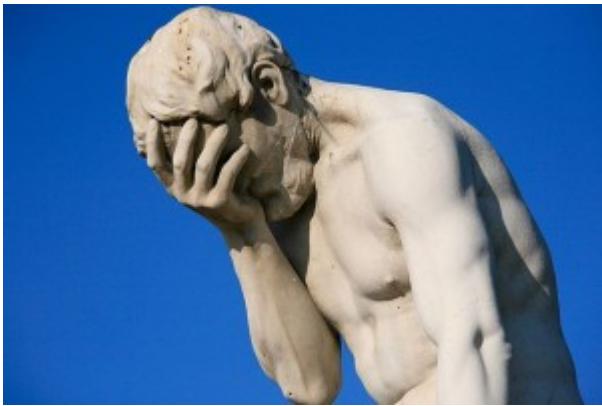


# Big Picture

## Table of Contents

1. Requirements failures .....	1
1.1. TUI's reservation system .....	2
1.2. Integration vs. Unit testing .....	2
2. DevOps / Agile / Test / Requirements .....	3
2.1. Example of approach (RE'18 tutorial) .....	3
2.2. Common situation .....	3
2.3. Requirements Engineering Process .....	4
2.4. Requirements in industry = ALM .....	6
2.5. Problems... .....	6
3. Situation overview .....	6
3.1. Typical situation .....	6
3.2. Who will use the product? .....	7
3.3. Writing Epics & Stories .....	7
3.4. Specifying with Stories .....	7
4. Testing .....	8
4.1. Test-Driven Development .....	8
4.2. Behavior-Driven Development .....	8
4.3. Quality Assessment .....	9
4.4. Automation (and CI) .....	9
5. Who are your clients? .....	10
5.1. Your client(s) .....	10
5.2. Your teacher(s) .....	10
5.3. At the same time! .....	11
5.4. Minimal Viable Product (usual) .....	12
5.5. Minimal Viable Product (improved) .....	13
5.6. MVP & EPICS .....	13
6. Back to the requirements .....	14
7. Stakeholders Value Networks .....	15
8. Traceability .....	15
Appendix A: Useful links .....	16
Appendix B: Credits .....	16

## 1. Requirements failures



## 1.1. TUI's reservation system

Tui plane in ‘serious incident’ after every ‘Miss’ on board was assigned child’s weight

Service from Birmingham to Majorca took off with less thrust because pilot thought it was 1,200kg lighter



▲ Tui says the software error that affected flights including one from Birmingham to Majorca has been corrected.  
Photograph: Sancia Steinbach/EPA

Figure 1. (source [here](#))

It was programmed in an unnamed foreign country where the title “Miss” is used for a child and “Ms” for an adult female.

## 1.2. Integration vs. Unit testing



## 2. DevOps / Agile / Test / Requirements

- Plan, Test, OK
- But towards WHAT?

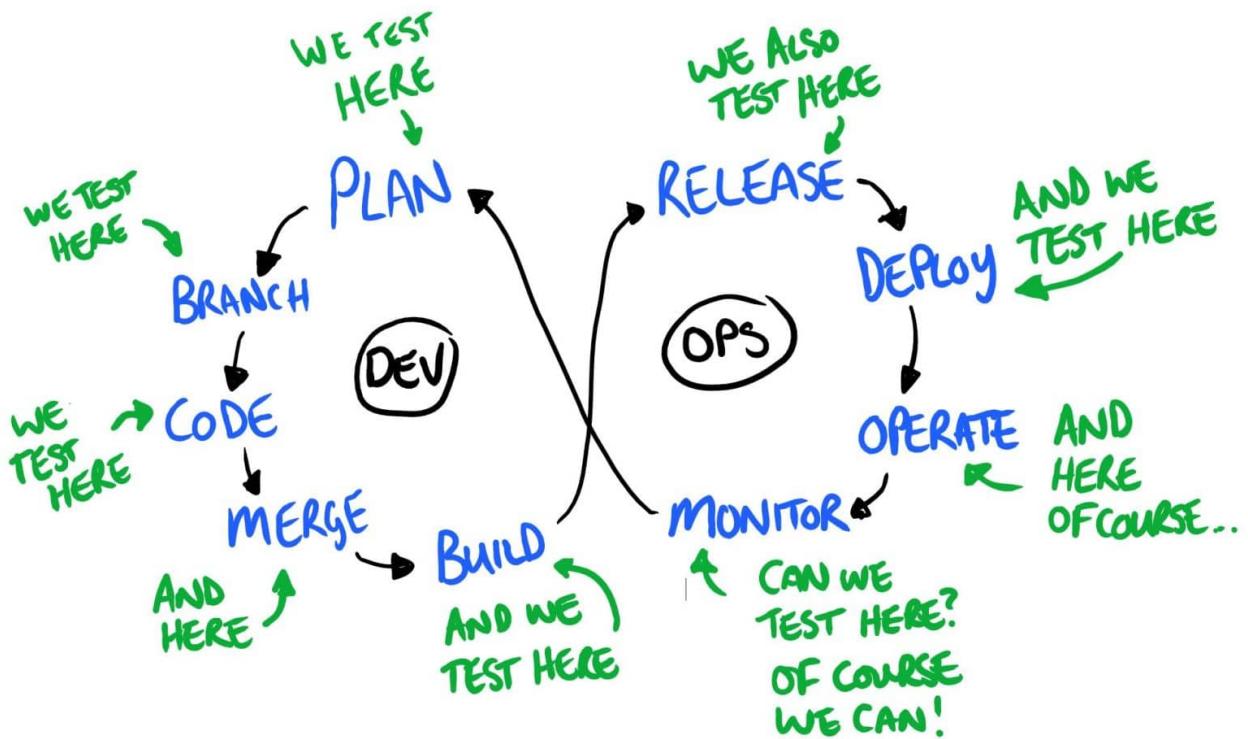


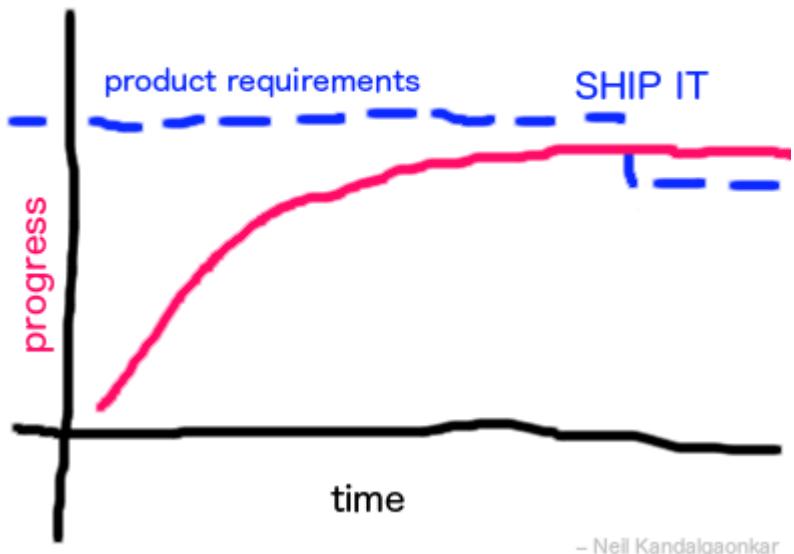
Figure 2. Tests in DevOps

### 2.1. Example of approach (RE'18 tutorial)

- Express requirements through User Stories
- Formally express US acceptance tests (BDD/TDD)
- Use CI/CD to ensure feature availability

