Big Picture

Table of Contents

1. DevOps / Agile / Test / Requirements

- Plan, Test, OK
- But towards WHAT?

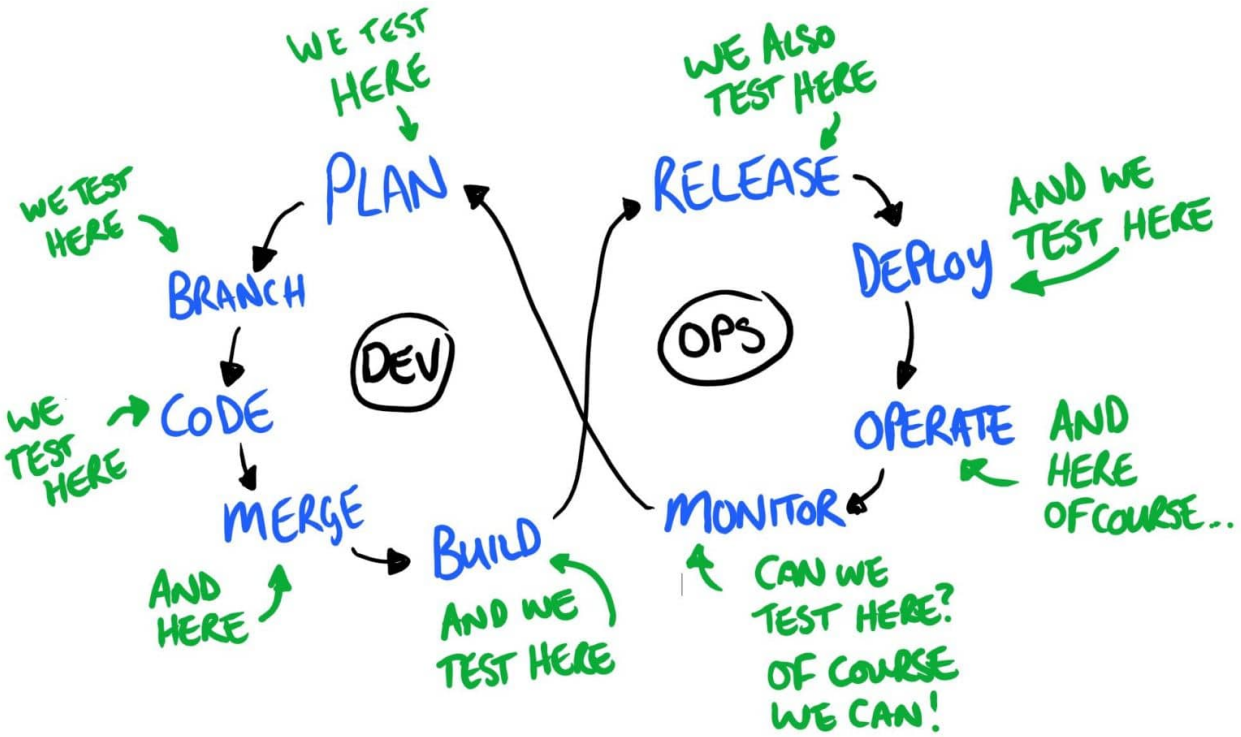


Figure 1. Tests in DevOps

1.1. 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

1.2. Requirements Engineering Process

