Requirements and Business Analysis

Preface Chapter

v.2024.04.19

https://requirements.university

Official reference:

https://requirements.university

How to cite:

"Jean-Michel Bruel, Handbook of Requirements and Business Analysis Teaching Materials. https://requirements.university."

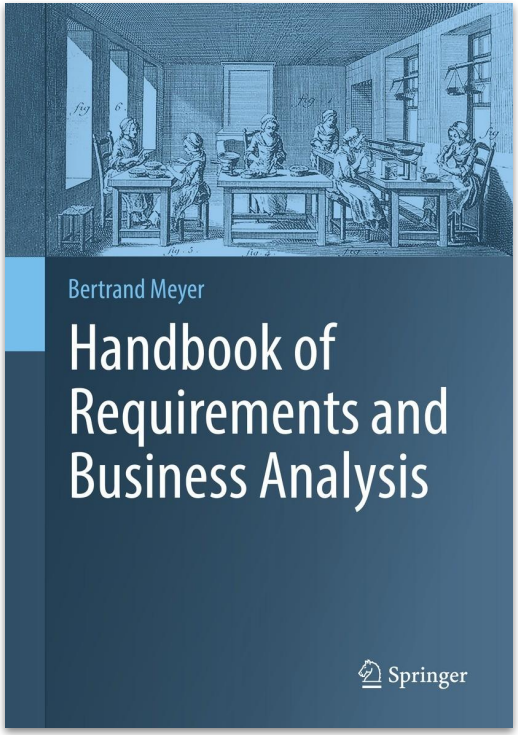
About the slides' author

- Professor at Toulouse University
  - Teaching modeling, requirements and DevOps
- Member of the CNRS-IRIT Laboratory
  - Model-Based Systems Engineering
- Leader of the companion book

https:/bit.ly/jmbruel

Get the slides (pdf)

Disclaimer



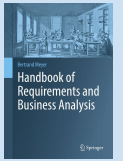
This material is based on this book, by Bertrand Meyer.

But it only reflects the point of view of its author.

It is part of additional materials developed

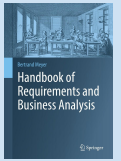
and available at https://requirements.university

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



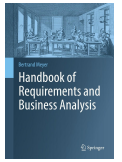
Outline

- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek

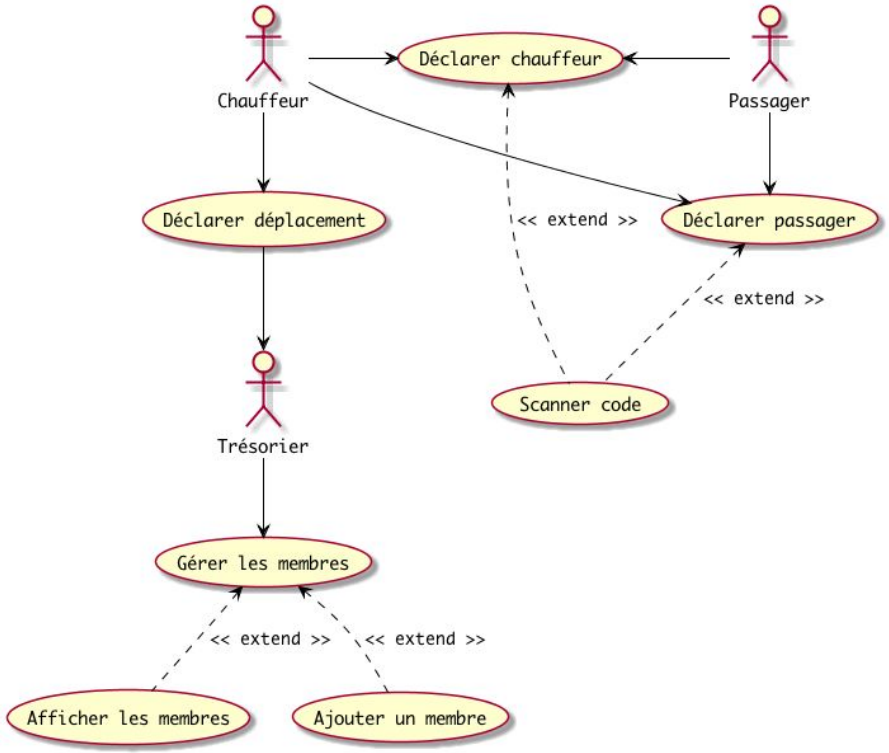


Outline

- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek



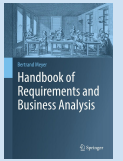
Don't need requirements, I have UML Use Cases



Requirements are old school, we're Agile

SCRUMstudy
Targeting success
Global Accreditation Body for
Scrum and Agile Certifications