### 2.2. Common situation

the way of all projects



– Neil Kandalgaonkar

Figure 3. The way of all projects (source [Twitter](#))

## 2.3. Requirements Engineering Process

- Requirements Elicitation
- Requirements Analysis & Negotiation
- Requirements Validation
- Requirements Documentation
- Requirements Management

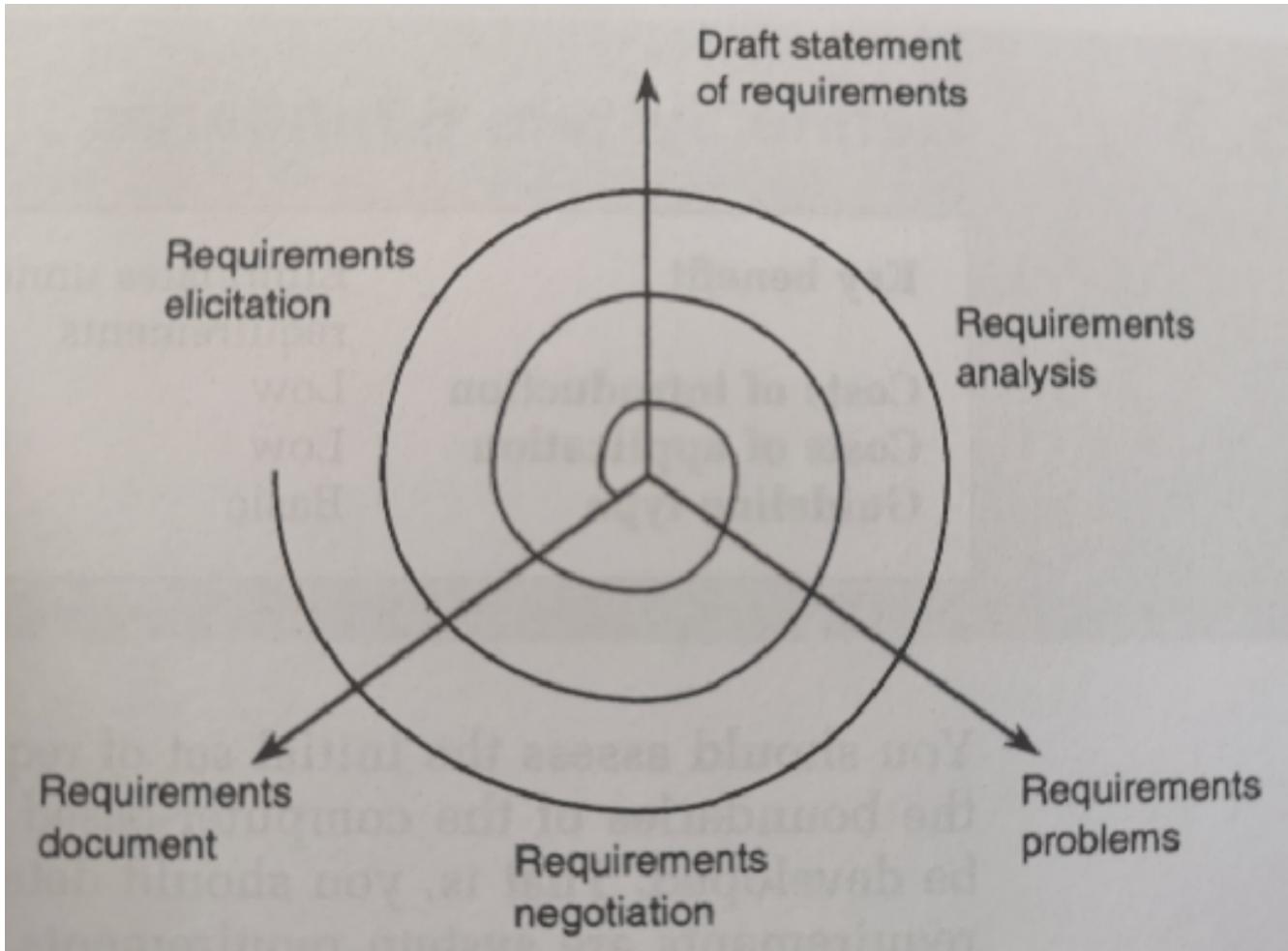


Figure 4. Requirements Engineering Process (source [Sommerville & Sawyer 1997])