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

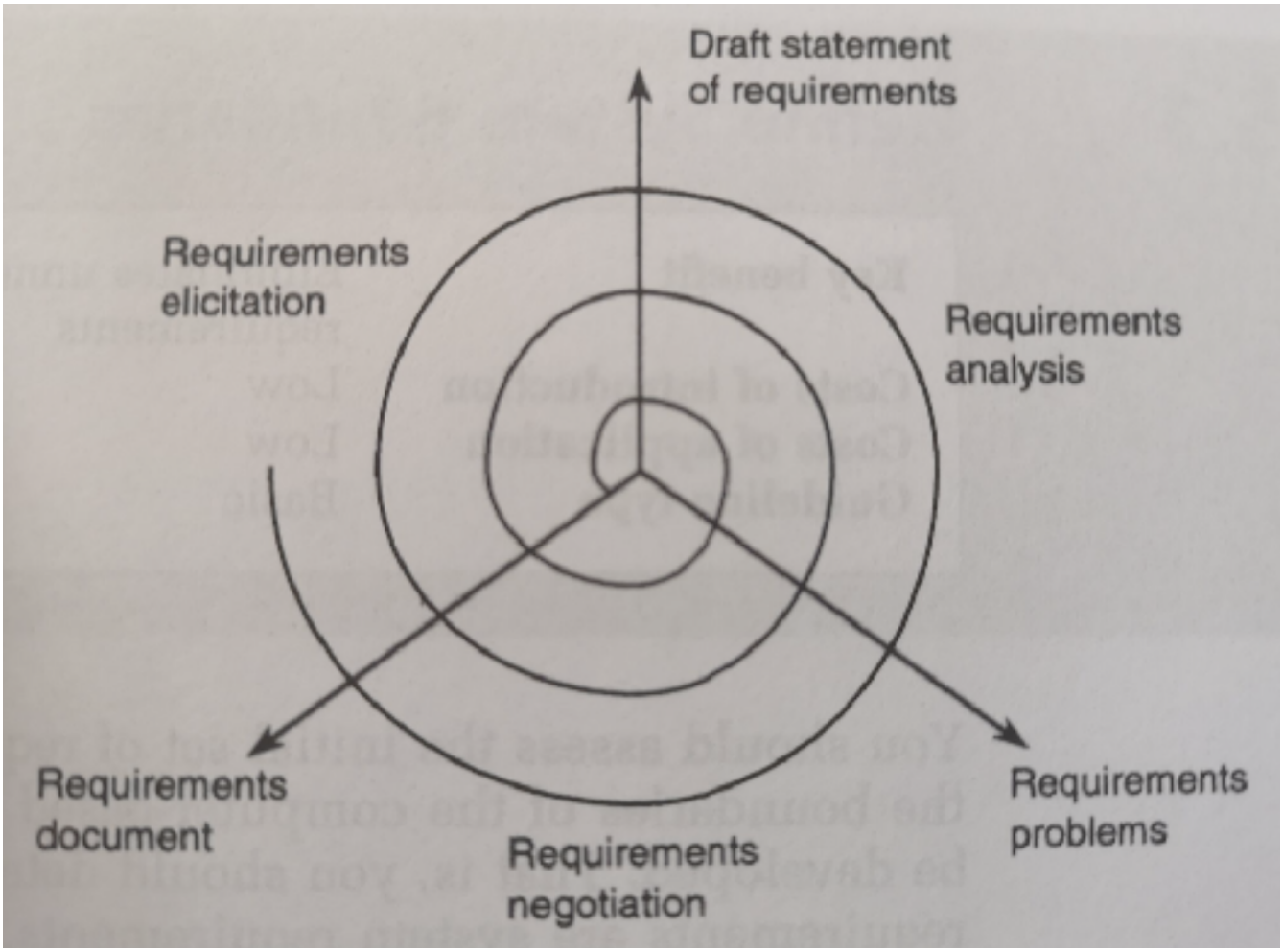


Figure 2. Requirements Engineering Process (source [Sommerville & Sawyer 1997])

Expected properties vs. description

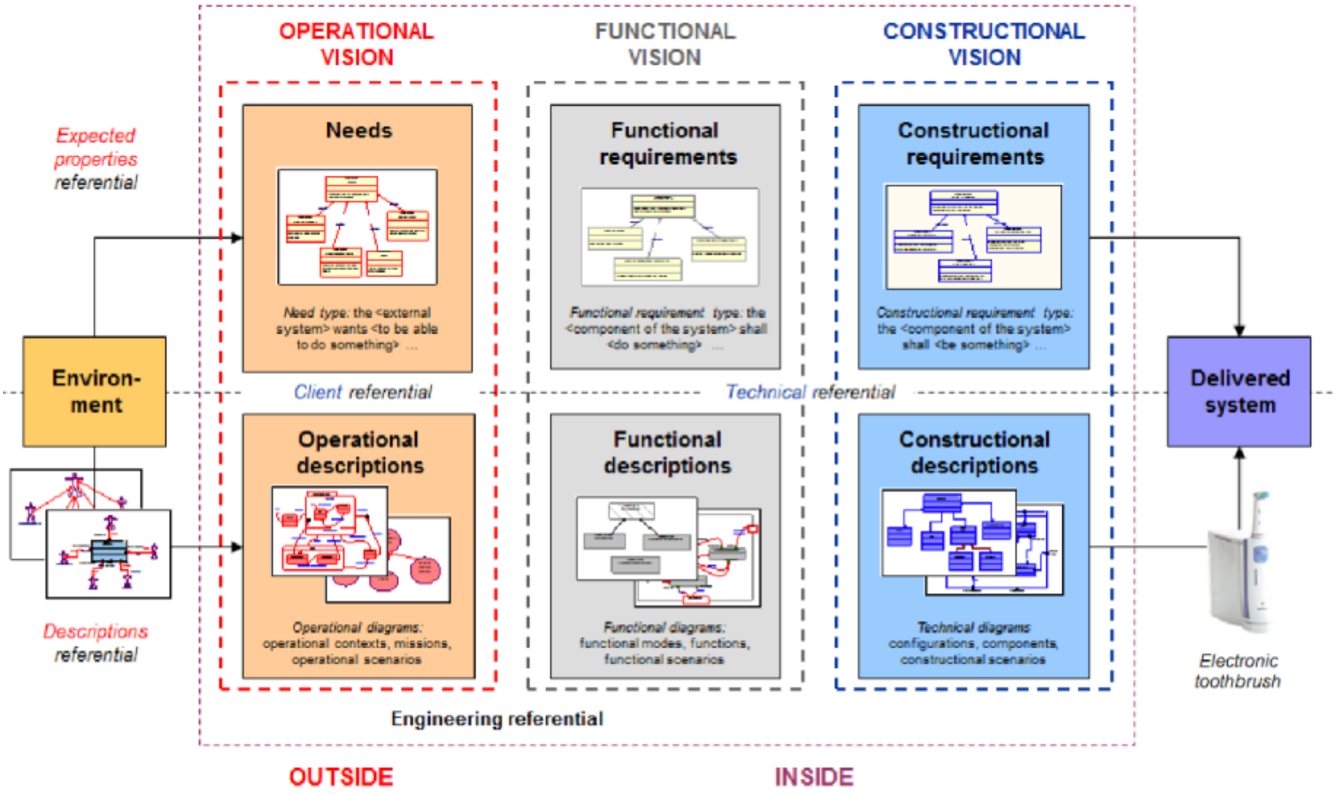


Figure 3. Expected properties vs. description (source CESAM)