https://www.scrumstudy.com/article/agile-vs-

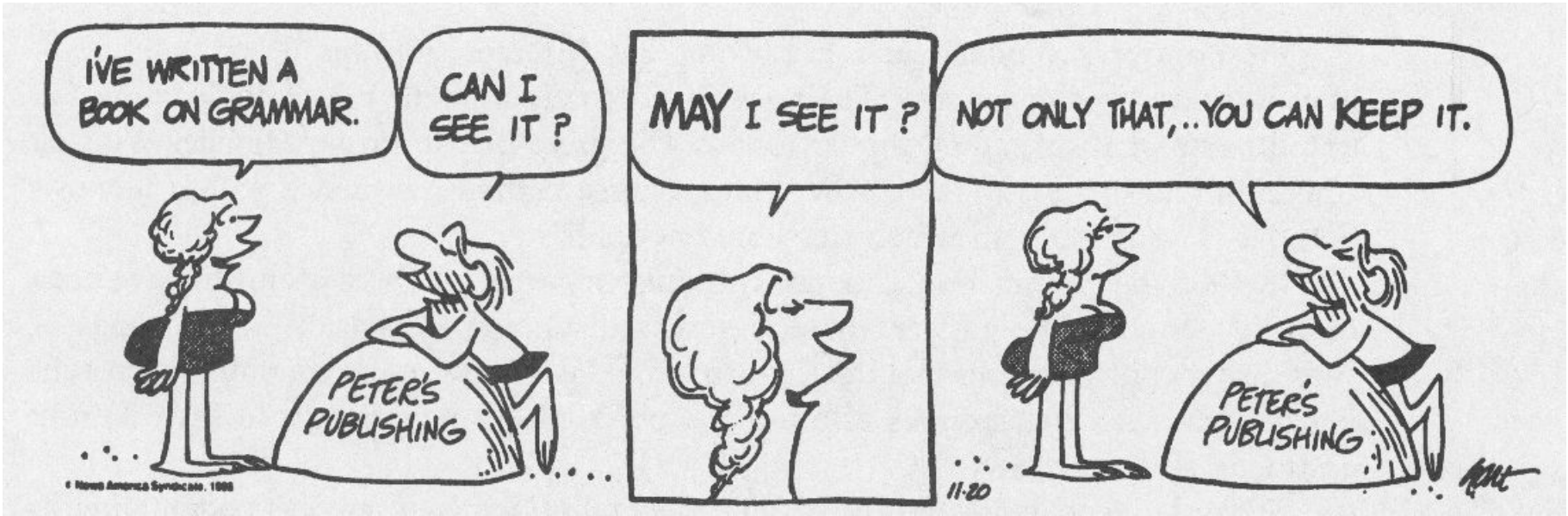


Outline

- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek

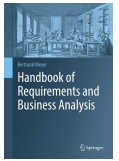
Preface

We are not prescriptive!



https://noharmspilt.com/2015/10/09/descriptive-vs-prescriptive-grammar/

IEEE/SWEBOK/ISO (vague) definition of a Requirement



Preface

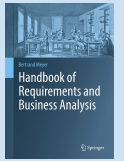
“A **1.1 Definition of a Software Requirement**

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.

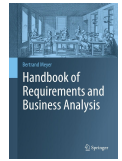
”

http://swebokwiki.org/Chapter_1:_Software_Requirements

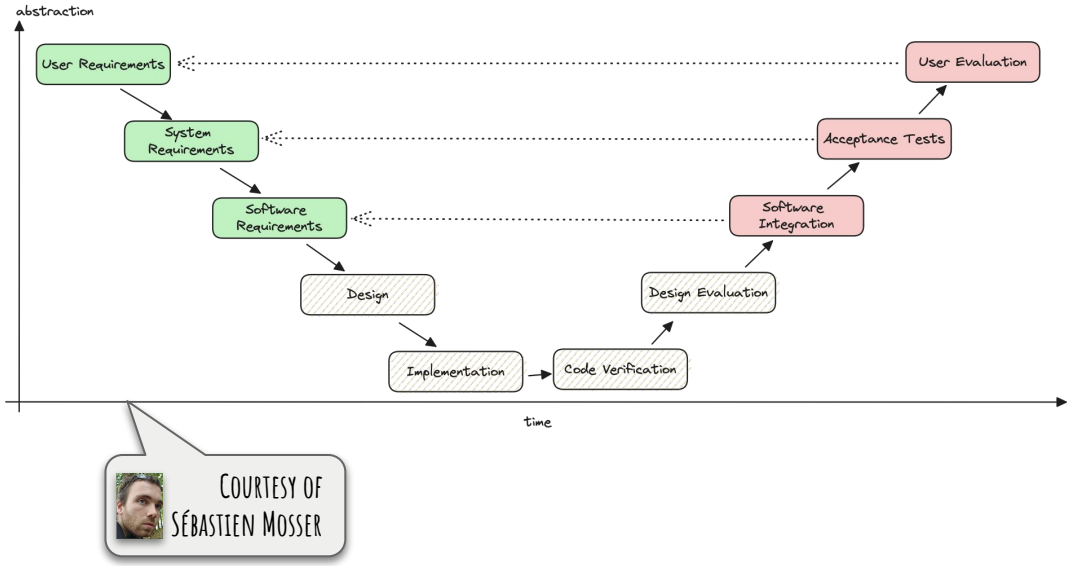
Outline

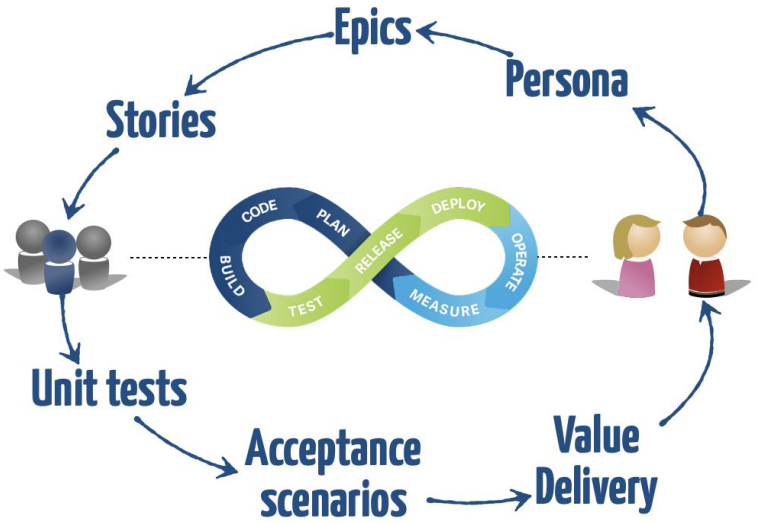


- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek

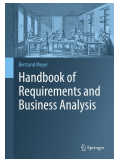


Between “Big Upfront” and “Just enough”