Expected properties vs. description

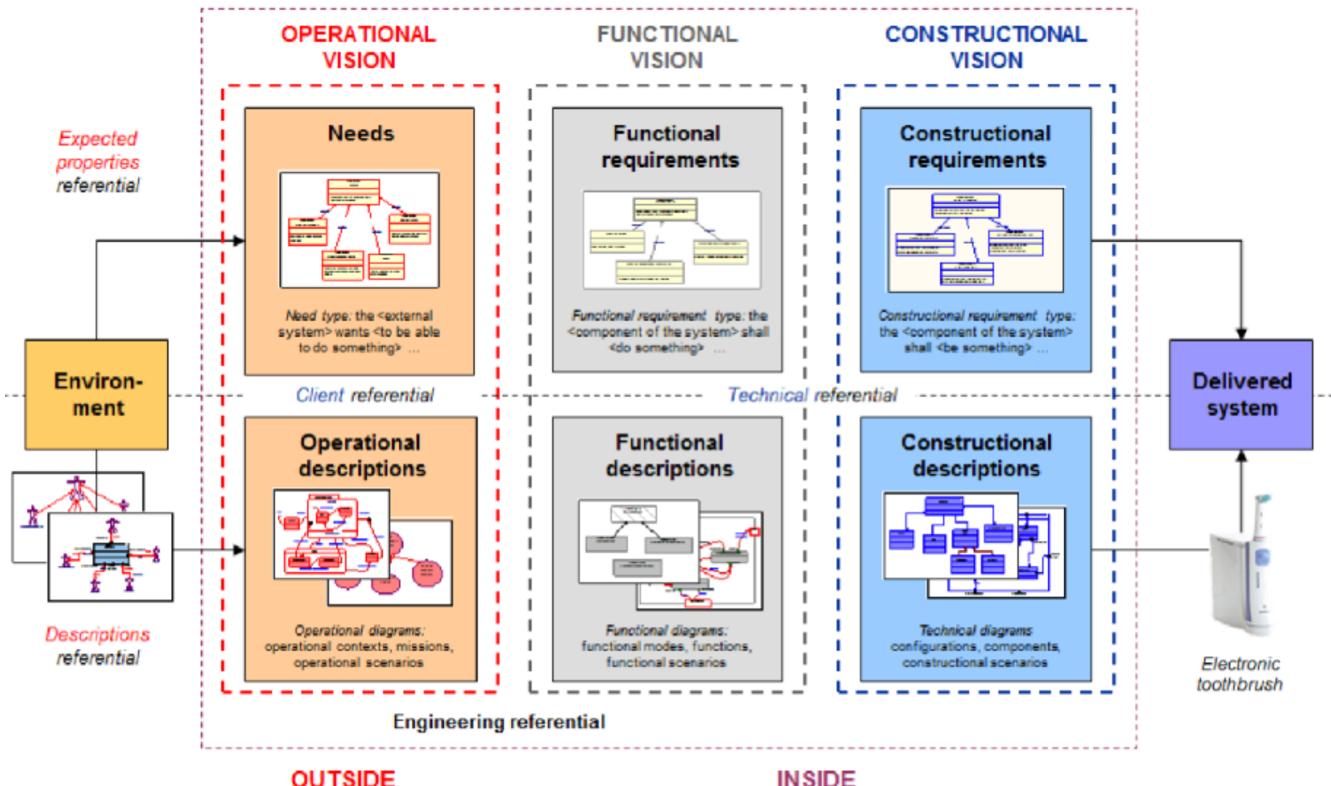


Figure 5. Expected properties vs. description (source CESAM)

## 2.4. Requirements in industry = ALM

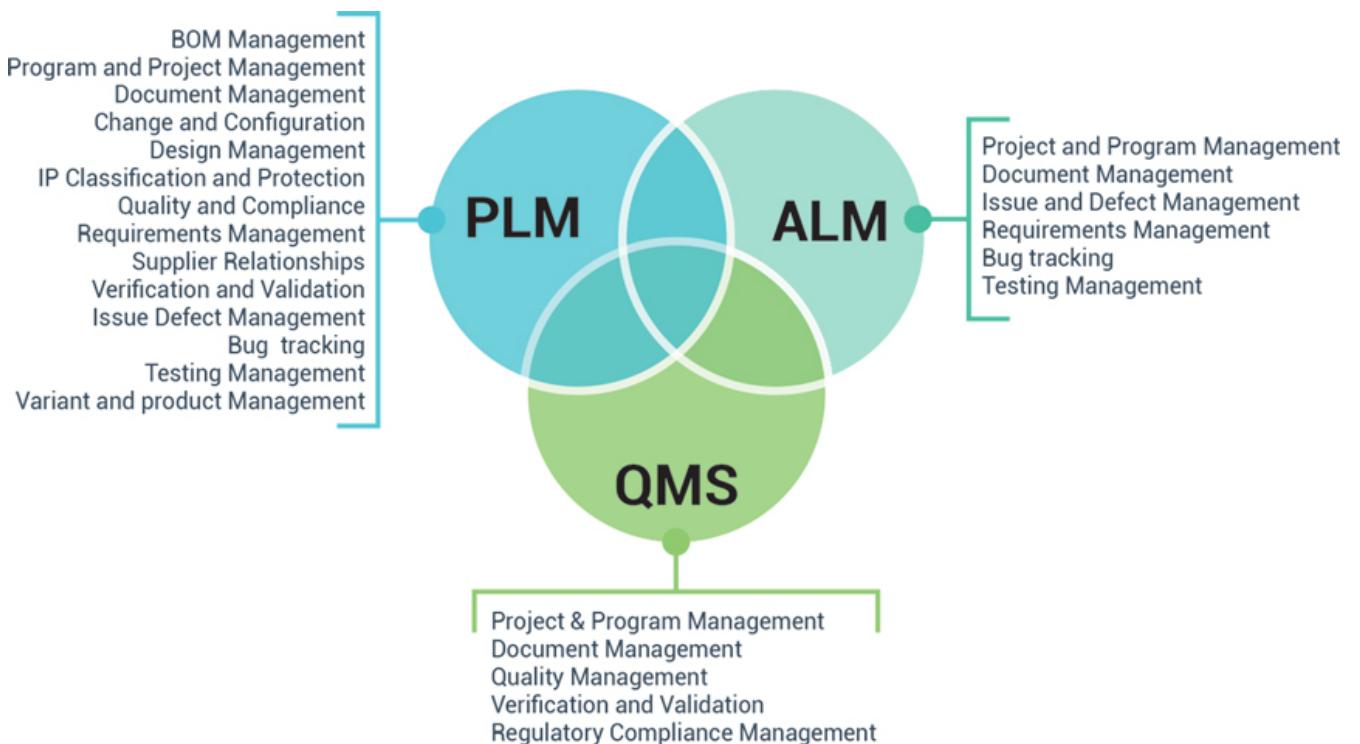


Figure 6. (source [here](#))

## 2.5. Problems...

- Readability?
- Compliance with the specs?
- Maintainability?
- Extension to fulfill the specs?
- Testability?