1.3. Requirements in industry = ALM

(source here) image::alm.jpg[ALM, link="https://visuresolutions.com/application-lifecycle-management-alm-tool"]

1.4. Problems…

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

2. Situation overview

2.1. Typical situation

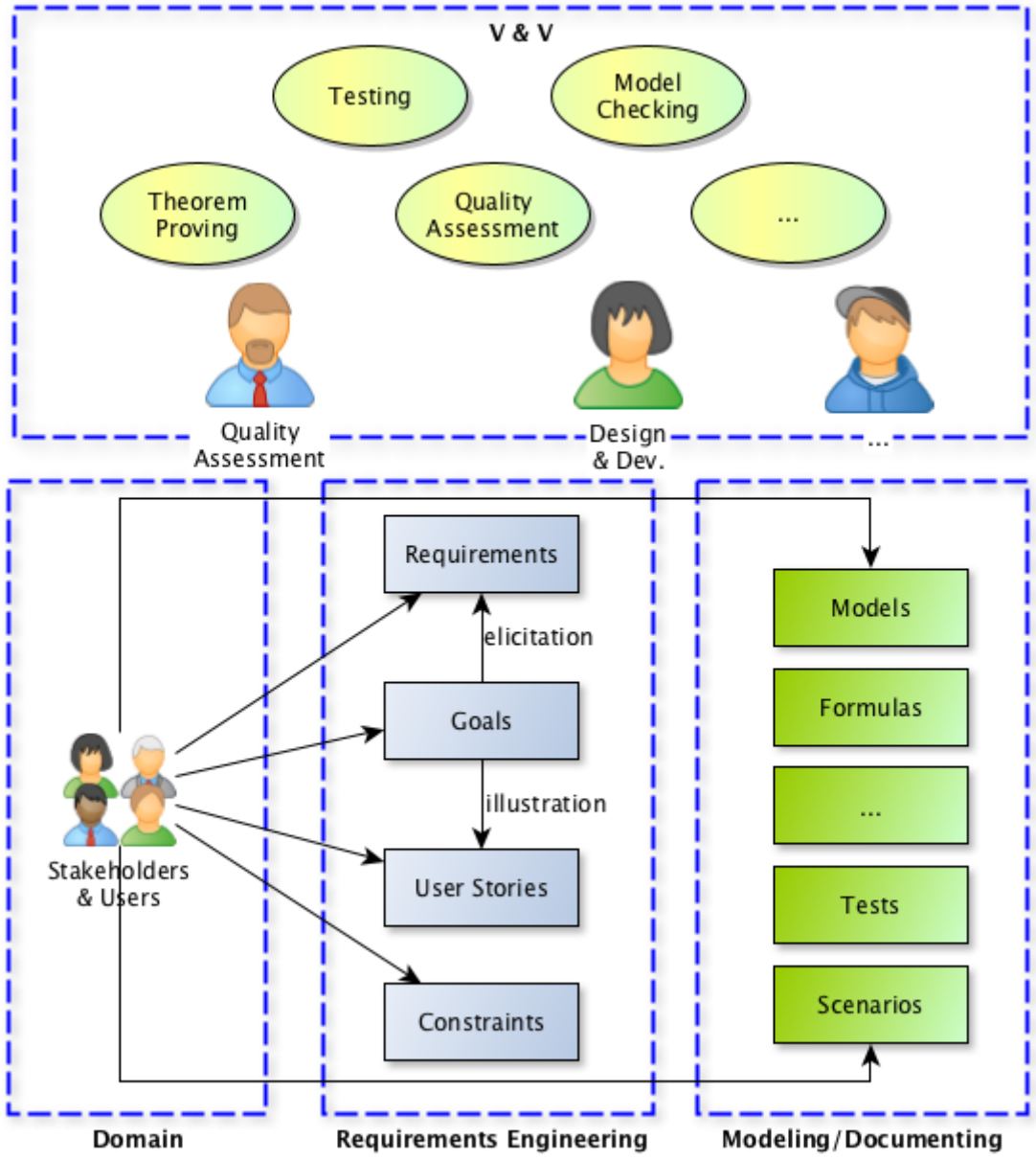


Figure 4. DevOps Quality Assessment

2.2. Who will use the product?

Persona = Name + Bio + Objectives