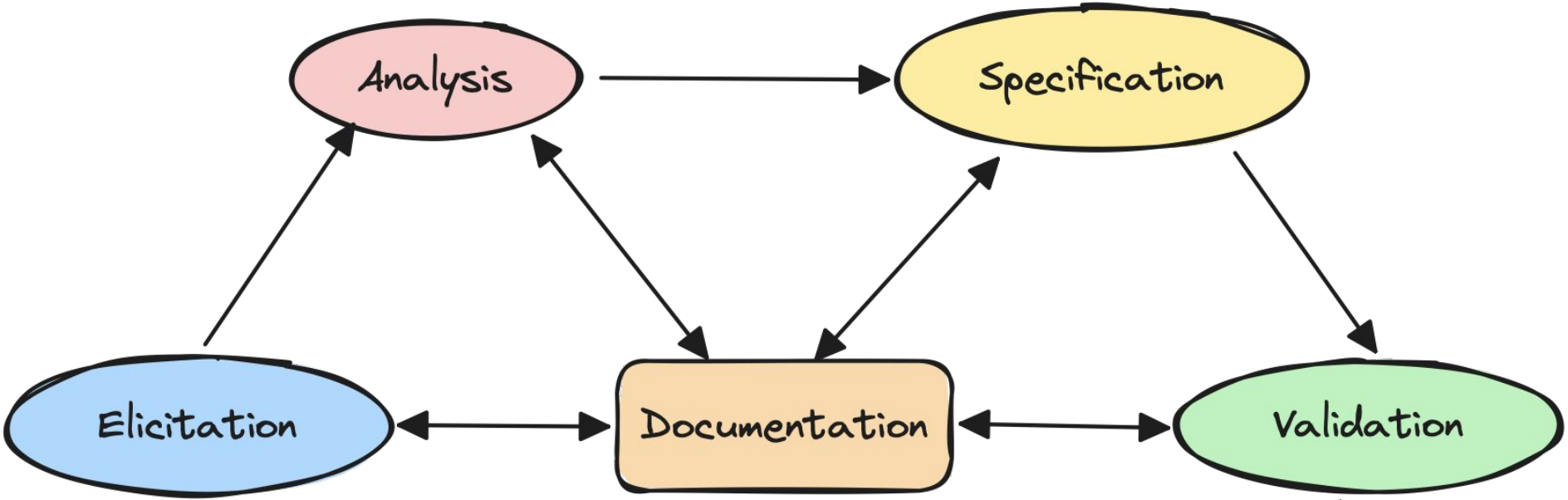


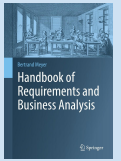


Source: http://meshfields.de/continuous-integration-testing-delivery-ionic2-hybrid-mobile-apps-buddybuild/



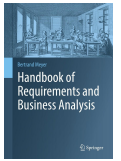
Between “Big Upfront” and “Just enough”





Outline

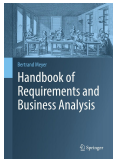
- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek



Key ideas

- A Standard Plan
- A proper scope for requirements
- Requirements as a question-and-answer device
- Not just documents
- Just enough requirements
- Upfront and evolving
- Requirements are software
- Requirements as living assets
- Taking advantage of the object-oriented method
- Taking advantage of formal approaches

A Standard Plan



Preface

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

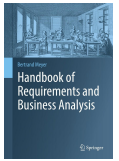
Contents

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



A proper scope for 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*

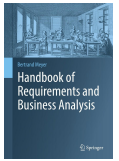
Environment (E)

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

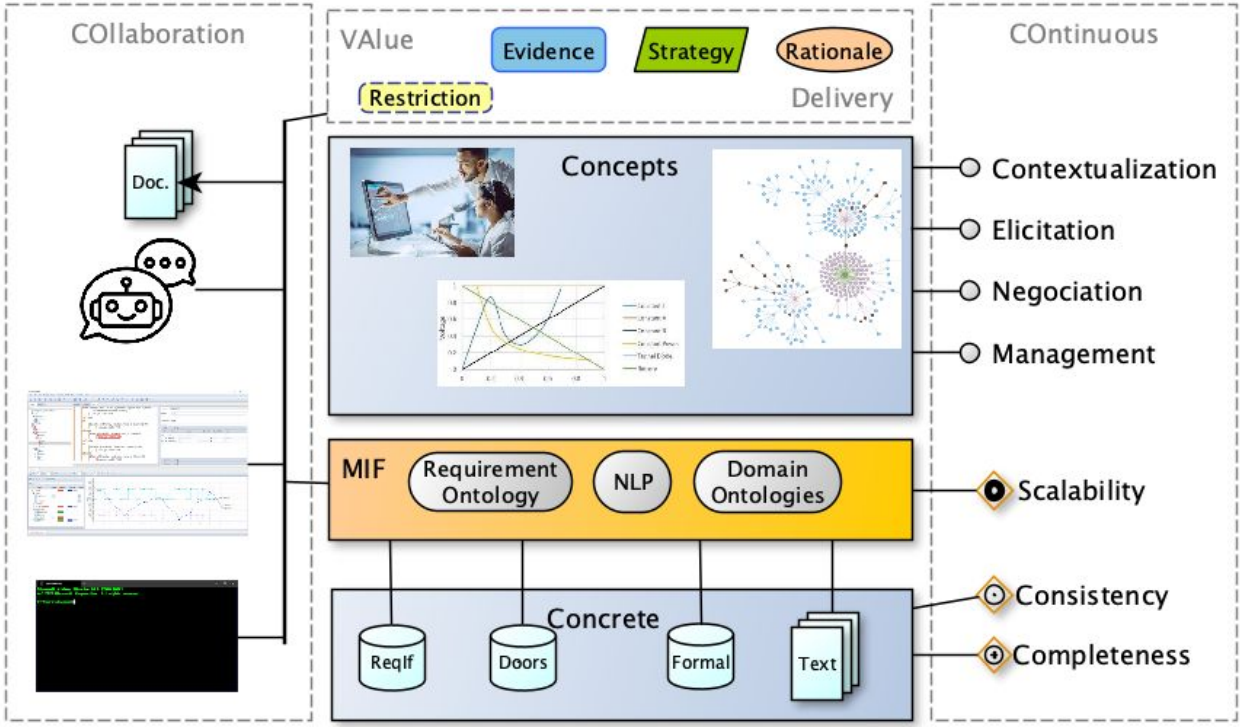
* These chapters should not be empty
(following the Minimum Requirements Outcome Principle)