## 3. Situation overview

### 3.1. Typical situation

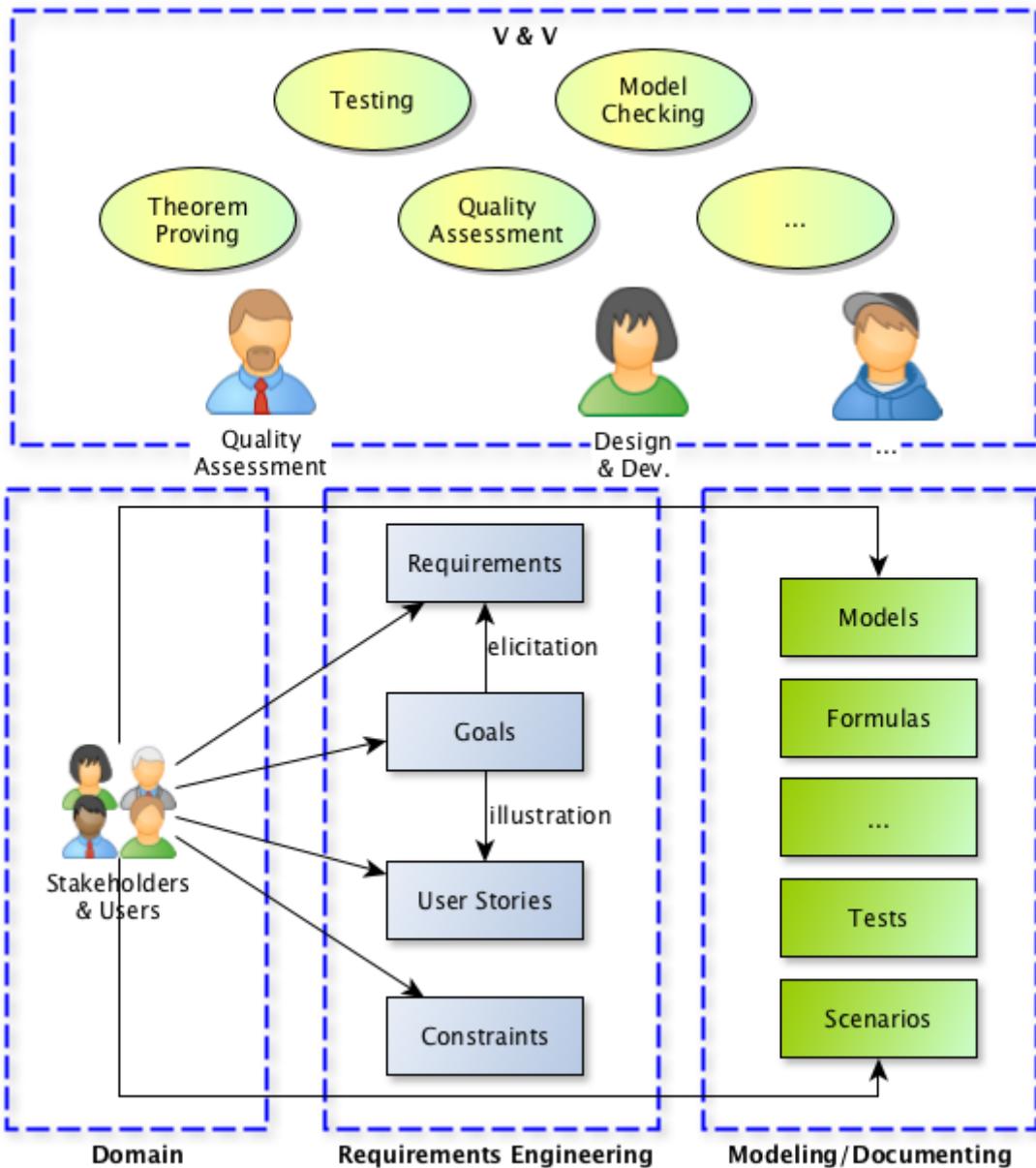


Figure 7. DevOps Quality Assessment

### 3.2. Who will use the product?

Persona = Name + Bio + Objectives

### 3.3. Writing Epics & Stories

Specifying in an agile way

### 3.4. Specifying with Stories

EPIC = Persona + Action + Benefits

Story = Epic + Acceptance criteria + Tests (+ Estimations)

# 4. Testing

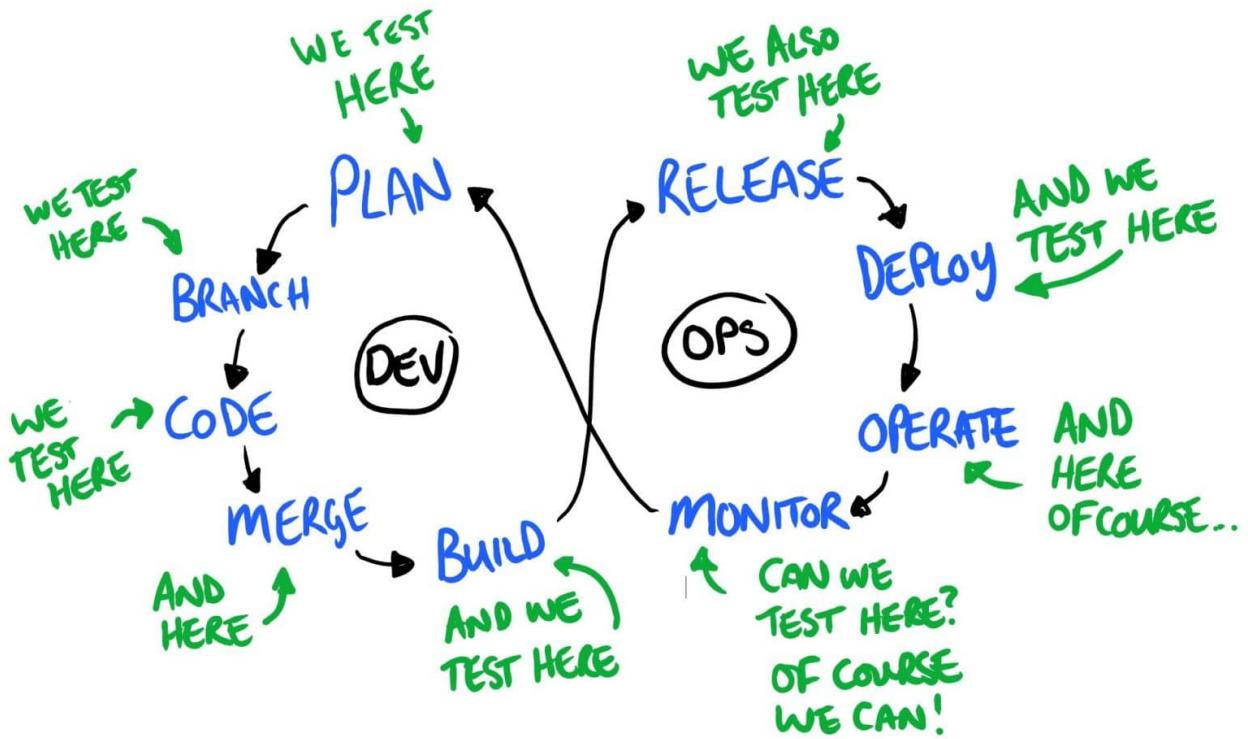


Figure 8. Tests in DevOps

## 4.1. Test-Driven Development

1. (Write an issue about the bug, with details)
2. Write a failing test (reproduce the bug)
3. Correct the bug
4. Make the test pass
5. (close the issue)

