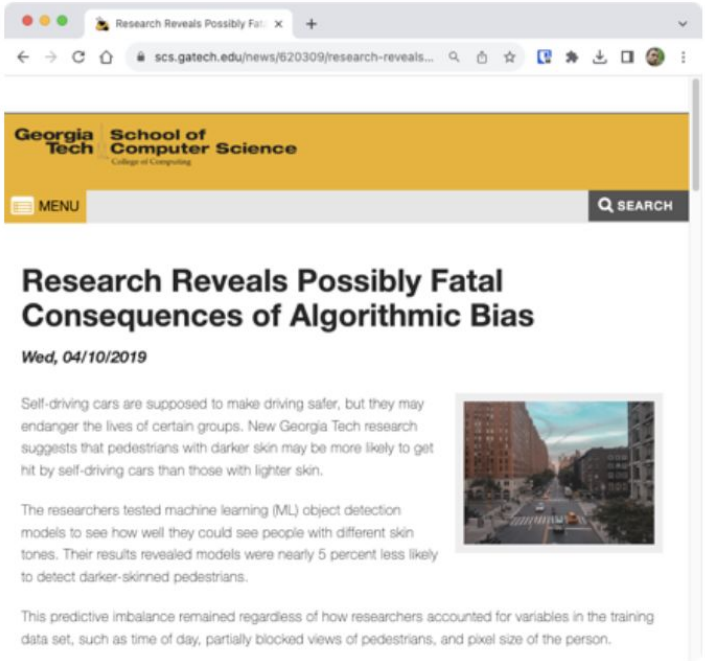
Space Exploration Fiascos

- **Ariane V - Flight v88** (1st flight)
  - Went kaboom after 37s
  - Estimated cost: **$370M USD**
    - Inflation + exchange: **$1B CAD**
  - **Bad code reuse, dead code, overflow,...**
- **Mars Climate Orbiter** (lost after 11 months)
  - **$110M USD** to build and deploy
  - **Unit conversion error** ($lb.s^2 \neq N.s^2$)



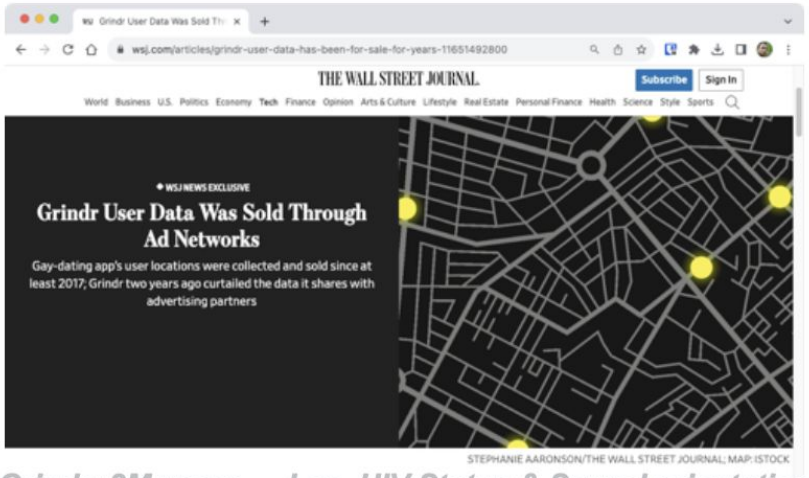
I'm not going to space, I want to do AI!

- **Tesla auto-pilot** anti-collision not able to identify black pedestrians at night
- **Amazon HR** recruitment assistant amplifying Ivy League recruitment biases
- **Google Photo** recognizing black people as gorillas or chimpanzees
- **Contact tracing** application during COVID-19
- **Microsoft navigation** system patented an "avoid ghetto" mode for pedestrian walks



I don't care, I want to be a mobile developer!

