



# Reconciling Requirements and Continuous Integration in an Agile Context

pictures: [sxc.hu](#), [pixels](#)

20.08.18, RE'18, Banff, Canada  
Jean-Michel Bruel & Sébastien Mosser



# Outline

---

- **Opening session**
- 1. Identifying Personae
- 2. Writing Epics & Stories
- 3. Implementing & Testing stories
- **BREAK**
- 4. Automating Acceptance Scenarios
- 5. Deploying a Cont. Int. stack
- 6. From CI to Cont. Deployment
- **Closing session**



# Legend

---



**Solution, Material, Example**



**Time to Practice!**

# Material & Resources



<http://bit.ly/RE18-T2>



<https://github.com/mosser/agile-tutorial.git>

# Technical Prerequisites

---



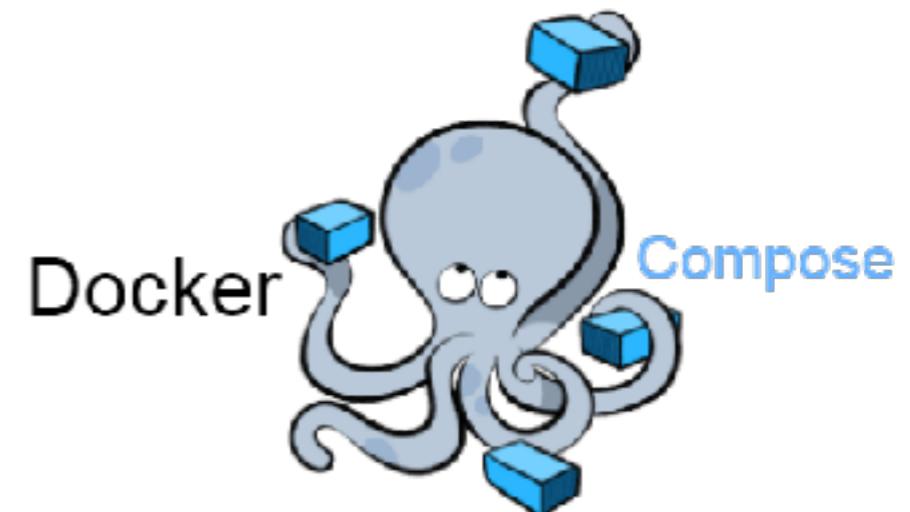
git



maven



<http://bit.ly/re18-t2-prereq>



# Set up your environment!



```
~$ git clone https://github.com/mosser/agile-tutorial.git
```

```
~$ cd agile-tutorial
```

```
~/agile-tutorial$ mvn clean package
```

```
~/agile-tutorial$ cd ci
```

```
~/agile-tutorial/ci $ docker network create re18-network
```

```
~/agile-tutorial/ci$ docker-compose up -d
```

```
~/agile-tutorial/ci$ docker-compose down
```