2.3. Writing Epics & Stories

Specifying in an agile way

2.4. Specifying with Stories

EPIC = Persona + Action + Benefits

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

3. Testing

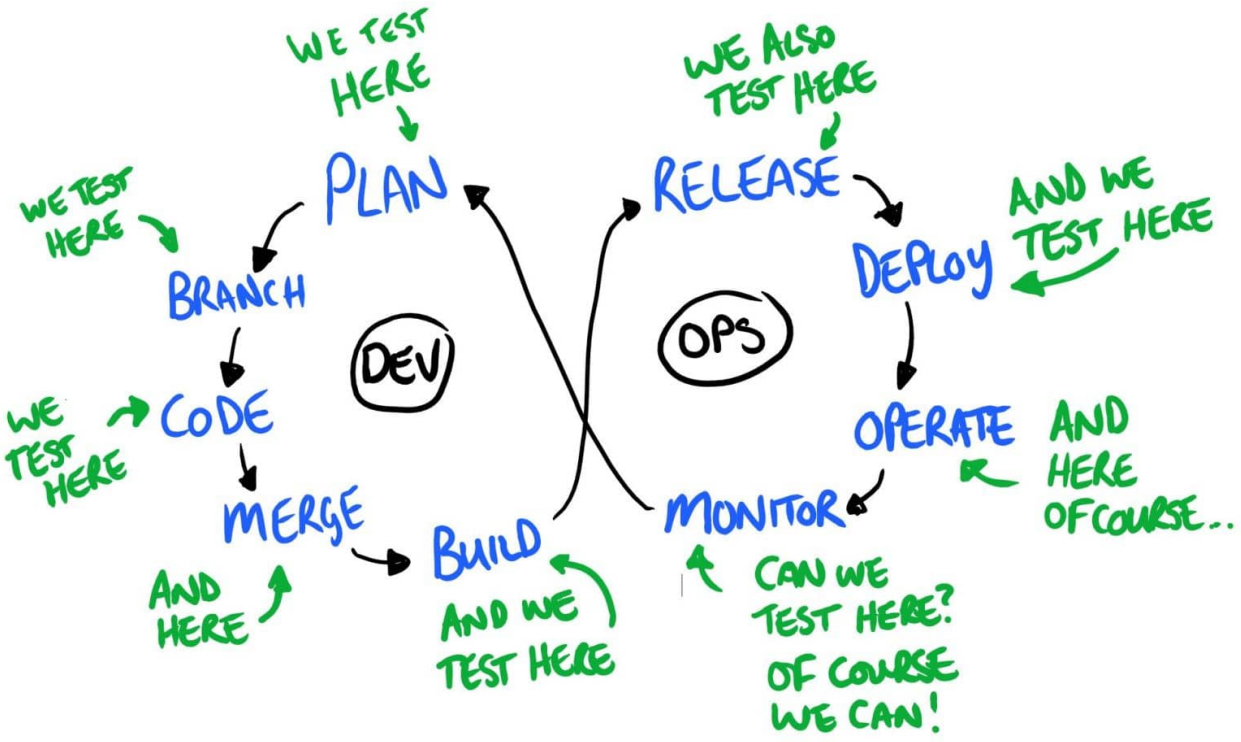


Figure 5. Tests in DevOps

3.1. 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)

3.2. Behavior-Driven Development

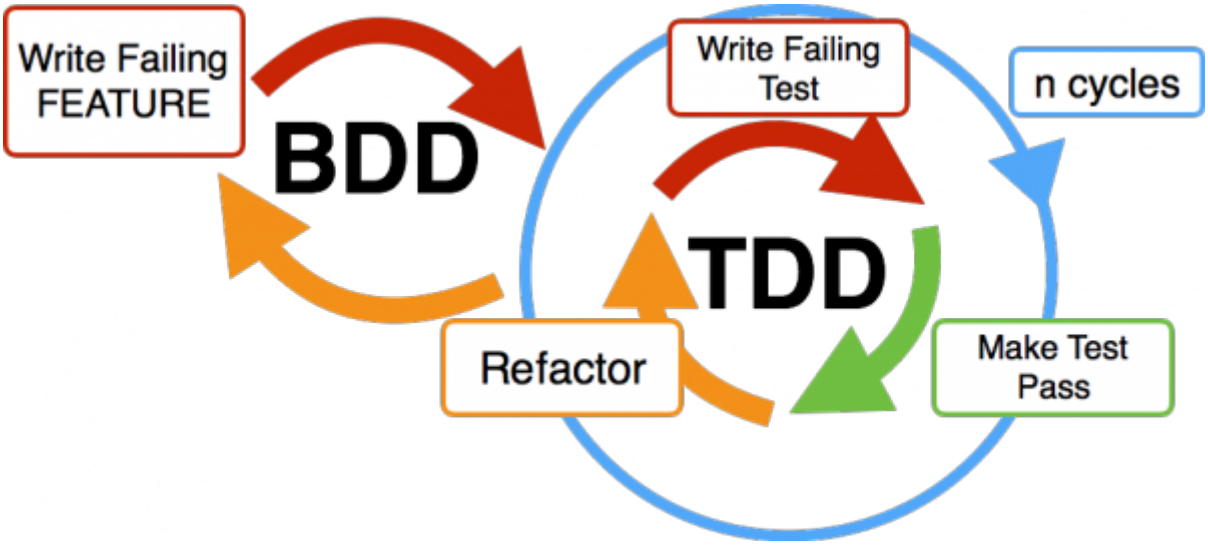
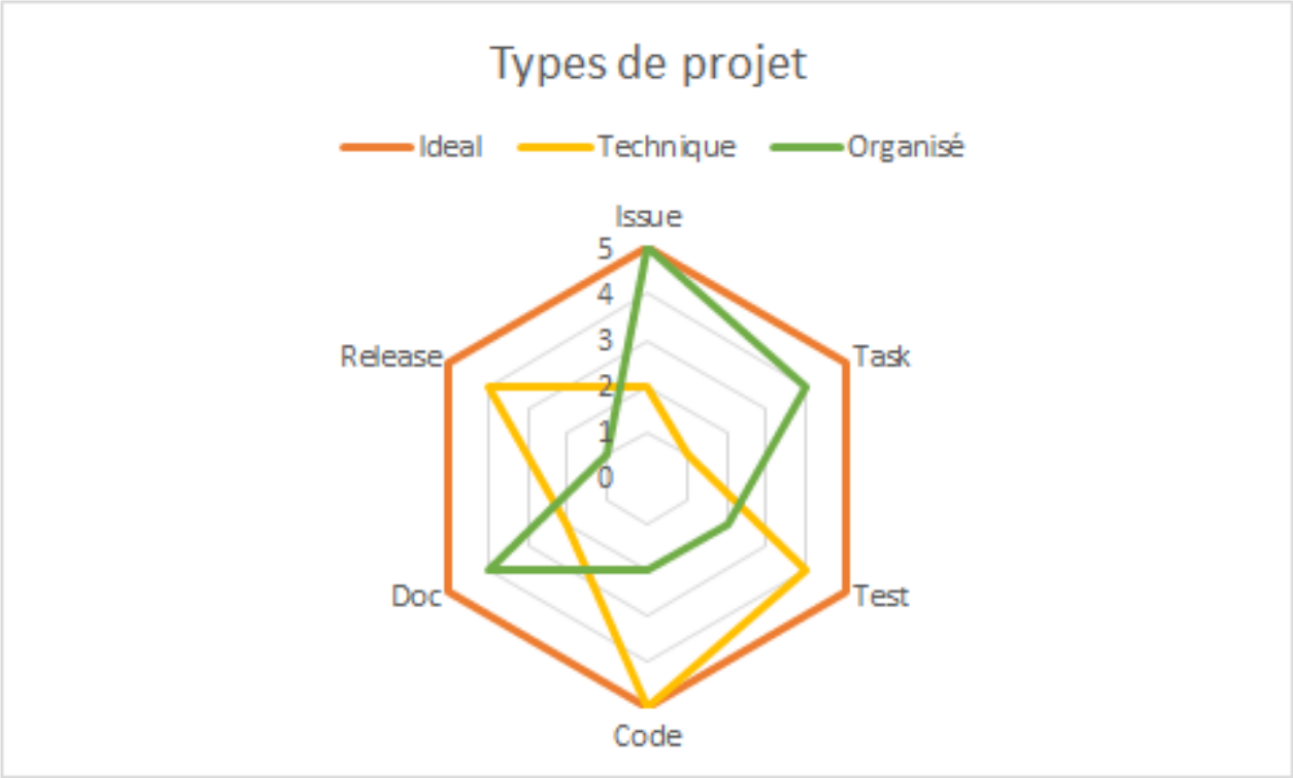