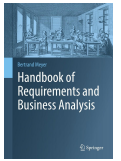


Requirements as a question-and-answer device

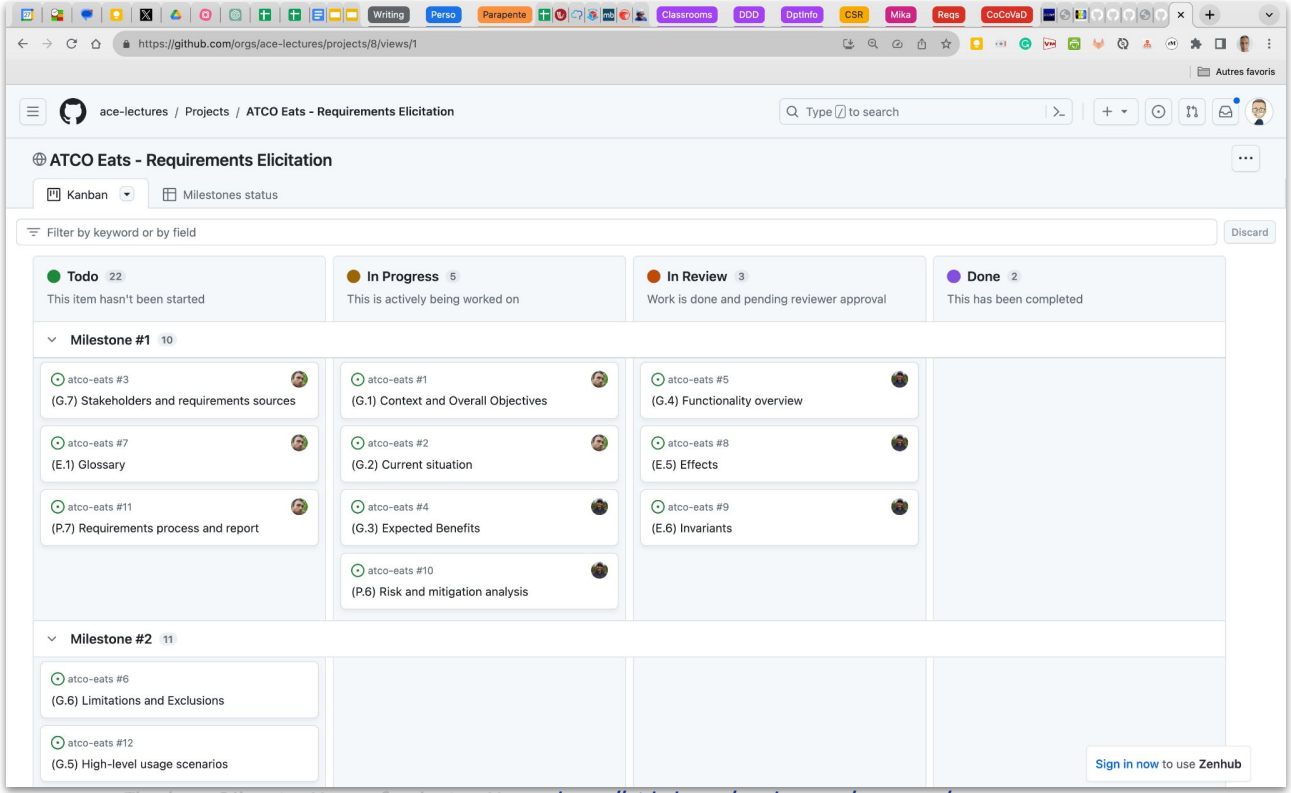
Example of the CoCoVaD

Airbus MBSE Chair





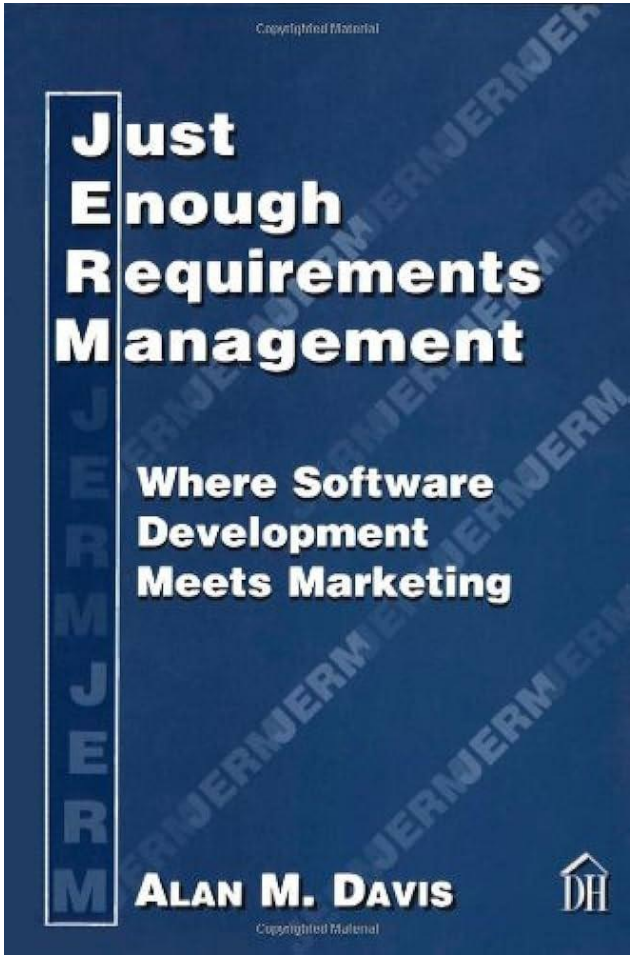
Not just documents

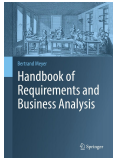


Thanks to Sébastien Mosser for sharing. More at https://github.com/ace-lectures/atco-eats/

Just enough requirements

2005 reference!