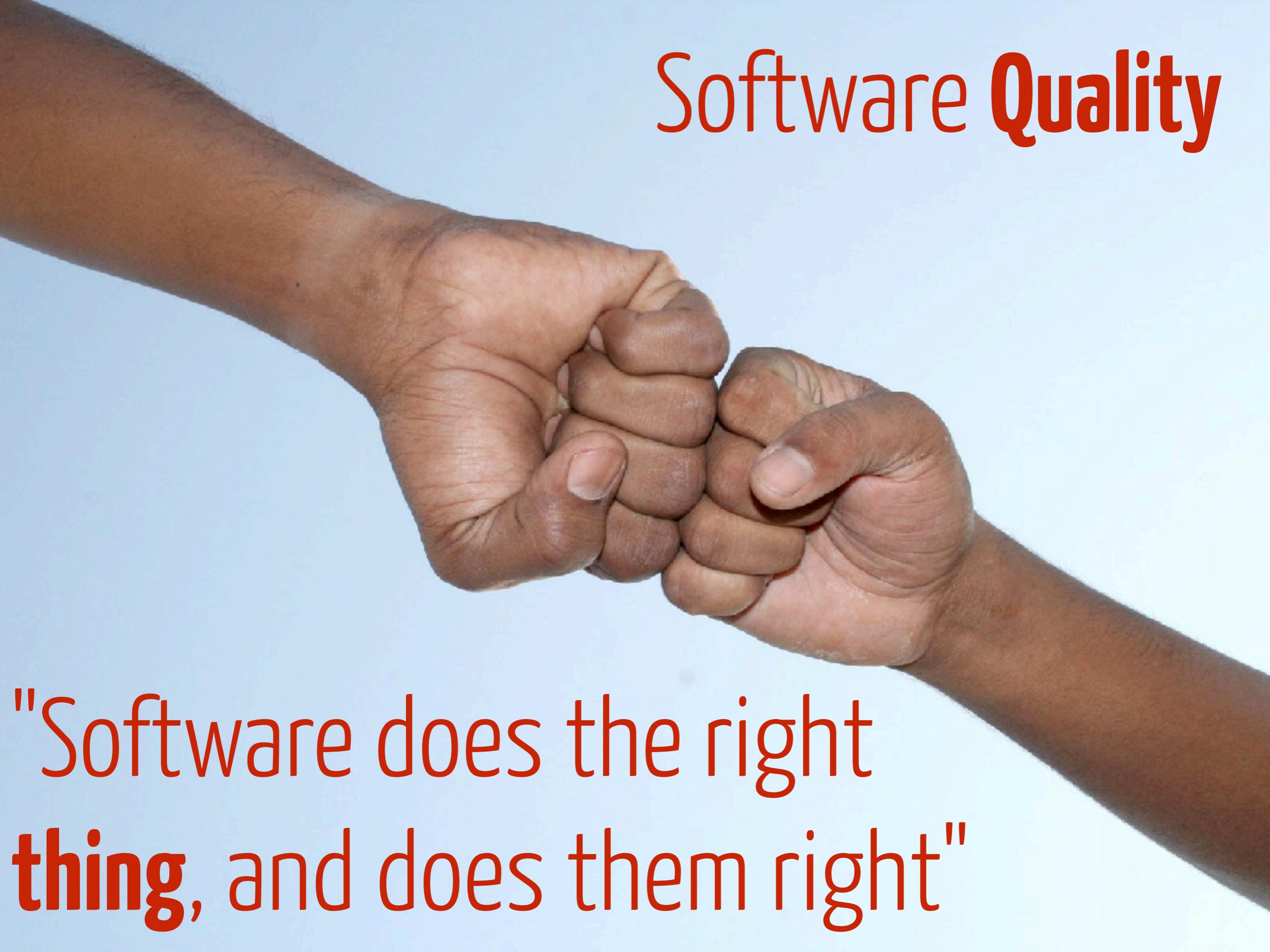




# Opening Session

Be curious about Agility / DevOps

# Software Quality



"Software does the right  
**thing**, and does them right"

# Software...

---

Does the right thing

- Validation
- « Building the right system »



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

Does them right

- Verification
- « Building the system right »



<https://www.techopedia.com>

What are  
those things?

# Requirements

---

What the *system* should do

- [Q requirements vs — Search with Google](#)
- [Q requirements vs specifications](#)
- [Q requirements vs user stories](#)
- [Q requirements vs use cases](#)
- [Q requirements vs qualifications](#)
- [Q requirements vs design](#)
- [Q requirements vs constraints](#)
- [Q requirements vs acceptance criteria](#)
- [Q requirements vs scope](#)
- [Q requirements vs needs](#)
- [Q requirements vs features](#)

# Requirements vs Specification

---

Both describe the **what** (more than **how**)

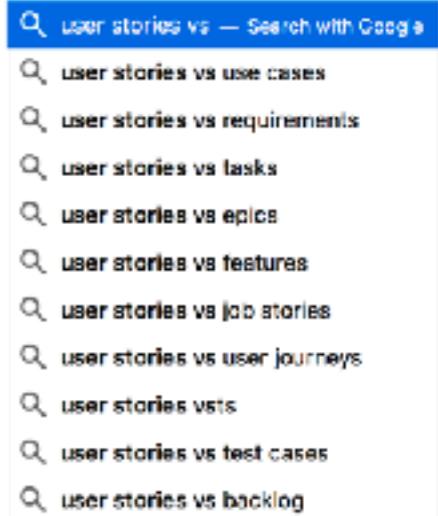
Different purpose and scope:

- specification => technical properties
- reqs => properties from the user or environment

# User Stories

---

What the user wants to do  
with the system



# Requirements vs User Stories

🔍 requirements vs — Search with Google

🔍 requirements vs specifications

🔍 requirements vs user stories

🔍 requirements vs use cases

🔍 requirements vs qualifications

🔍 requirements vs design

🔍 requirements vs constraints

🔍 requirements vs acceptance criteria

🔍 requirements vs scope

🔍 requirements vs needs

🔍 requirements vs features

🔍 user stories vs — Search with Google

🔍 user stories vs use cases

🔍 user stories vs requirements

🔍 user stories vs tasks

🔍 user stories vs epics

🔍 user stories vs features

🔍 user stories vs job stories

🔍 user stories vs user journeys

🔍 user stories vsts

🔍 user stories vs test cases

🔍 user stories vs backlog

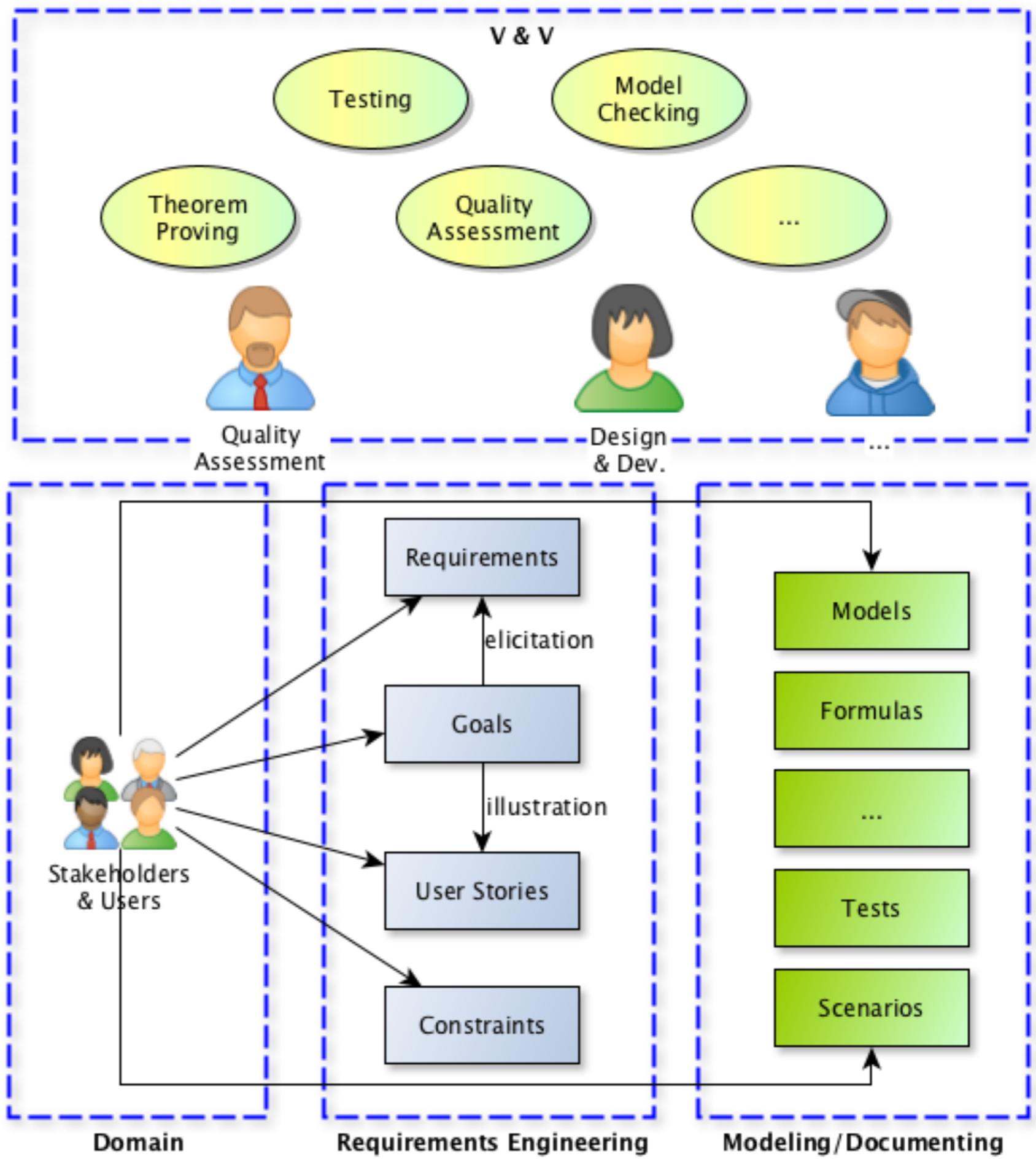
# Agility: Lots of « concepts »...