## 4.2. Behavior-Driven Development

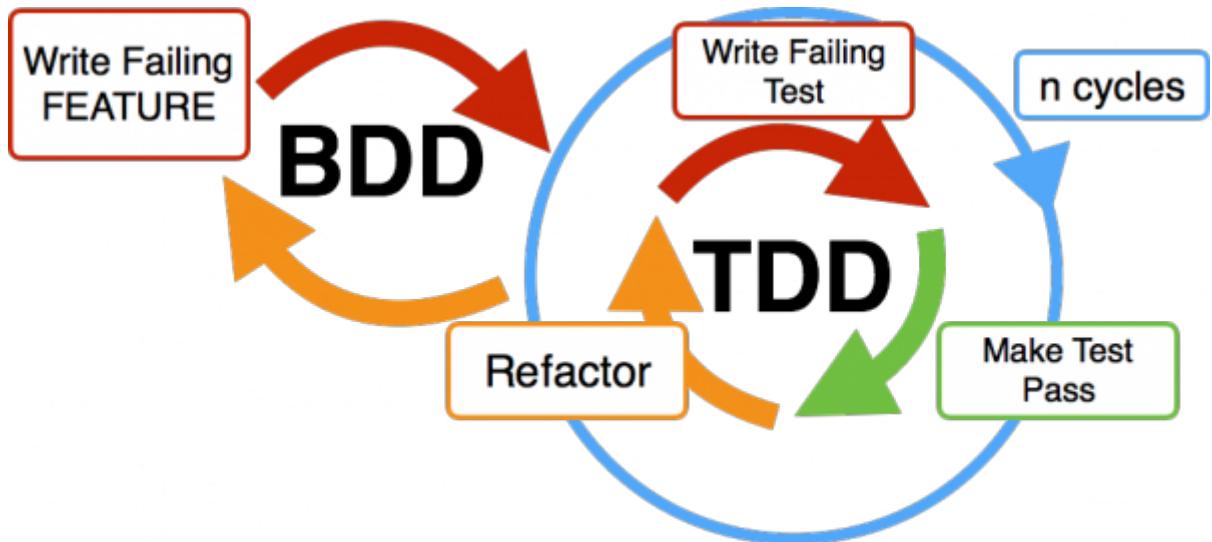
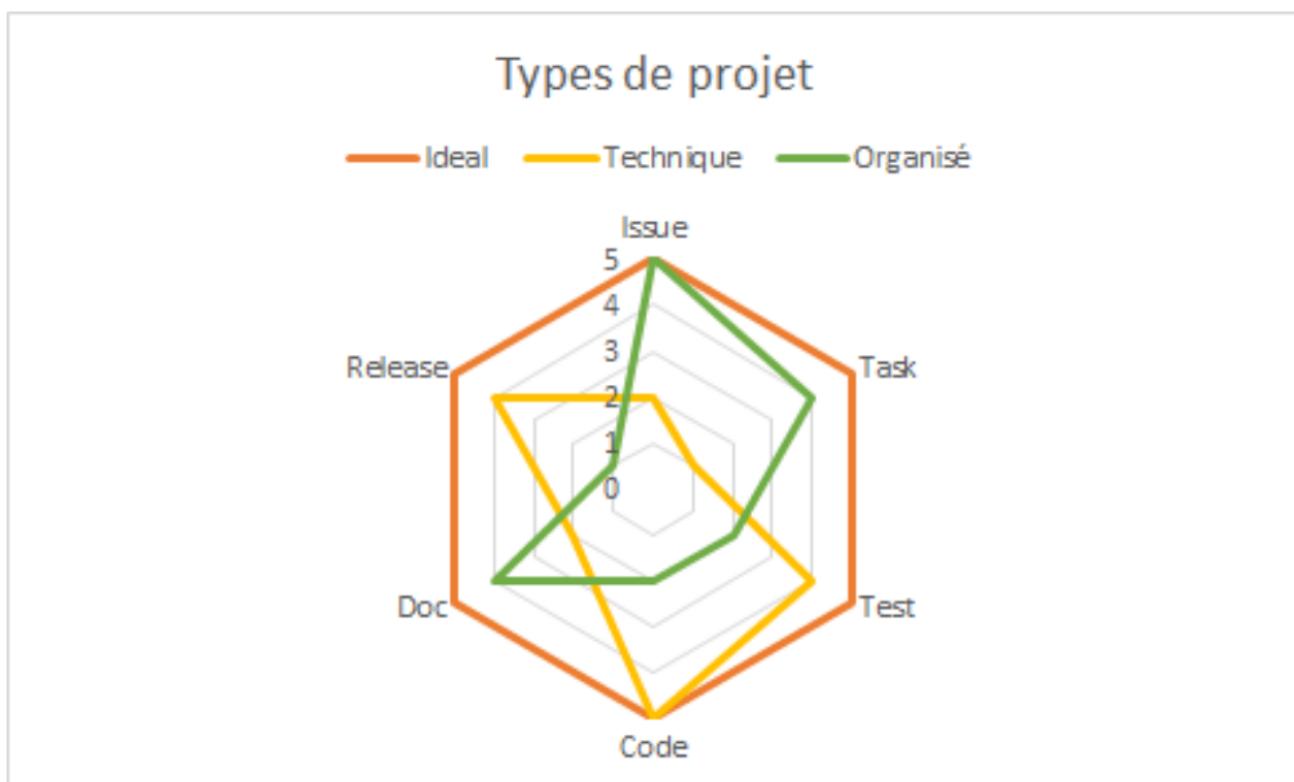