Waze Drive Now: 151M users — Home Location



Grindr: 2M users — Loc, HIV Status & Sexual orientation

***Leaking personal information**, **sold to** external **companies**.*

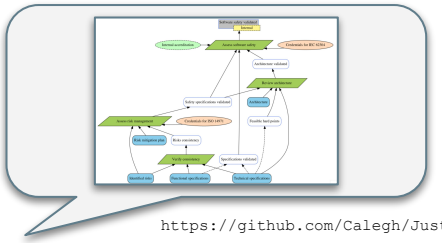
https://www.linkedin.com/posts/daniel-abrahams_reminder-people-dont-buy-products-they-ugcPost-7010015948820680704-CTJD?utm_source=share&utm_medium=member_android

People don’t buy products

They buy solutions to their problem

[...] they buy solutions to their problem

- **Play** with the product
  - Not so easy with an airplane...
- Don't need details
  - **Early** V&V
- Validation => **Rational**



https://github.com/Calegh/JustificationDiagram

Identifying **ambiguities**

Validation & Verification (V&V)

Does the right thing

- Validation
- « Building the right system »

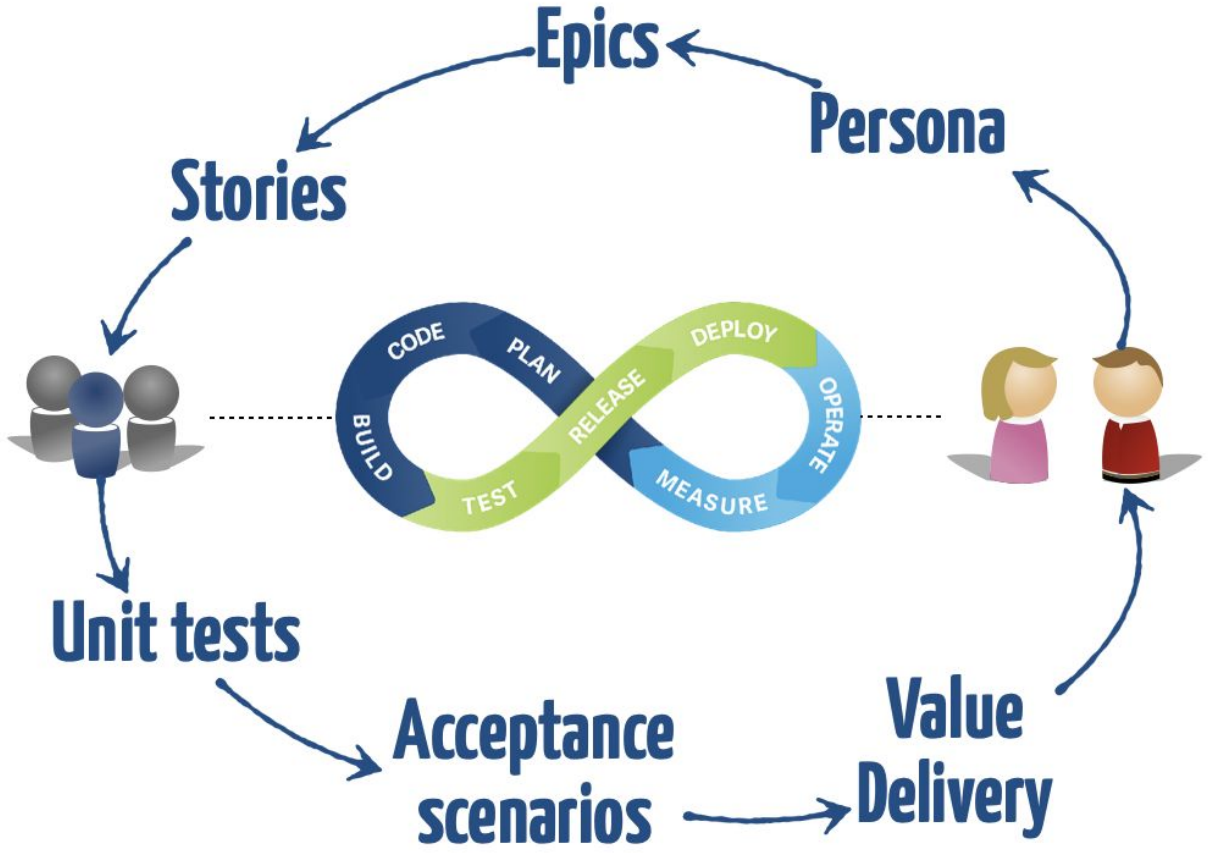
Does them right

- Verification
- « Building the system right »