---

A word cloud diagram centered around the word "user". Other prominent words include "features", "cases", "requirements", "user", "use", "jobs", "tasks", "constraints", "journeys", "qualifications", "acceptance", "specifications", "backlog", "epics", "scope", "test", "design", "needs", "criteria", and "constraints". The words are arranged in a circular, overlapping pattern, with colors ranging from black, red, blue, and orange.

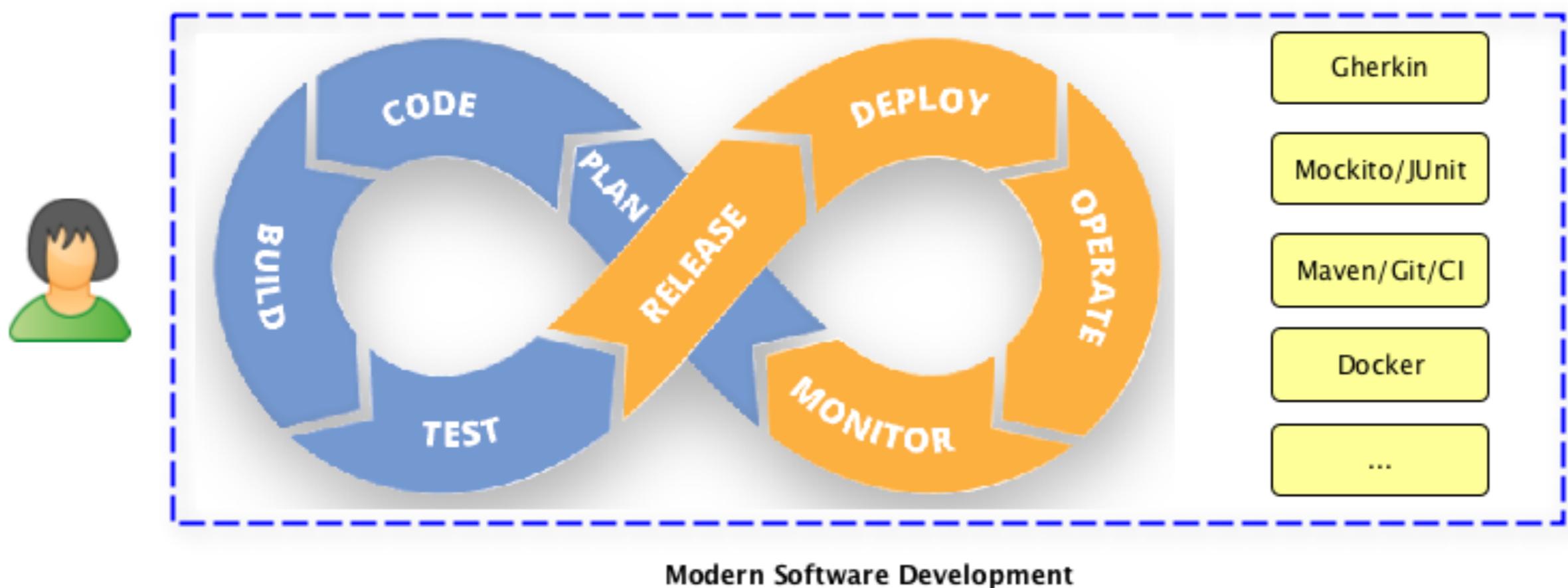
So, what's this  
tutorial about?