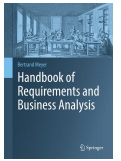




Preface

Preface

Just enough requirements



Preface

Upfront and evolving

Total Entries:	398				
Components:	25	Open:	23	Closed:	2
Requirements:	99	Open:	32	Closed:	67
Design Definitions:	211	Open:	52	Closed:	159
Sub-Tasks:	63	Open:	0	Closed:	63
Links to Code:	892	Manual created Links:	338	Committed Links:	554

CO-90 -- GCS Middleware [Component]

Status: Open

Description:
Handles connections between Dronology and Ground Control Stations (GCS). Forwards commands monitoring and other messages from Dronology to its registered GCS and passes messages describing the state of the UAVs managed by each GCS back to dronology.

Contained Elements: DD-354 - DD-361 - DD-710 - DD-711 - DD-712 - DD-713 - DD-715 - DD-716 - DD-718 - DD-719 - DD-720 - DD-721 - DD-723 - DD-724 - DD-727 - DD-728 - DD-730 - DD-731 - DD-732 - DD-733 - DD-734 - DD-735 - DD-737 - DD-763 - DD-768 - RE-160 - RE-709 - RE-714 - RE-722 - RE-729 - RE-736

CO-91 -- GCS [Component]

Status: Open

Description:
Python based system that manages and controls UAVs. Communicates with Dronology via the Ground Station middleware. Each GCS is responsible for communicating directly with each UAV sending it commands and monitoring its state including its current position flight mode and health.

Contained Elements: DD-740 - DD-742 - DD-743 - DD-744 - DD-745 - DD-747 - DD-748 - DD-749 - DD-750 - DD-752 - DD-753 - DD-755 - DD-756 - DD-757 - RE-235 - RE-739 - RE-741 - RE-746 - RE-751 - RE-754

CO-105 -- UI Real-Time Flight View [Component]

Status: Open

Description:
Manages all aspects of displaying flights and UAVs in real-time and interacting with them. The flight view displays active routes UAV coordinates and their current health. The map uses zoom and panning features to follow one or more selected UAV.

Contained Elements: DD-113 - DD-121 - DD-229 - DD-682 - DD-683 - DD-684 - DD-685 - DD-686 - DD-687 - DD-688 - DD-690 - DD-692 - DD-694 - DD-696 - DD-697 - DD-699 - RE-114 - RE-120 - RE-681 - RE-689 - RE-691 - RE-693 - RE-695 - RE-698

CO-184 -- Internal Simulator [Component]

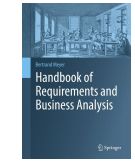
Status: Closed

Description:
The internal simulator provides low-fidelity features for supporting quick initial tests of a virtual UAV. Features include takeoff goto land and battery health.

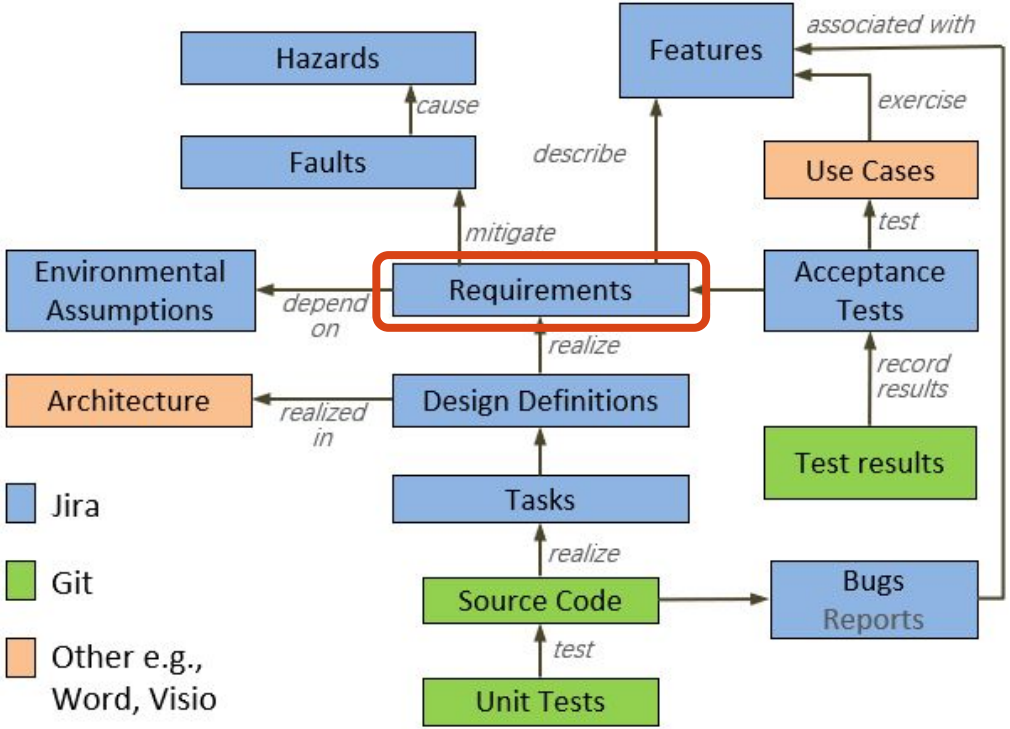
Contained Elements: RE-593 - RE-594 - RE-595 - RE-596 - RE-597