https://www.canon.co.nz/software-solutions/iw-sam

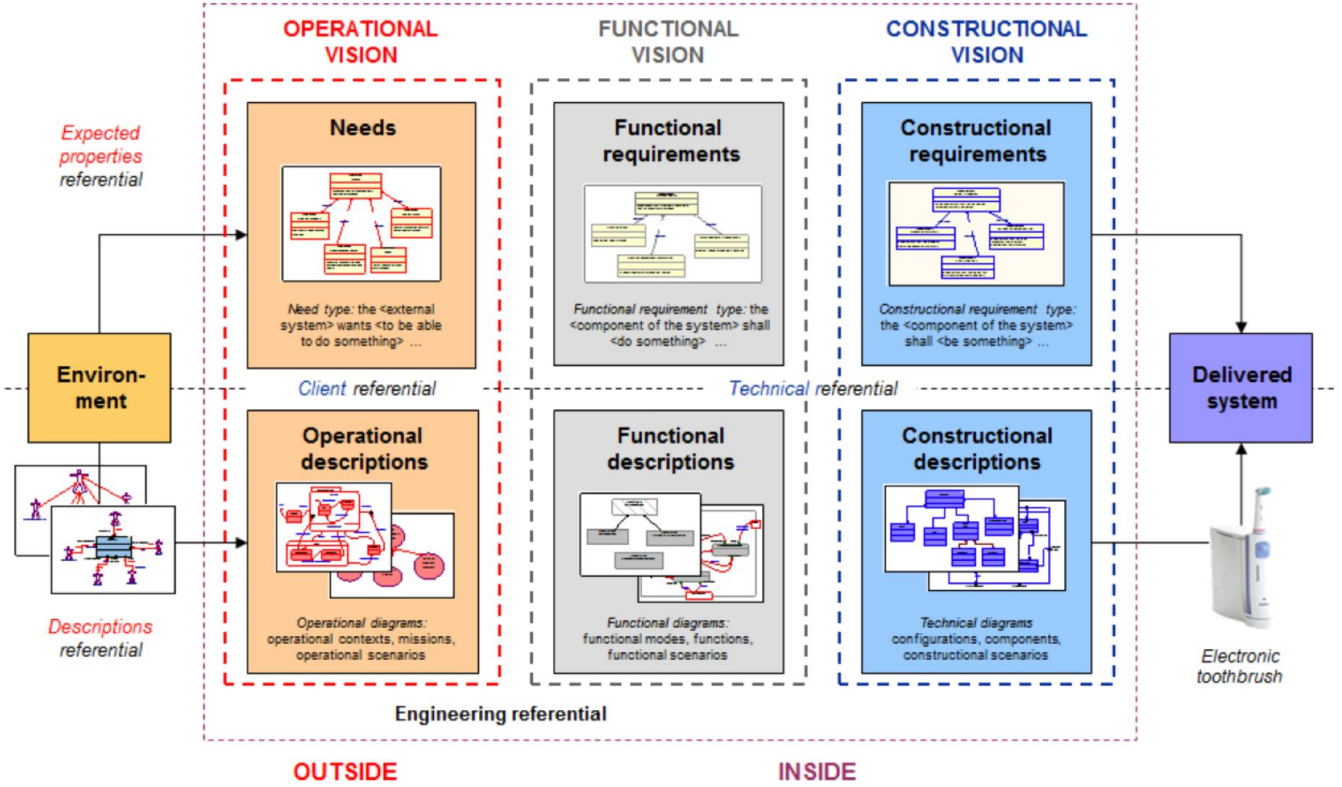
https://www.techopedia.com

Lean development



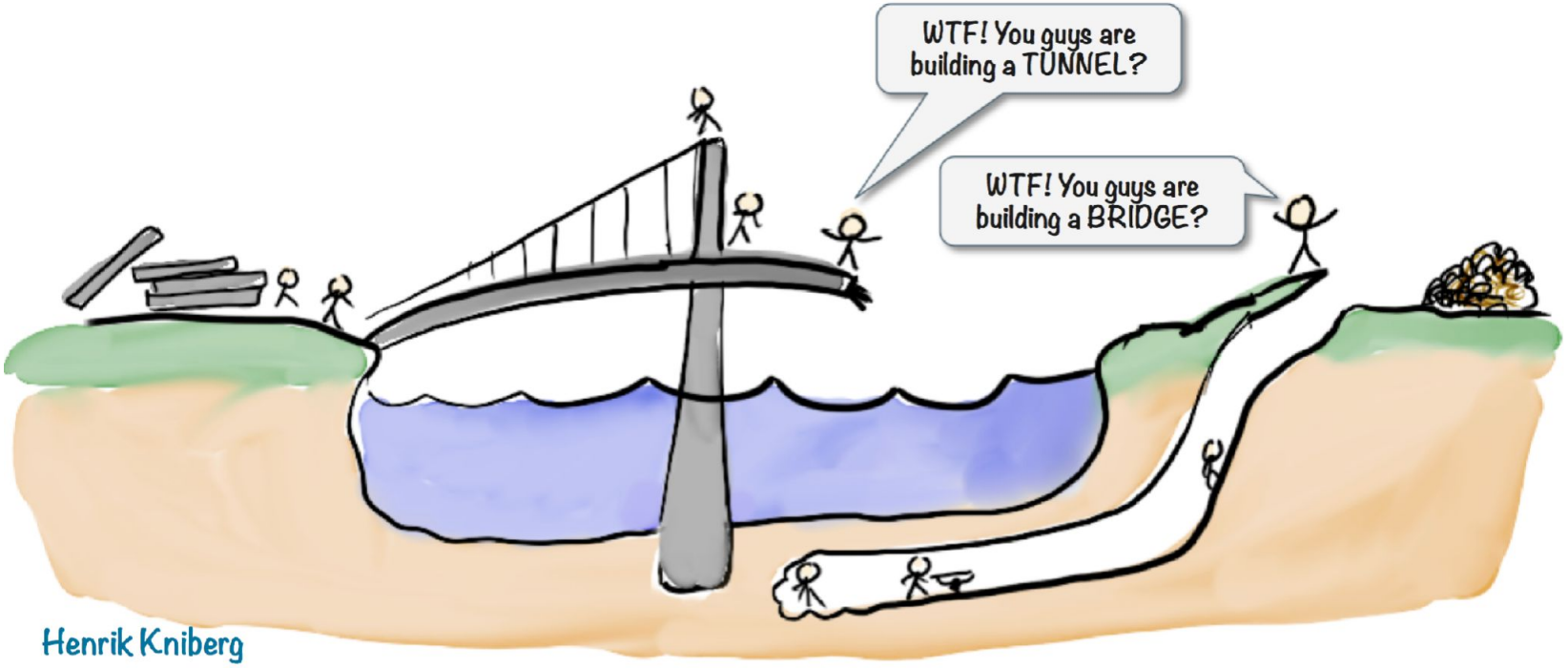
Source: http://meshfields.de/continuous-integration-testing-delivery-ionic2-hybrid-mobile-apps-buddybuild/