# Big Picture

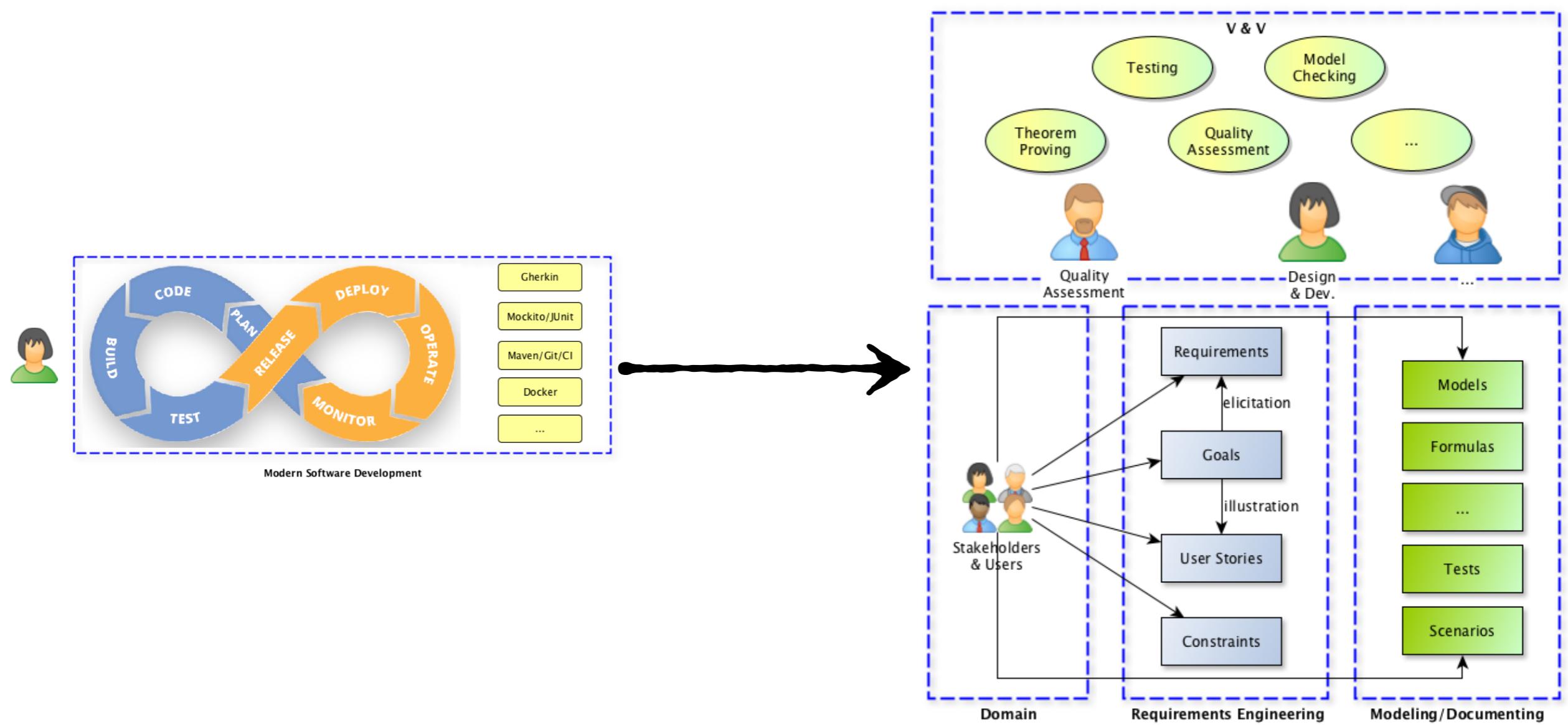


# Modern software development: **DEVOPS**

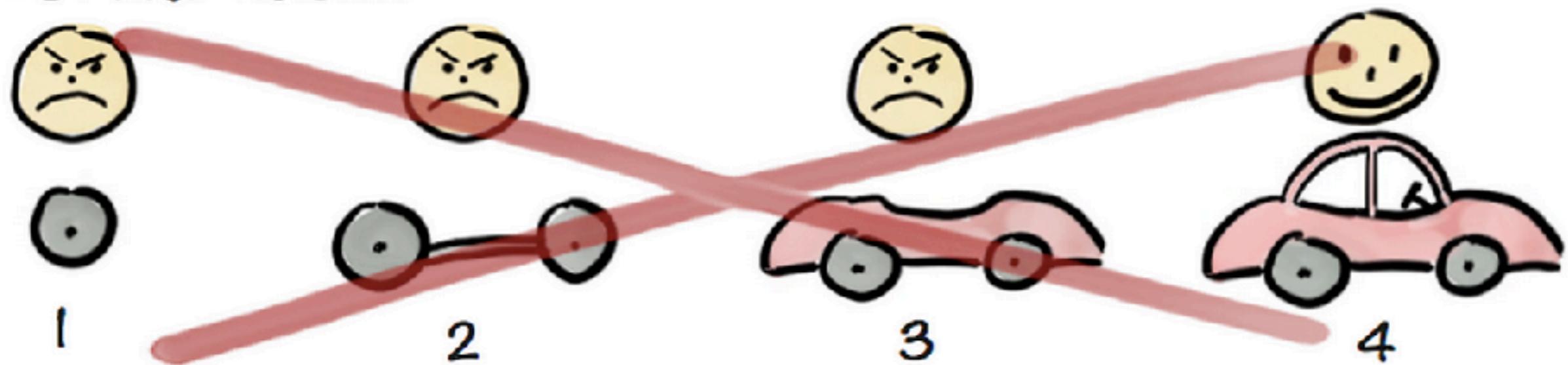
---



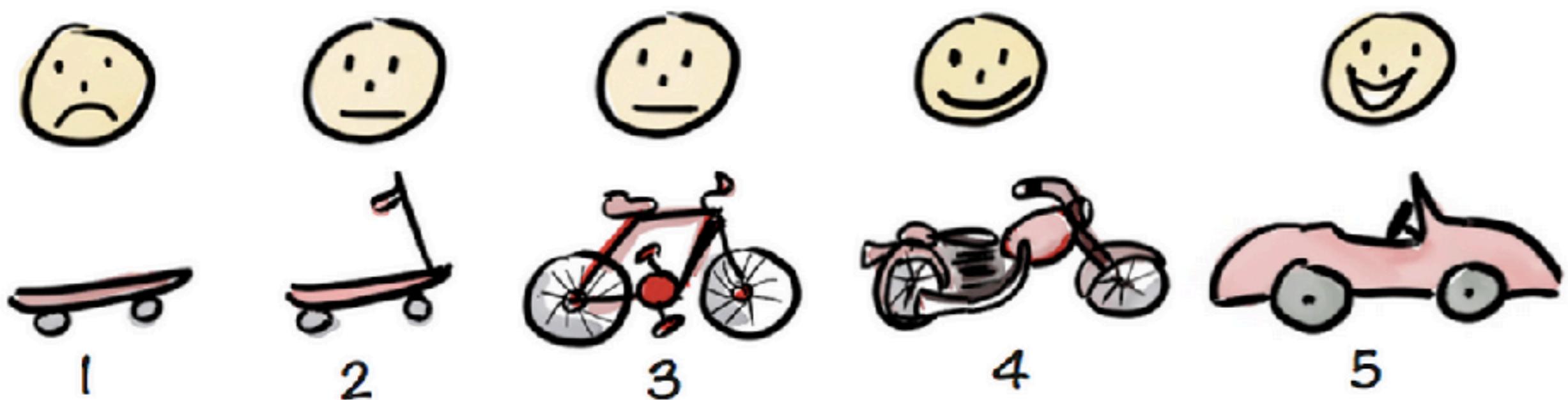
# Apply good practice to RE



**Not like this....**



**Like this!**



by Henrik Kniberg

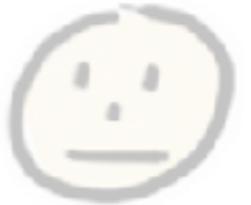
Not like this....