Figure 9. BDD vs TDD

### 4.3. Quality Assessment



### 4.4. Automation (and CI)

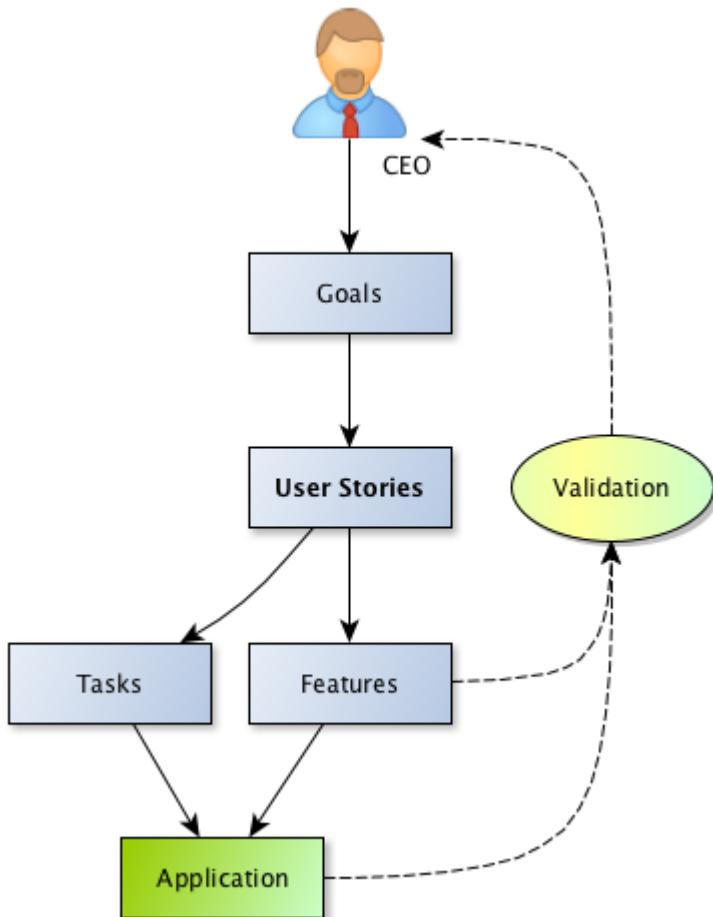
Running 0    Finished 327    All 327

List of finished builds from this project

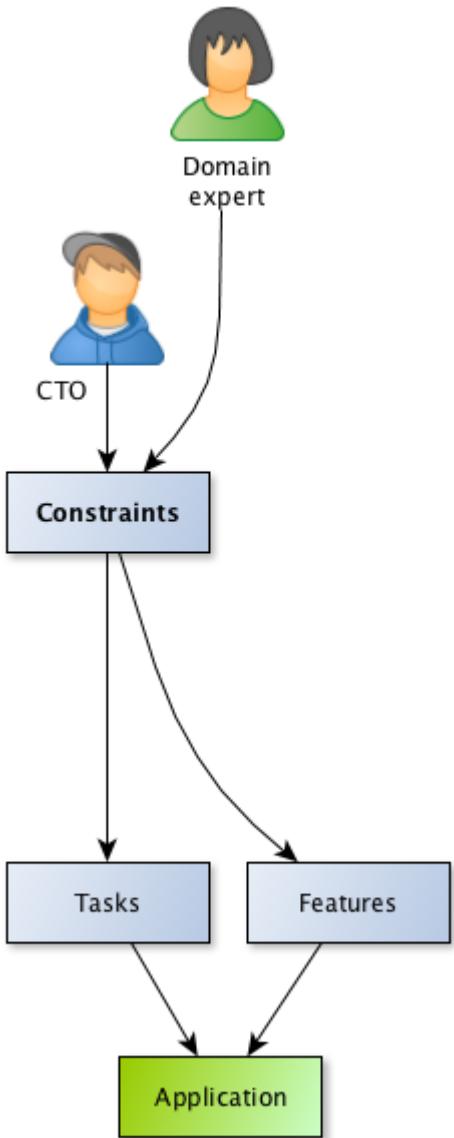
Status	Build ID	Commit	Ref	Runner	Name	Duration	Finished at
✓ success	Build #351965	23b89d99	artifacts	golang-cross#1059	Bleeding Edge	6 minutes 4 seconds	about 19 hours ago
✓ success	Build #351548	634b6f5e	artifacts	golang-cross#1059	Bleeding Edge	5 minutes 43 seconds	about 22 hours ago
✓ success	Build #349948	56329a8e	artifacts	golang-cross#1059	Bleeding Edge	6 minutes 2 seconds	1 day ago
✓ success	Build #349883	c01876c1	master	golang-cross#1059	Bleeding Edge	5 minutes 39 seconds	1 day ago
✗ failed	Build #349807	623f3f5a	master	golang-cross#1059	Bleeding Edge	1 minute 50 seconds	1 day ago
✗ failed	Build #349804	338d0a8b	artifacts	golang-cross#1059	Bleeding Edge	1 minute 35 seconds	1 day ago

## 5. Who are your clients?

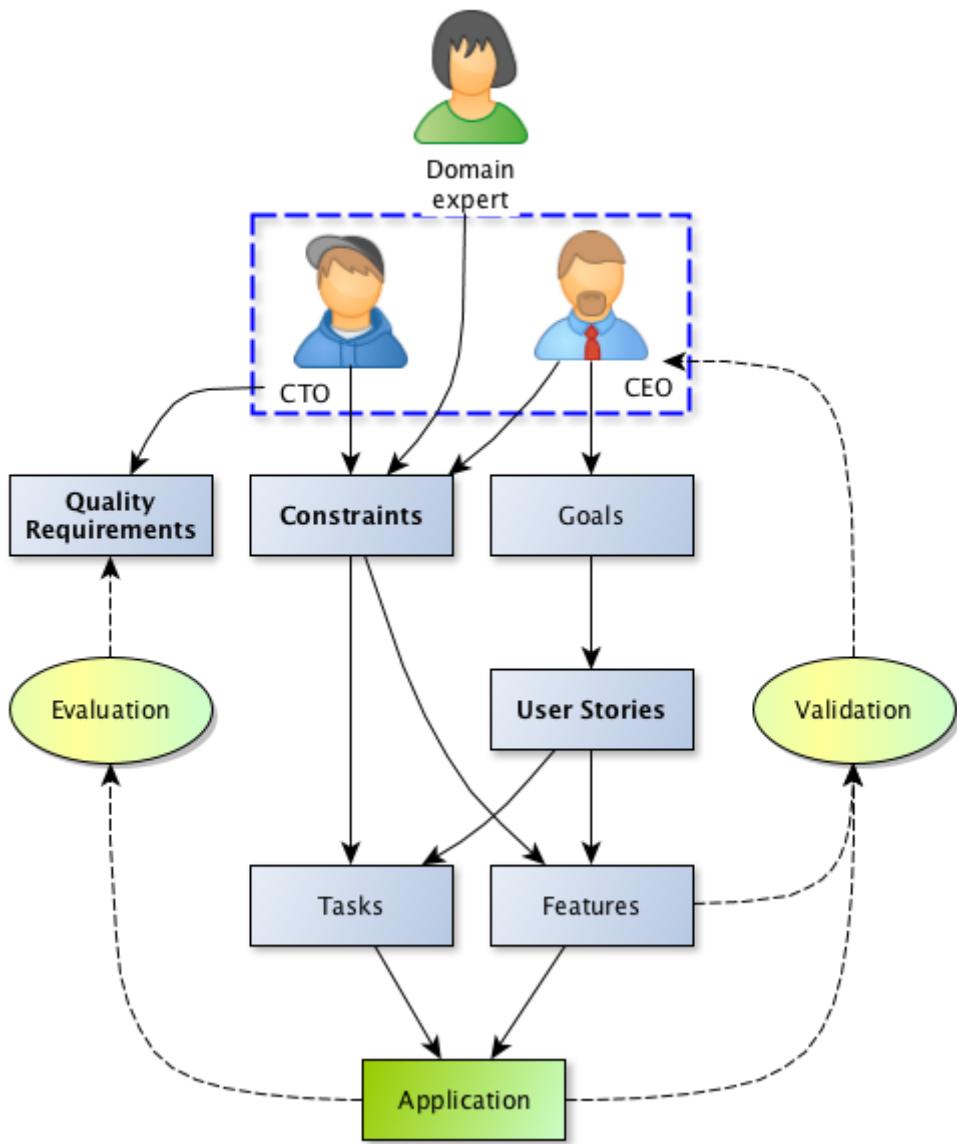
### 5.1. Your client(s)