Systems Engineering

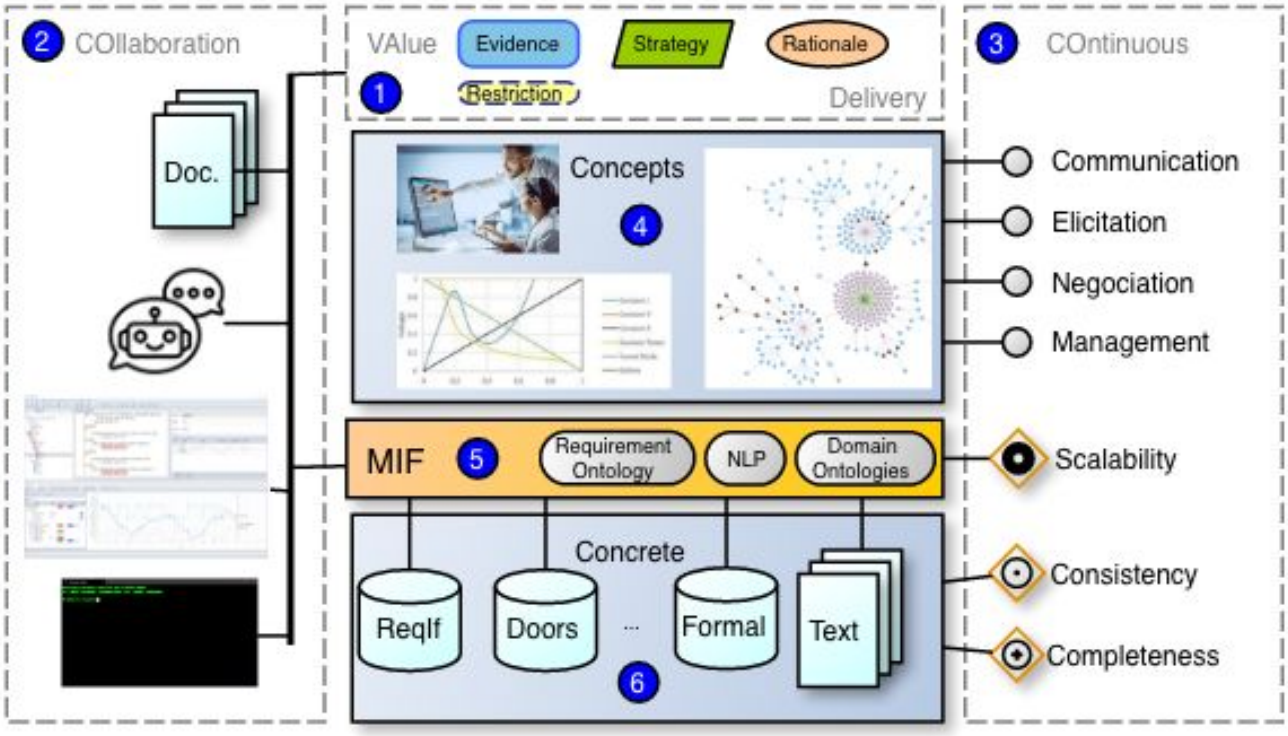


http://cesames.cn/wp-content/uploads/2020/06/CESAM-Systems-Architecting-Method-Pocket-Guide-CESAMES.pdf

Misalignment



Requirements as first-class citizens



IEEE/SWEBOK/ISO definition of a Requirement

“A **1.1 Definition of a Software Requirement**

At its most basic, a software requirement is a property that must be exhibited by something in order to solve some problem in the real world. It may aim to automate part of a task for someone to support the business processes of an organization, to correct shortcomings of existing software, or to control a device—to name just a few of the many problems for which software solutions are possible. The ways in which users, business processes, and devices function are typically complex. By extension, therefore, the requirements on particular software are typically a complex combination from various people at different levels of an organization, and who are in one way or another involved or connected with this feature from the environment in which the software will operate.

”

http://swebokwiki.org/Chapter_1:_Software_Requirements

Templates (docx, LaTeX, Google Doc, ...)

Goals

Goals are "needs of the target organization, which the system will address". While the development team is the principal user of the other books, the Goals book addresses a wider audience: essentially, all stakeholders (see Handbook).

It must contain enough information to provide — if read just by itself — a general sketch of the entire project. To this effect, chapter G.3 presents a short overview of the system and G.1 will typically include some key properties of the environment. As it addresses a wide readership, it should be clear and minimize the use of specialized technical terms. Together, G.1, G.2 and G.3 describe the rationale for the project. It is important to state these justifications explicitly. Typically, they are well understood at the start of the project, but management and priorities can change (see Handbook).

G.1 Context and overall objectives

High-level view of the project: organizational context and reason for building a system (see Handbook).

This section should not be empty (following the *Minimum Requirements Outcome Principle*, p.27 of the Handbook).

1 Example of numbered requirement that can be referenced.

G.2 Current situation

Current state of processes to be addressed by the project and the resulting system (see Handbook).

1 Goals

Contents

Comment: *Goals are "needs of the target organization, which the system will address". While the development team is the principal user of the other books, the Goals book addresses a wider audience: essentially, all stakeholders.*