1

Like this

**Requirements**



1



2



3



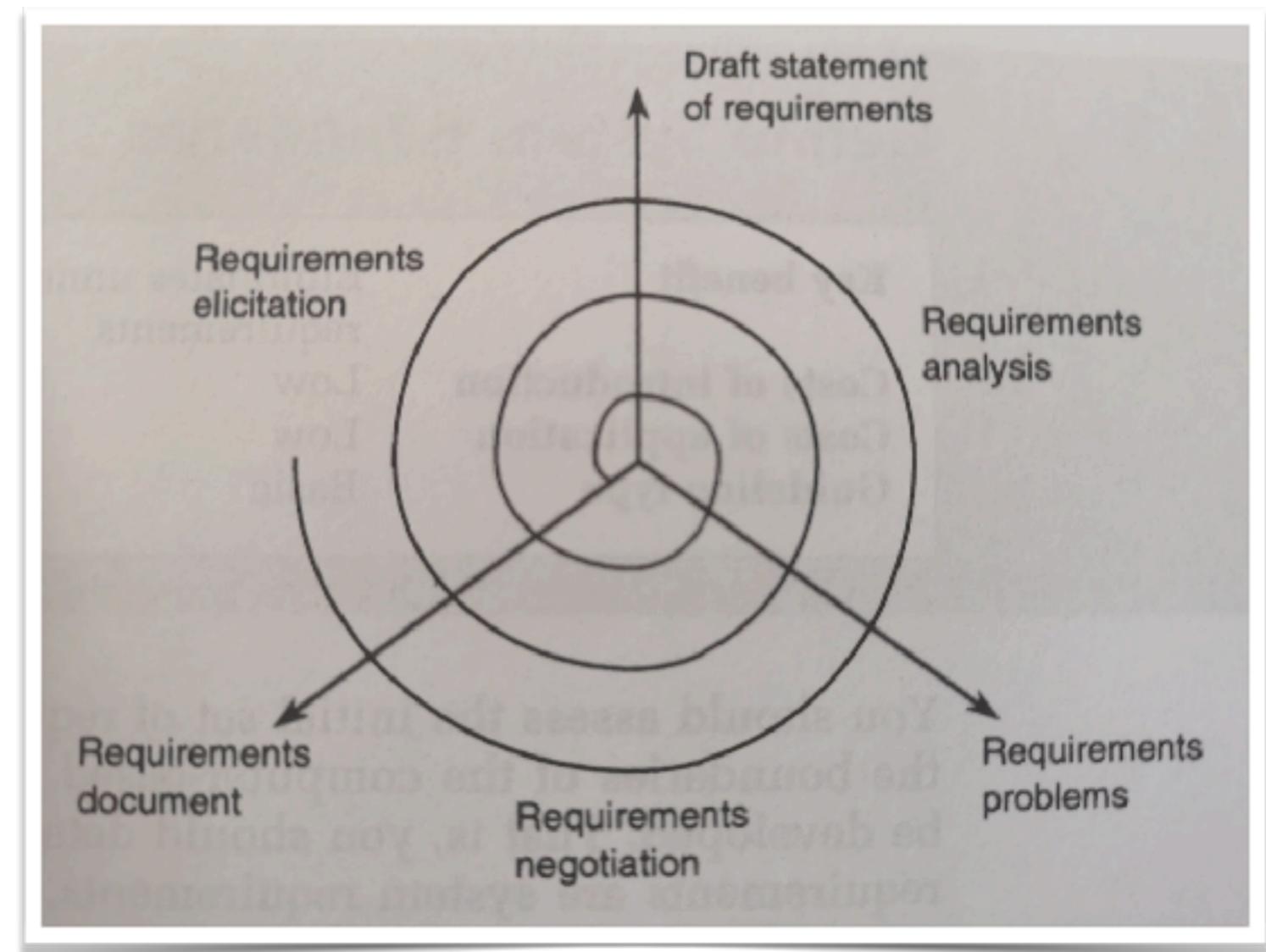
4



5