Figure 6. BDD vs TDD

3.3. Quality Assesment



3.4. Automation (and CI)

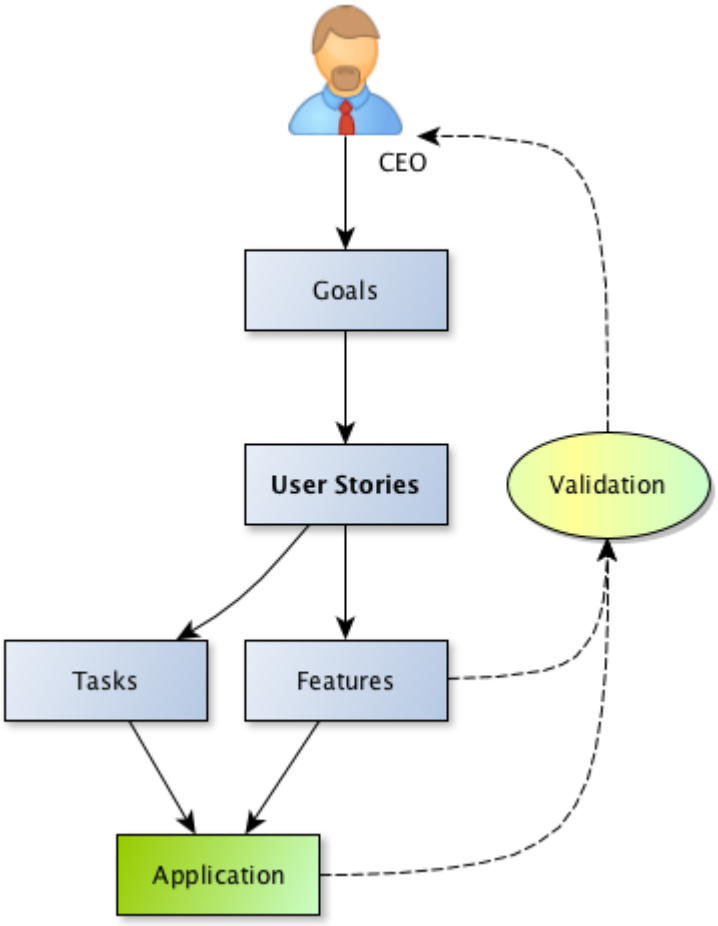
Running 0 Finished 327 All 327

List of finished builds from this project

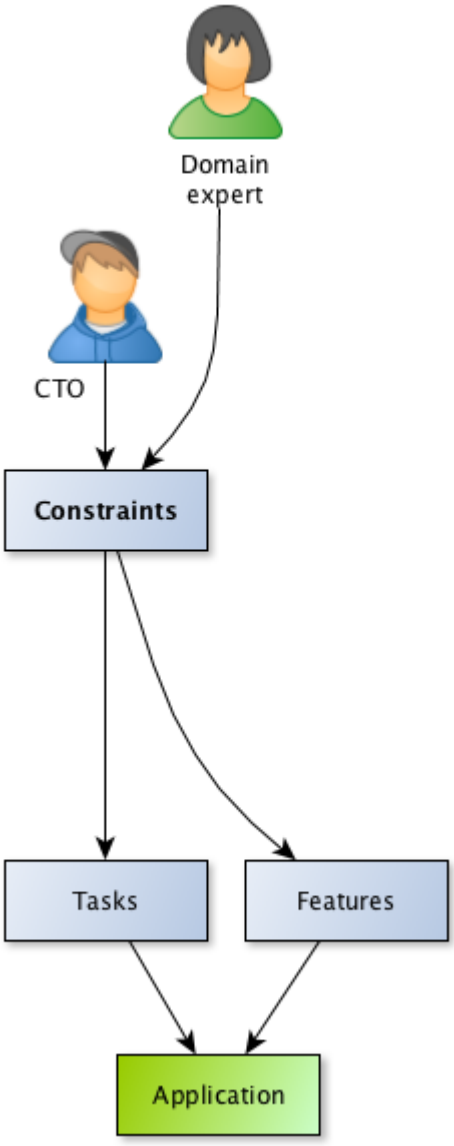
Status	Build ID	Commit	Ref	Runner	Name	Duration	Finished at