http://sarec.nd.edu/dronology/

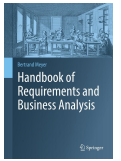
Requirements are software



Preface



http://sarec.nd.edu/dronology/

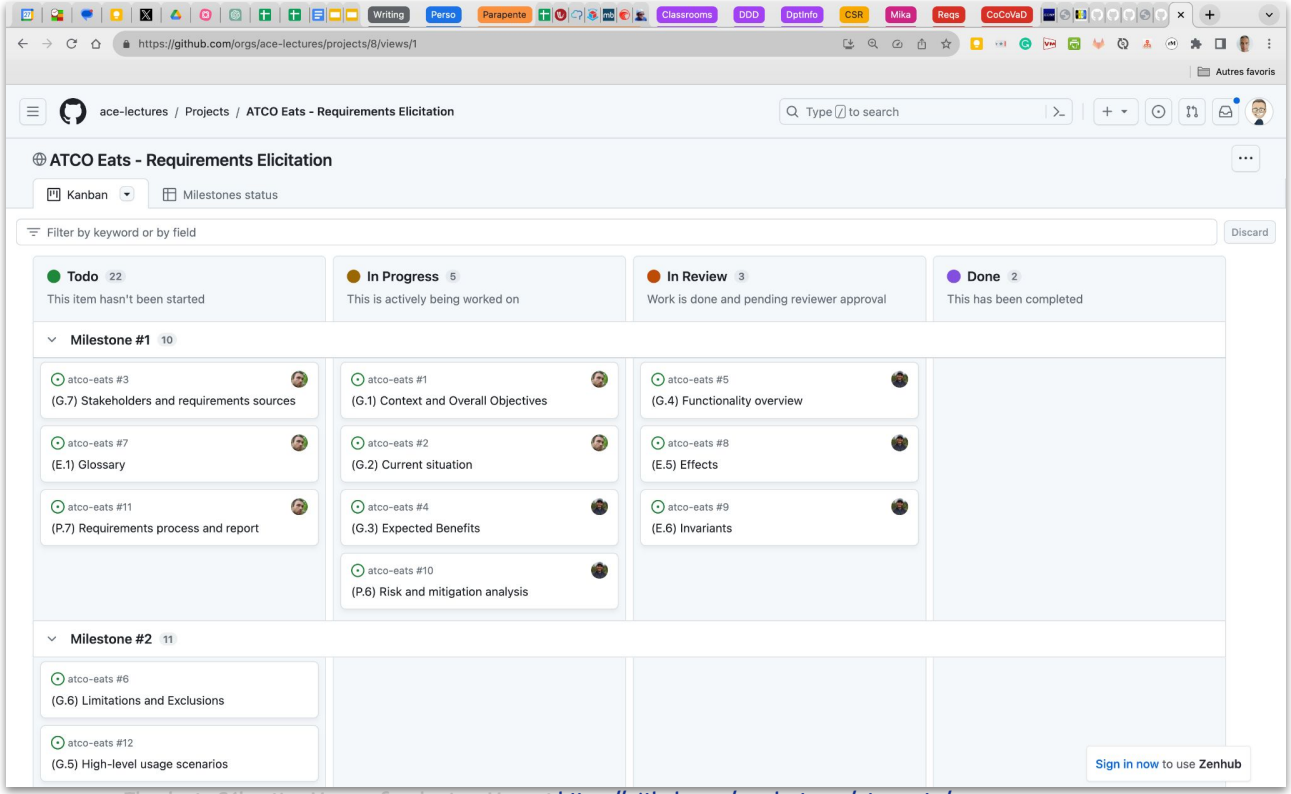


Requirements are software

They can be tested!