1.1 G.1 Context and overall objective

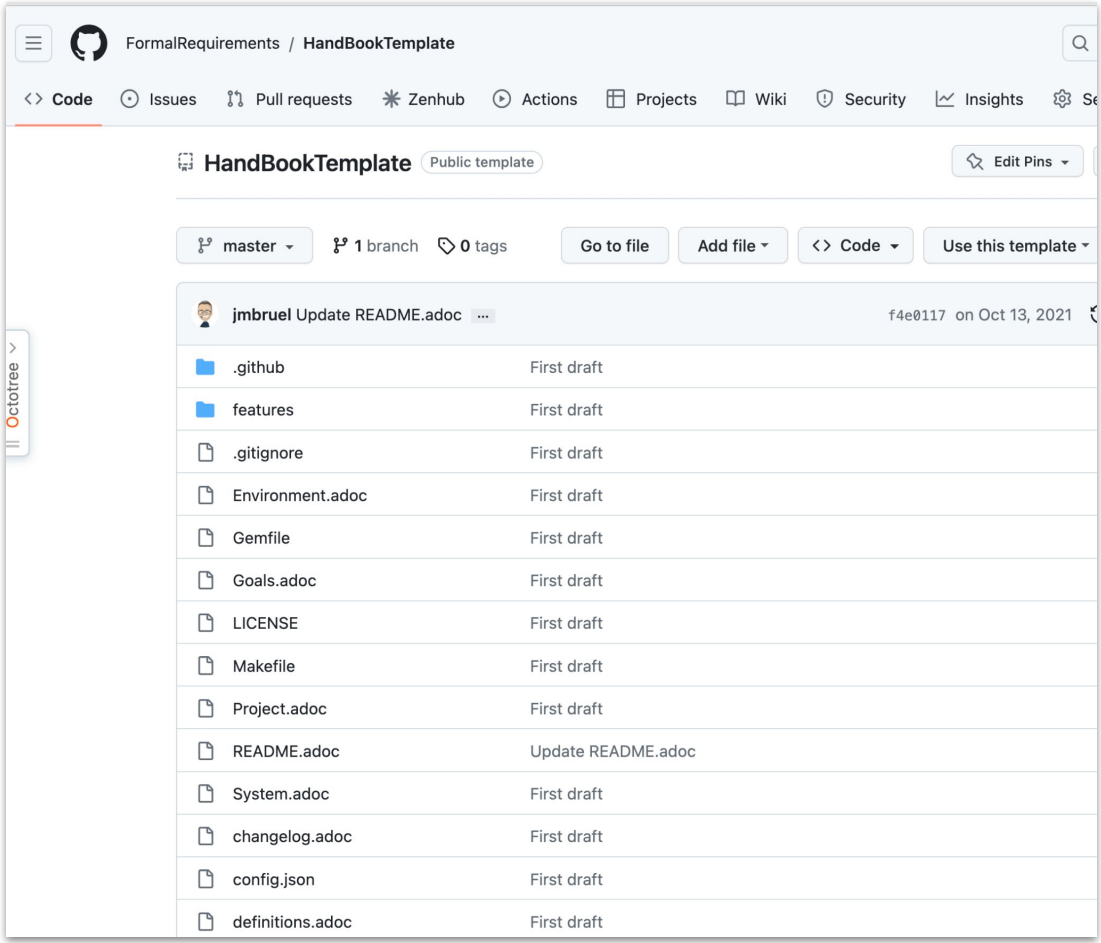
Comment: *High-level view of the project: organizational context and reason for building a system. This chapter should not be empty!*