success	**Build #351965**	23b89d99	artifacts	golang-cross#1059	Bleeding Edge	6 minutes 4 seconds	about 19 hours ago
success	**Build #351548**	634b6f5e	artifacts	golang-cross#1059	Bleeding Edge	5 minutes 43 seconds	about 22 hours ago
success	**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

4. Who are your clients?

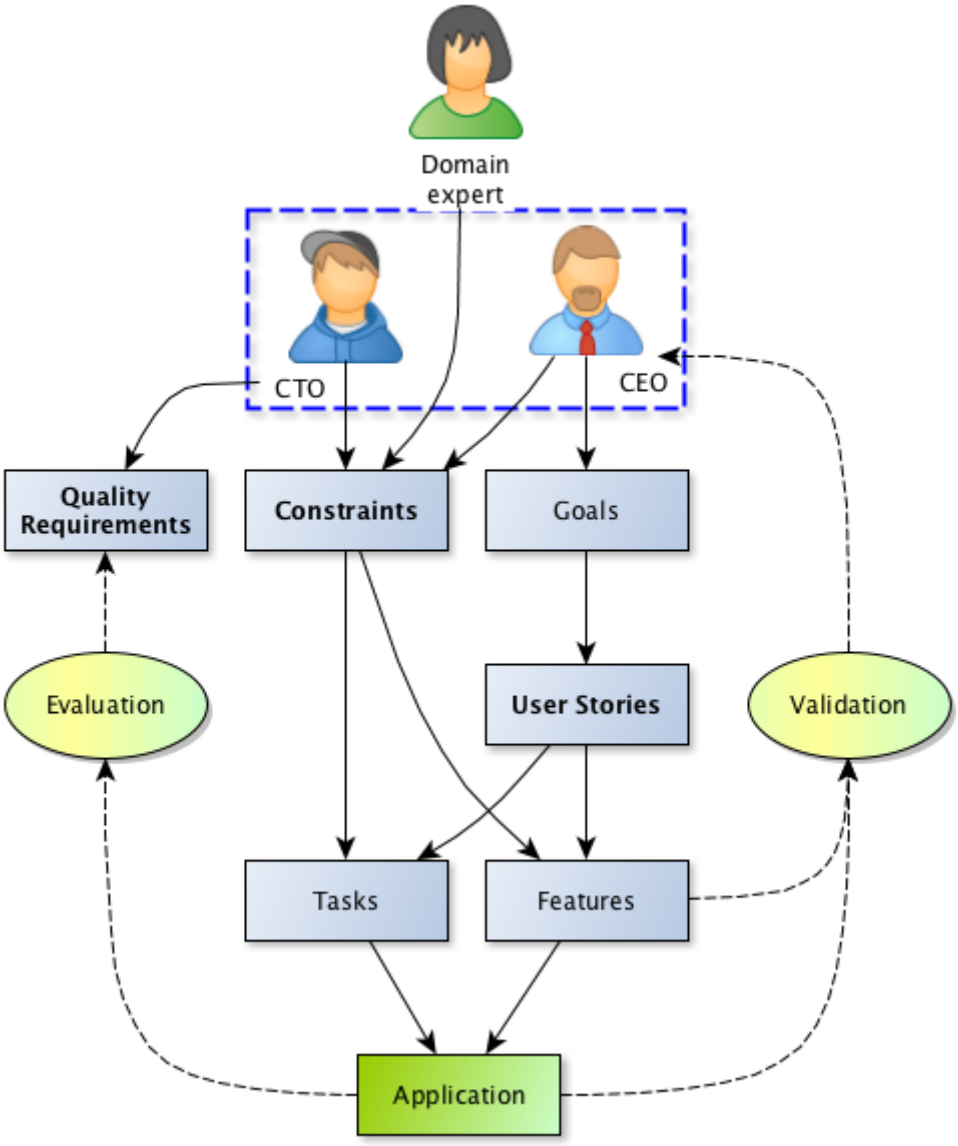
4.1. Your client(s)