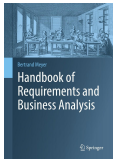
```
#----------------------------
# language: en
Feature: Book mutual references
    The books should follow the mutual references rules.

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 as living assets

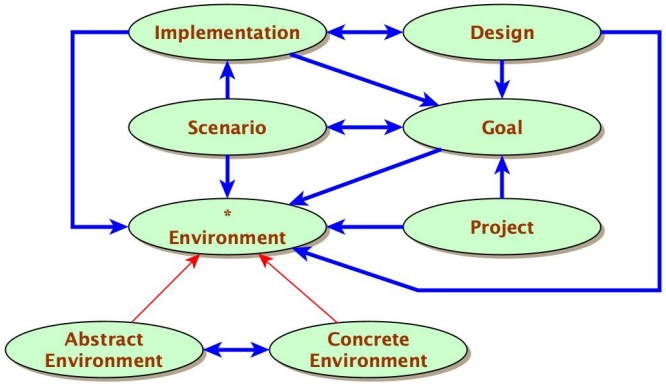


Thanks to Sébastien Mosser for sharing. More at https://github.com/ace-lectures/atco-eats/

Taking advantage of the object-oriented method



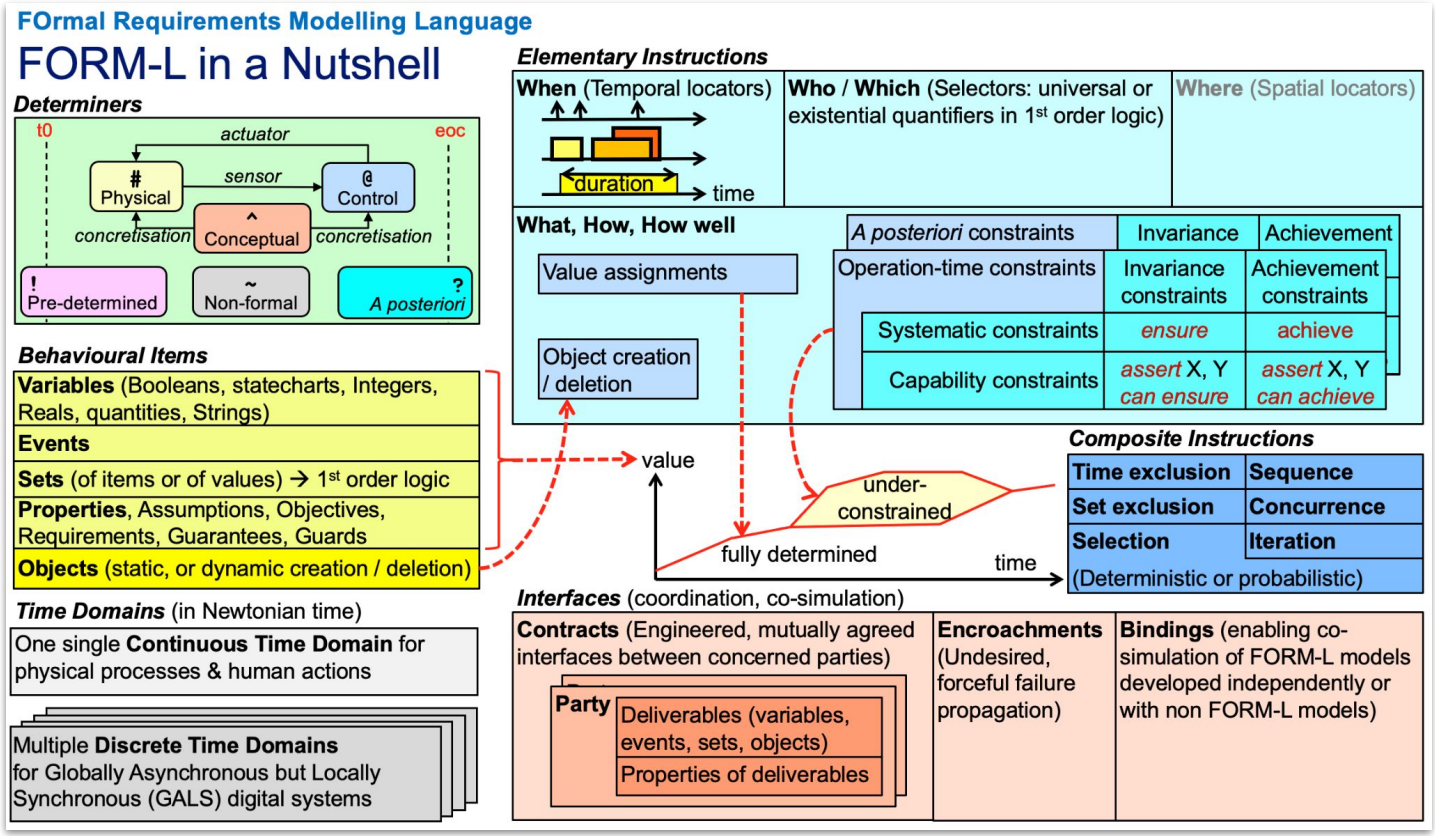
Preface

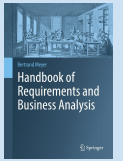


Extract of the Sensor.e *class*

```
deferred class
    SENSOR

feature
    position: LOCATION_3D
        --location in the world coordinates of the scene