Goal 1.1.1. This is a goal example. If you need explicit (and automatic) numbering, you can use the definitions in the `.tex` template. Is is refined by 1.2.1

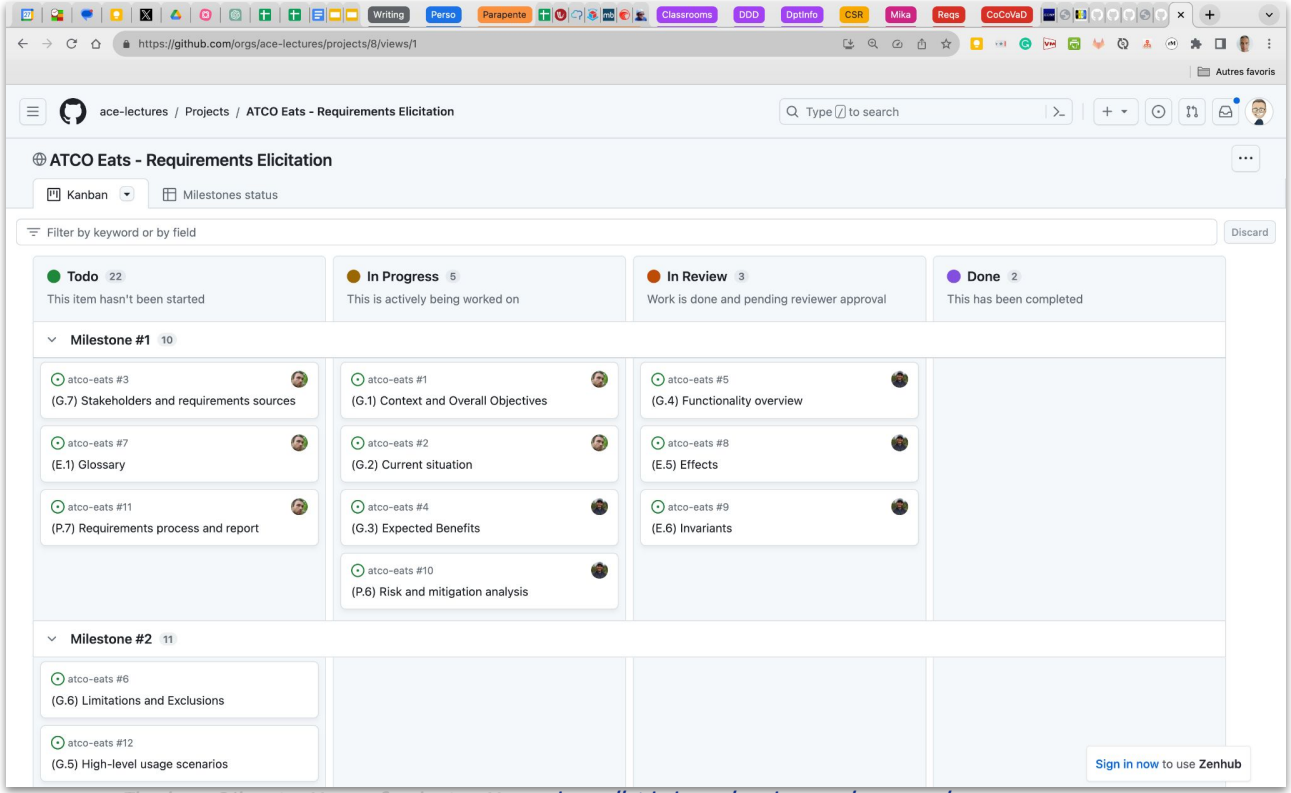
More than Word!