4.2. Your teacher(s)



4.3. At the same time!



4.4. Minimal Viable Product (usual)

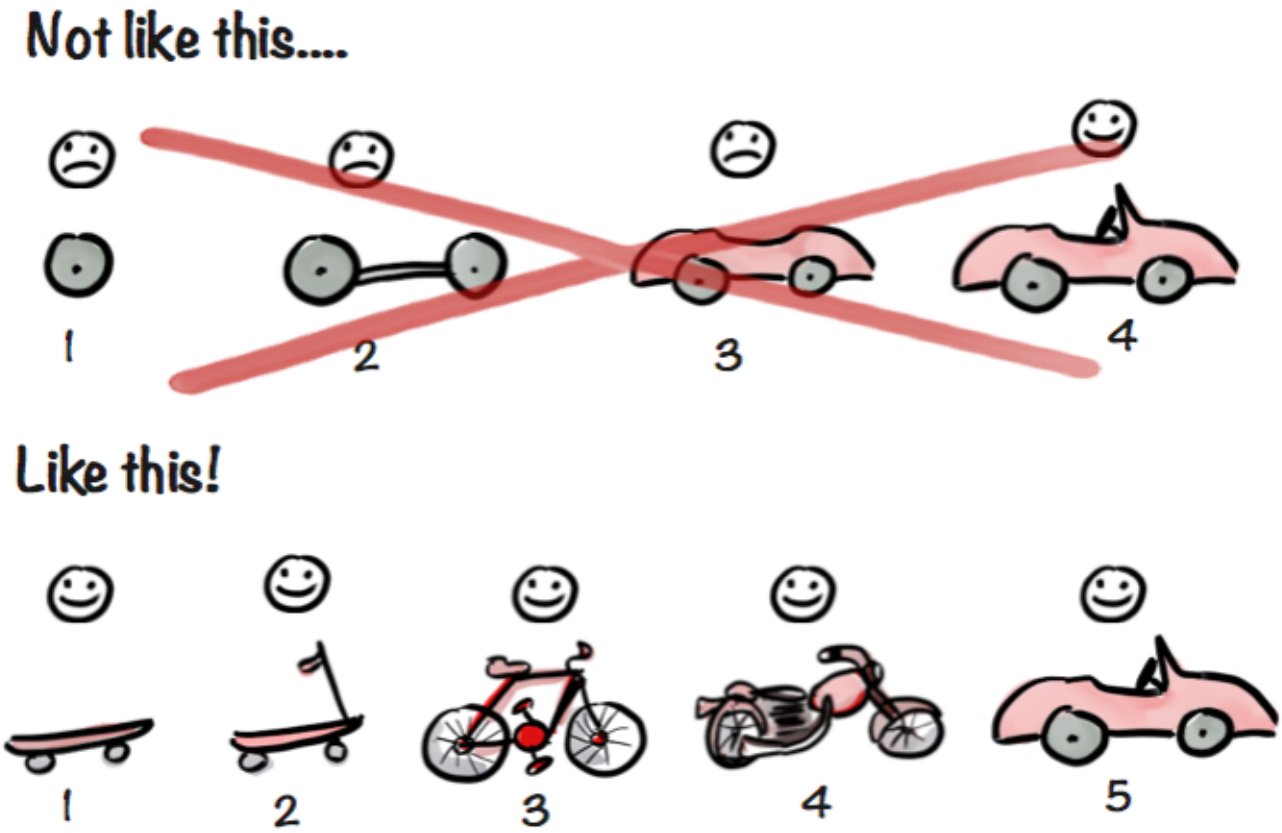


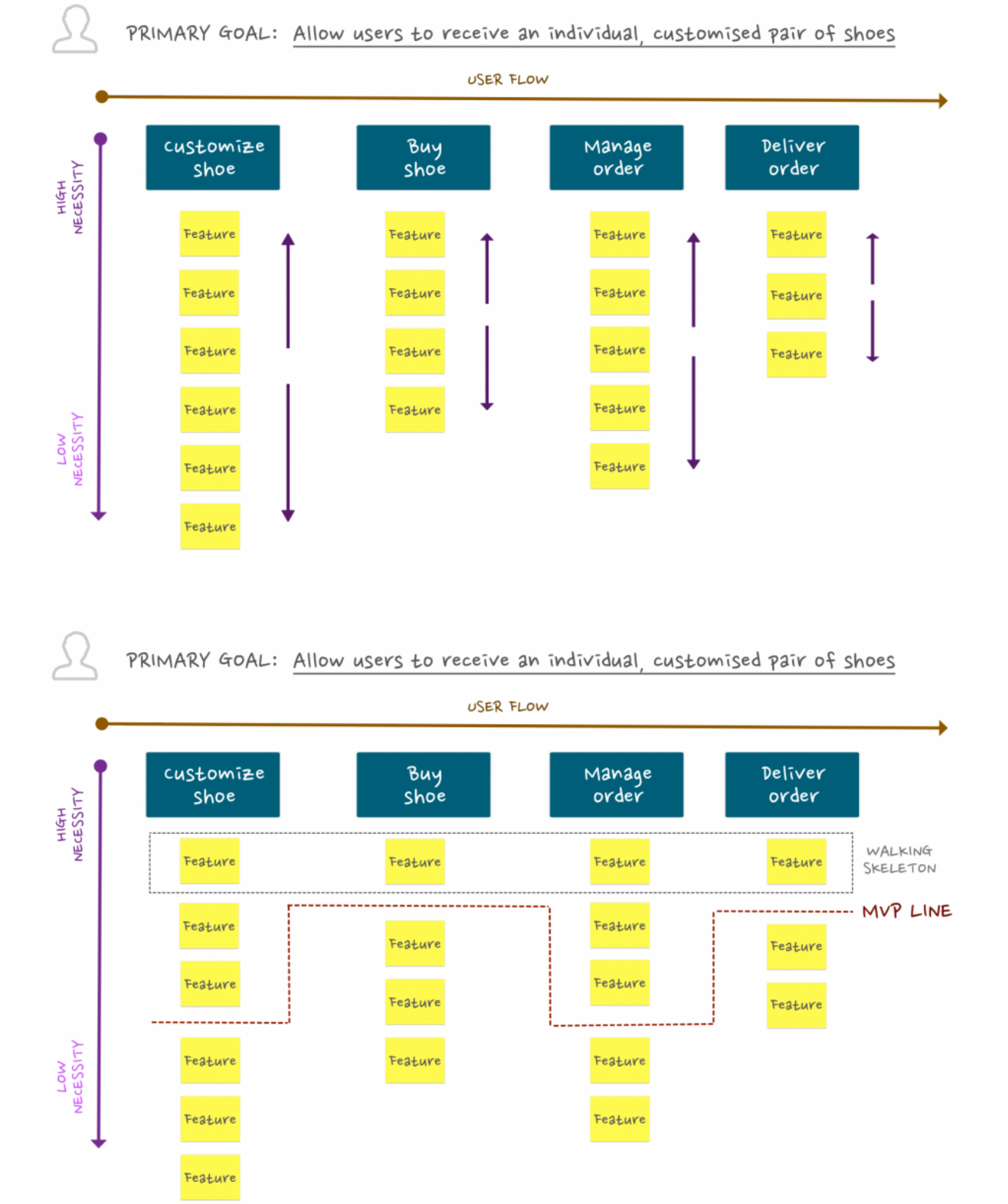
Figure 7. Minimal Viable Product (by Henrik Kniberg)

4.5. Minimal Viable Product (improved)

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