### 5.2. Your teacher(s)

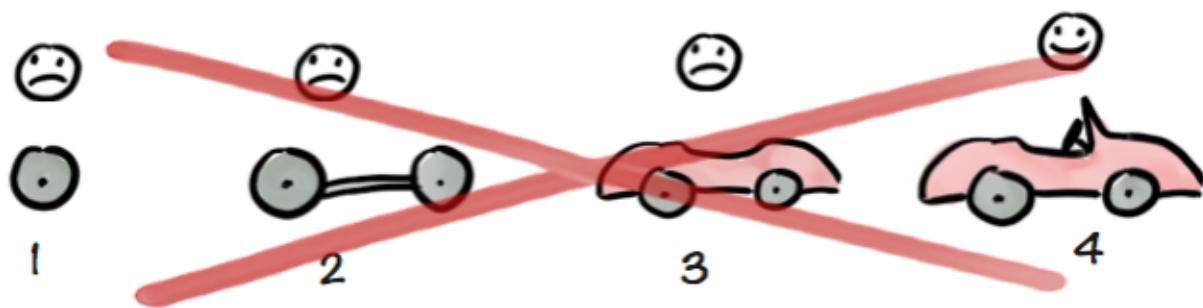


### 5.3. At the same time!



## 5.4. Minimal Viable Product (usual)

**Not like this....**



**Like this!**

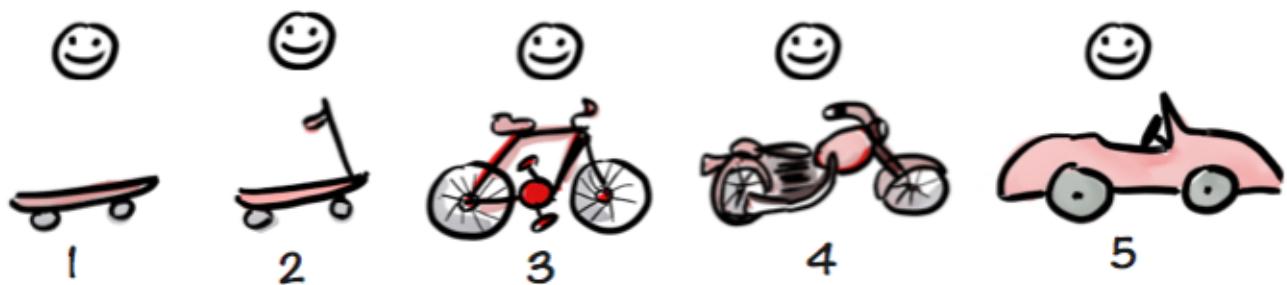


Figure 10. Minimal Viable Product (by Henrik Kniberg)

## 5.5. Minimal Viable Product (improved)

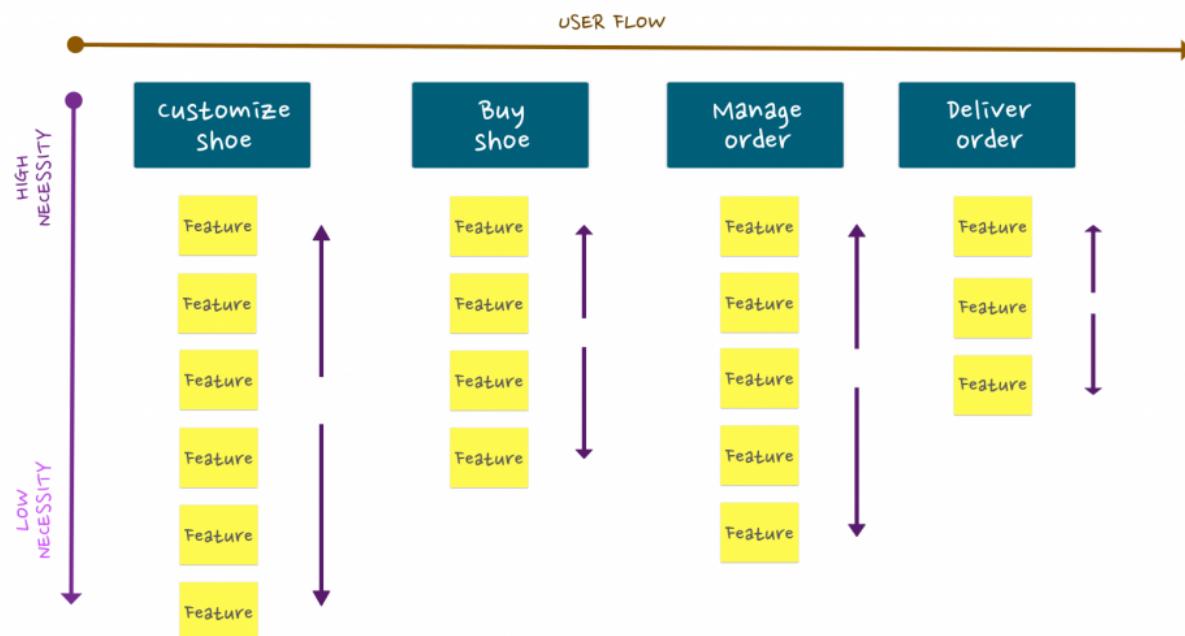
[minimum viable product] | *minimum-viable-product.png*

Figure 11. A more accurate representation (source <https://alkomsoftware.pl/en/blog/mvp-insurance/>)