- Markdown-like format
- GitHub itself
- Quality metrics & rules **implemented**

Github repo template



PEGS chapters to organize requirements writing



Thanks to Sébastien Mosser for sharing. More at https://github.com/ace-lectures/atco-eats/

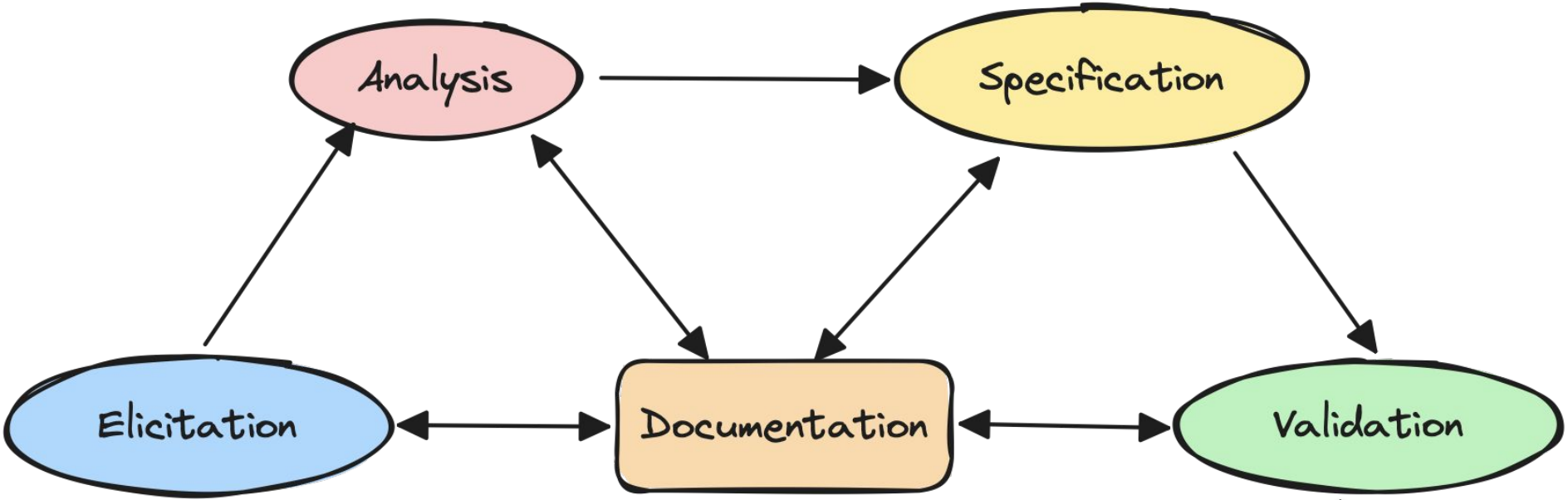
Requirements documents can be tested!

```
#-----------------------------
# language: en
Feature: Book mutual references
    The books should follow the mutual references rules.

Scenario: The Environment book must not refer to the Goals and Project books
    Given The Environment book
    Then No reference should include the Goals book
    And No reference should include the Project book
    And Only E.5 section can refer to the System book

Scenario: The Goals book must not refer to the Project and System books
    Given The Goals book
    Then No reference should include the Project book
    And No reference should include the System book

Scenario: The System book must not refer to the Project book
    Given The System book
    Then No reference should include the Project book
```