    update_rate: REAL
        --sensor update rate
end
```

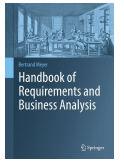
Taking advantage of formal approaches





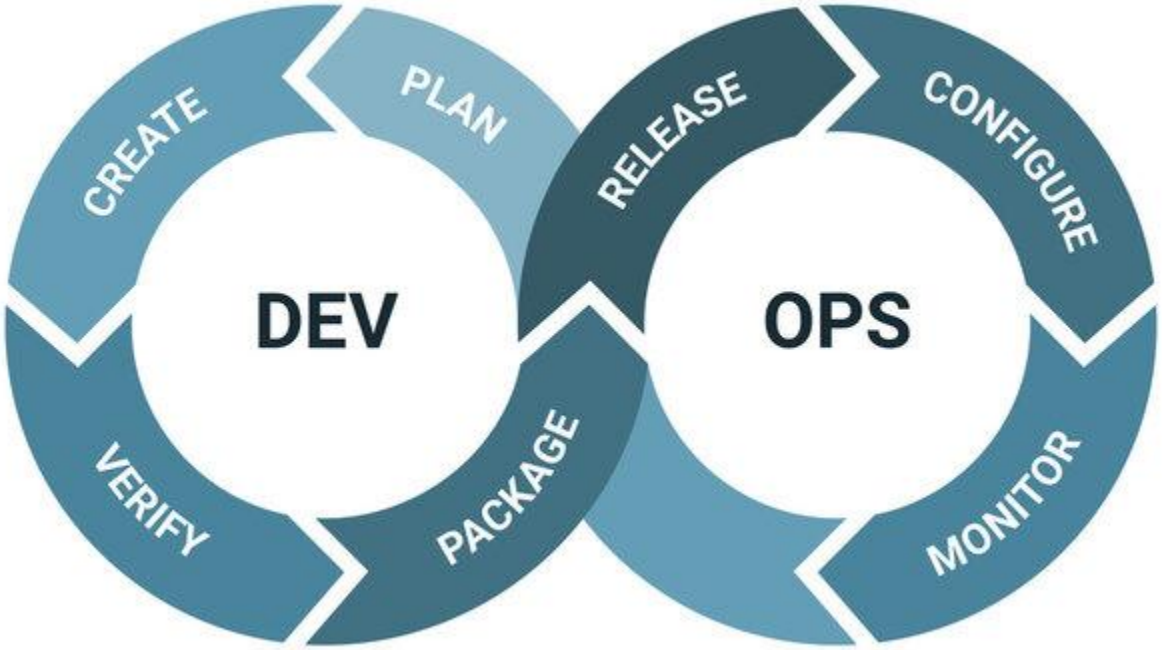
Outline

- Obstacle to quality
- Descriptive vs Prescriptive
- A balanced view
- Key ideas
- Geek and non-geek



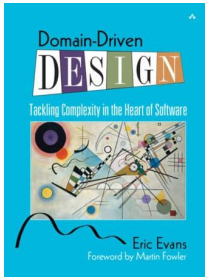
Dev vs Ops?

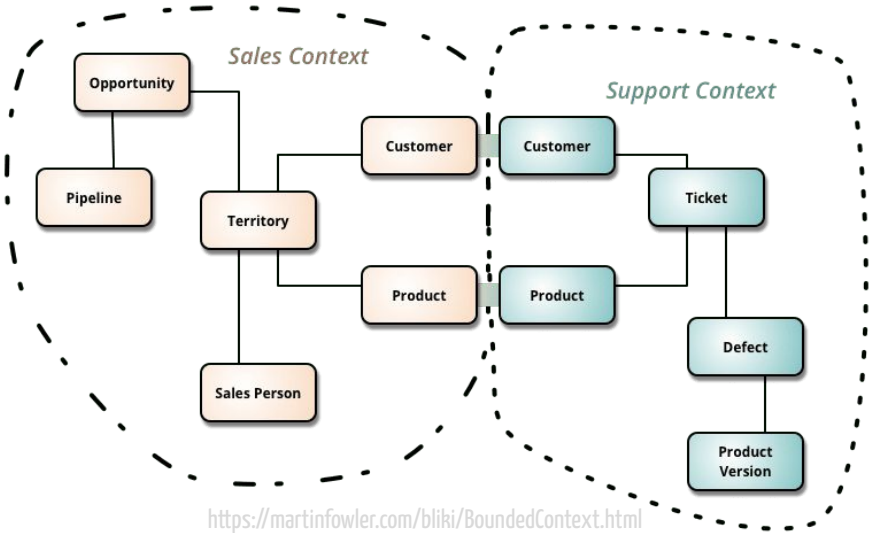
Not anymore!



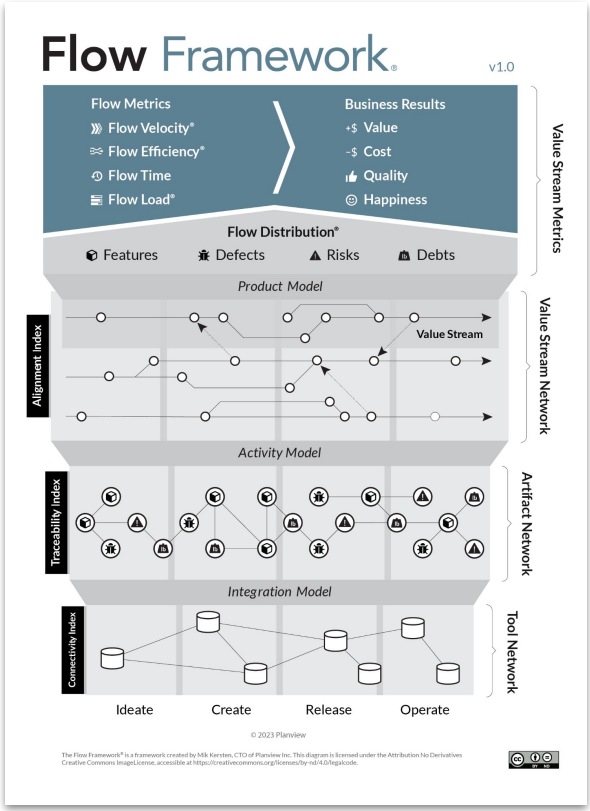
Engineering vs Business?

Not anymore!

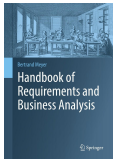




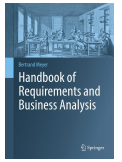
https://martinfowler.com/bliki/BoundedContext.html



https://flowframework.org/



Preface



And what about AI?

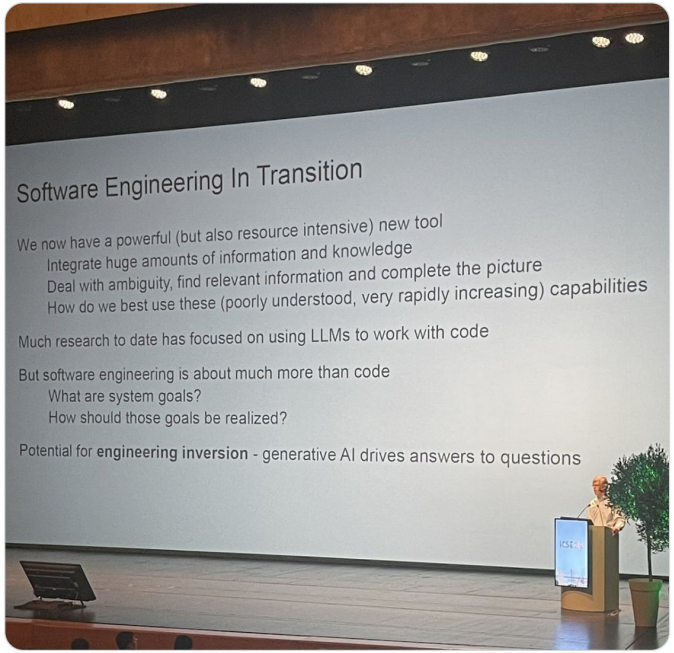
Still room for human…

…especially in RE!

Nelly Bencomo
@nellybencomo

SE in transition by Martin Rinard's keynote
#icse2024 the slide says a lot! What are the system goals? How to realise them?
It looks that @ieee_re RE has a big role

Traduire le post



11:12 AM · 18 avr. 2024 · **679** vues

https://t.co/CxFcdgttQ2

Discussions time

Get the slides

https://requirements.university