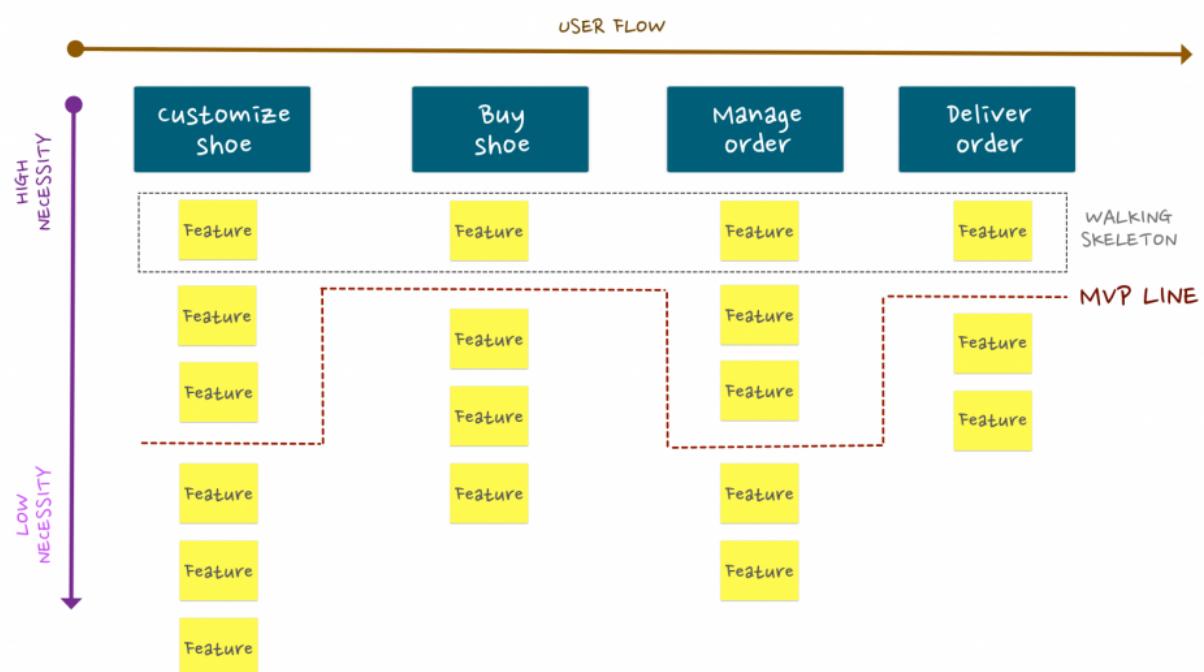
## 5.6. MVP & EPICS



PRIMARY GOAL: Allow users to receive an individual, customised pair of shoes



PRIMARY GOAL: Allow users to receive an individual, customised pair of shoes



## 6. Back to the requirements

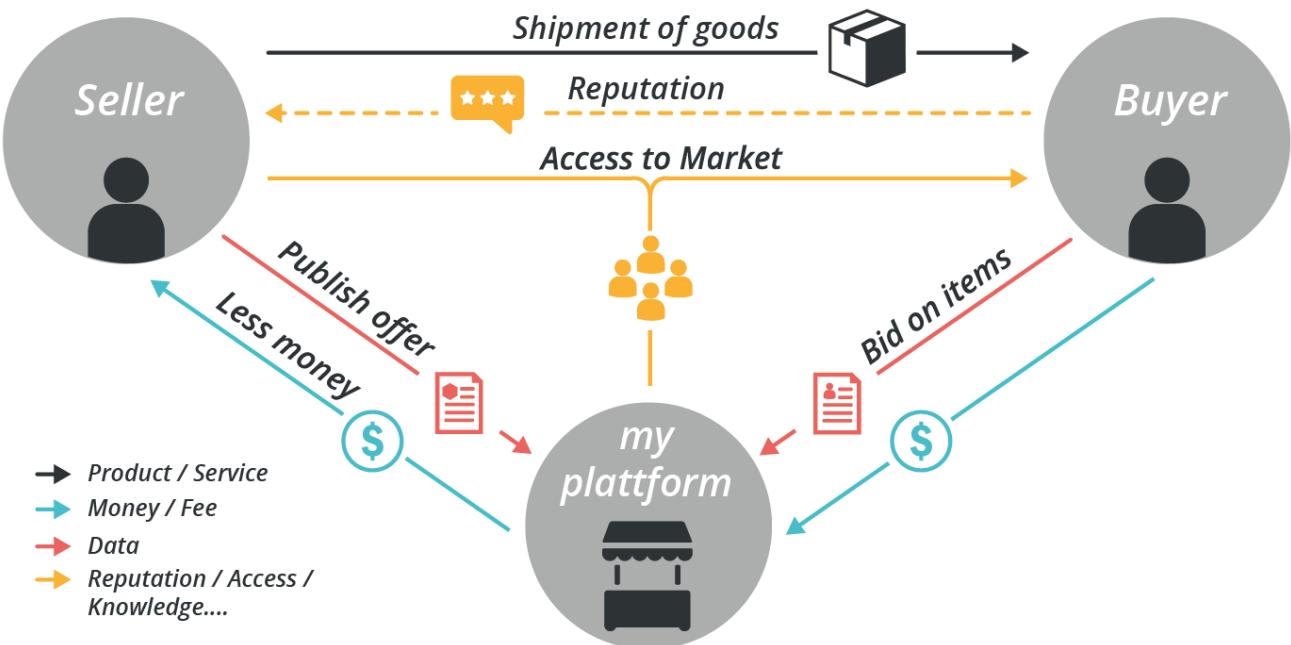
MOA

Req = Stakeholders Needs

MOE

Req = Systems Needs

## 7. Stakeholders Value Networks



## 8. Traceability

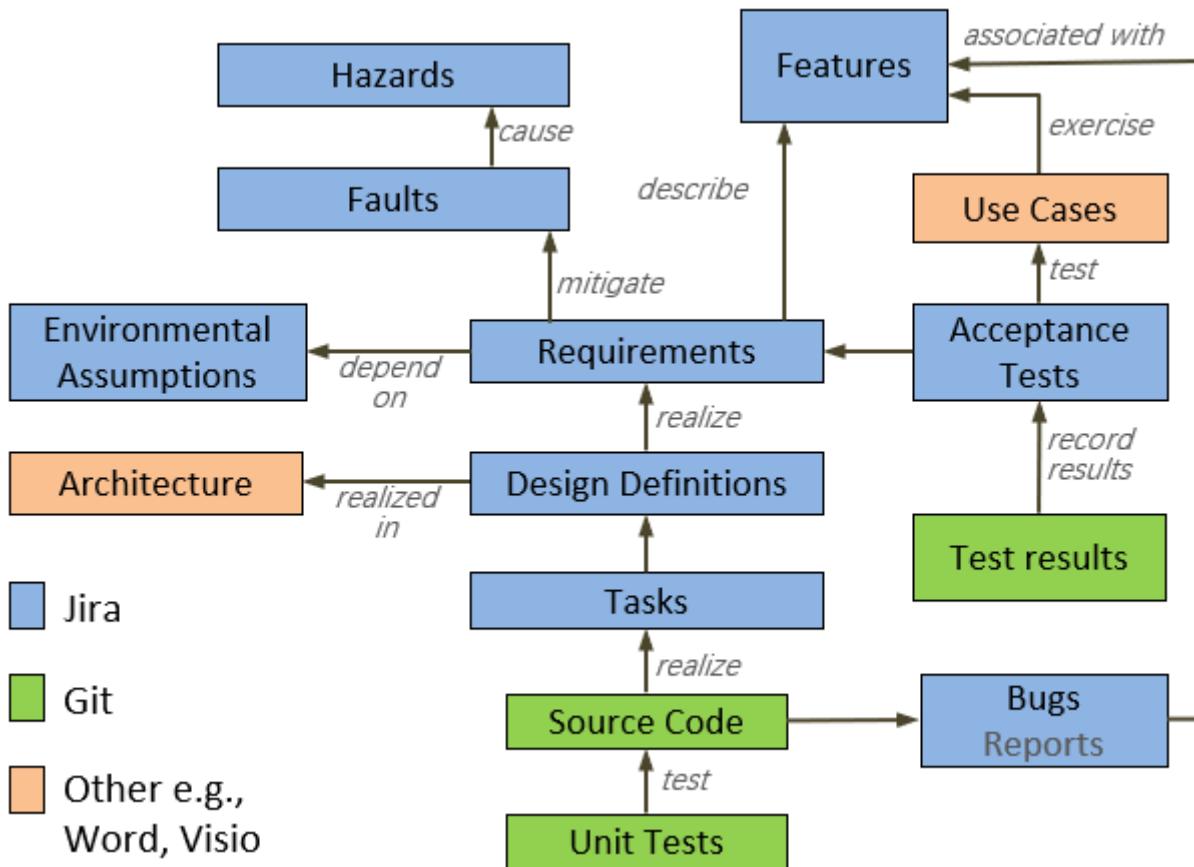


Figure 12. Example of traceability in the Dronology project (source [here](#))

# Appendix A: Useful links

## Gail Murphy's keynote at RE'2018

<https://www2.slideshare.net/murphygc/beyond-devops-finding-value-through-requirements>

# Appendix B: Credits

Images taken from:

- <https://abstracta.us/blog/devops/testing-driver-devops-culture/>
- <http://meshfields.de/continuous-integration-testing-delivery-ionic2-hybrid-mobile-apps-buddybuild/>
- <https://altkomsoftware.pl/en/blog/mvp-insurance/>
- <http://www.cesames.net/wp-content/uploads/2017/05/CESAM-guide.pdf>