Requirements documents can be tested!

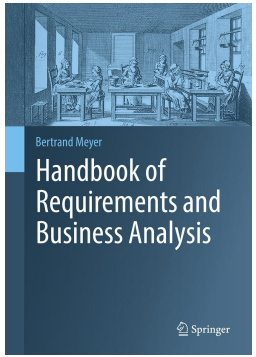
```
4   #-----------------------------
5   # language: en
6   Feature: Minimum Requirements Outcome Principle
7       The requirements effort must always produce the following elements.
8
9   Scenario: The Project book must have P3 P4 chapters
10      Given The Project book
11      Then P3 chapter must not be empty
12      And P4 chapter must not be empty
13
14  Scenario: The Environment book must have E3 chapter
15      Given The Environment book
16      Then E3 chapter must not be empty
17
18  Scenario: The Goals book must have G1 G3 G7 chapters
19      Given The Goals book
20      Then G1 chapter must not be empty
21      And G3 chapter must not be empty
22      And G7 chapter must not be empty
23
24  Scenario: The System book must have S1 S2 chapters
```

I'll try to cover all the aspects



Courtesy of Sébastien Mosser

Chapters from the Handbook



- Covered
- Overview
- Not covered

0. Preface
1. Requirements: Basic concepts and definitions
2. Requirements: General principles
3. Standard plan for requirements
4. Requirements quality and verification
5. How to write requirements
6. How to gather requirements
7. Scenarios: use cases, user stories
8. Object-oriented requirements
9. Benefiting from formal methods
10. Abstract data types
11. Are my requirements complete?
12. Requirements in the software lifecycle

Requirements for this course

Hard skills

- Github account
- Scrum/Agile knowledge
- Formal (maths) skills
- Windows PC (I know, I know, ...)

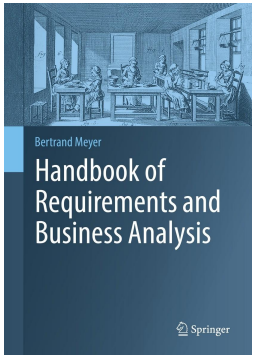
Soft skills

- Communication skills
- Writing skills
- “Out of the box” thinking

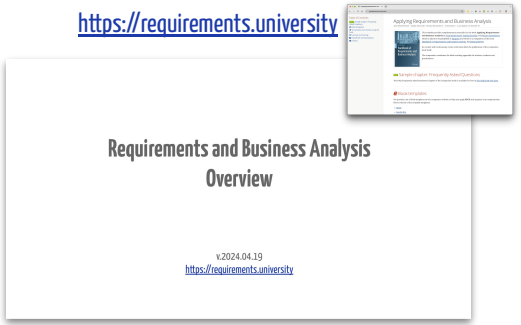
I need your help for this course...

- Design skills (any artist in the audience?)
- Data, reference, evidence (don't use Chat GPT, please ;-)
- Github (hooks, CI/CD)
- BDD (Gherkin/Cucumber)
- ... (be proactive)

Let's move to PEGS Handbook



https://requirements.university



0. Preface
1. Requirements: Basic concepts and definitions
2. Requirements: General principles
3. Standard plan for requirements
4. Requirements quality and verification
5. How to write requirements
6. How to gather requirements
7. Scenarios: use cases, user stories
8. Object-oriented requirements
9. Benefiting from formal methods
10. Abstract data types
11. Are my requirements complete?
12. Requirements in the software lifecycle

- Covered
- Overview
- Not covered

DON'T TRUST THIS!

Discussions time

Get the slides

https://bit.ly/jmbruel

@jmbruel

UNIVERSITÉ TOULOUSE Jean Jaurès

SM@RT

IRIT