Figure 8. A more accurate representation (source https://altkomsoftware.pl/en/blog/mvp-insurance/)

4.6. MVP & EPICS



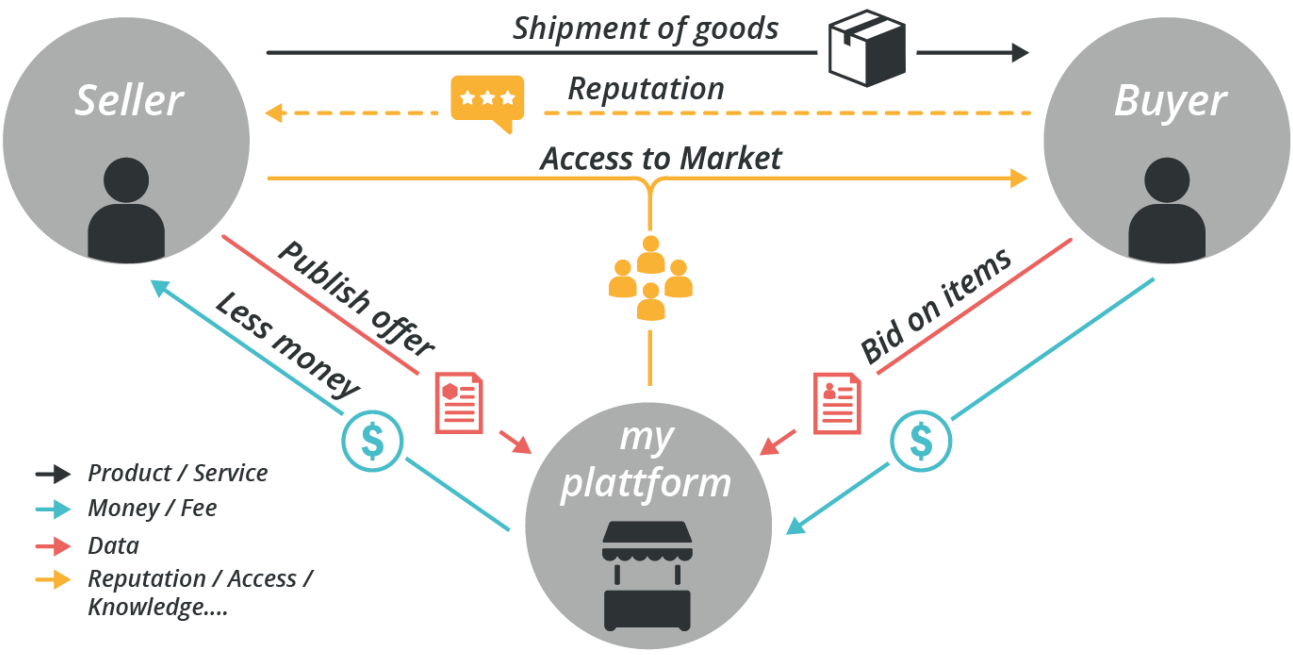
5. Back to the requirements

MOA

Req = *Stakeholders Needs*

MOE

6. Stakeholders Value Networks



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