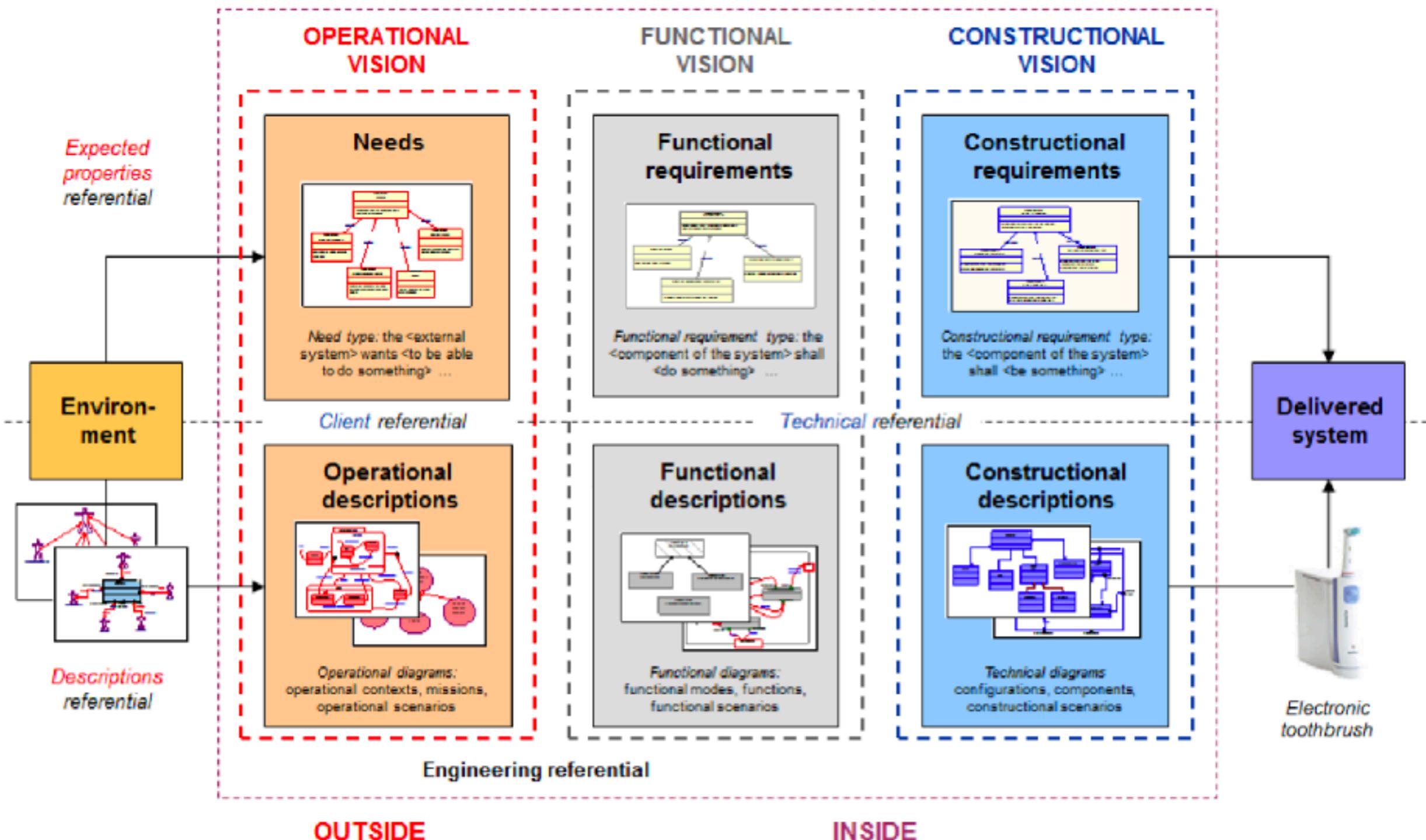
# Requirements Engineering Process

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

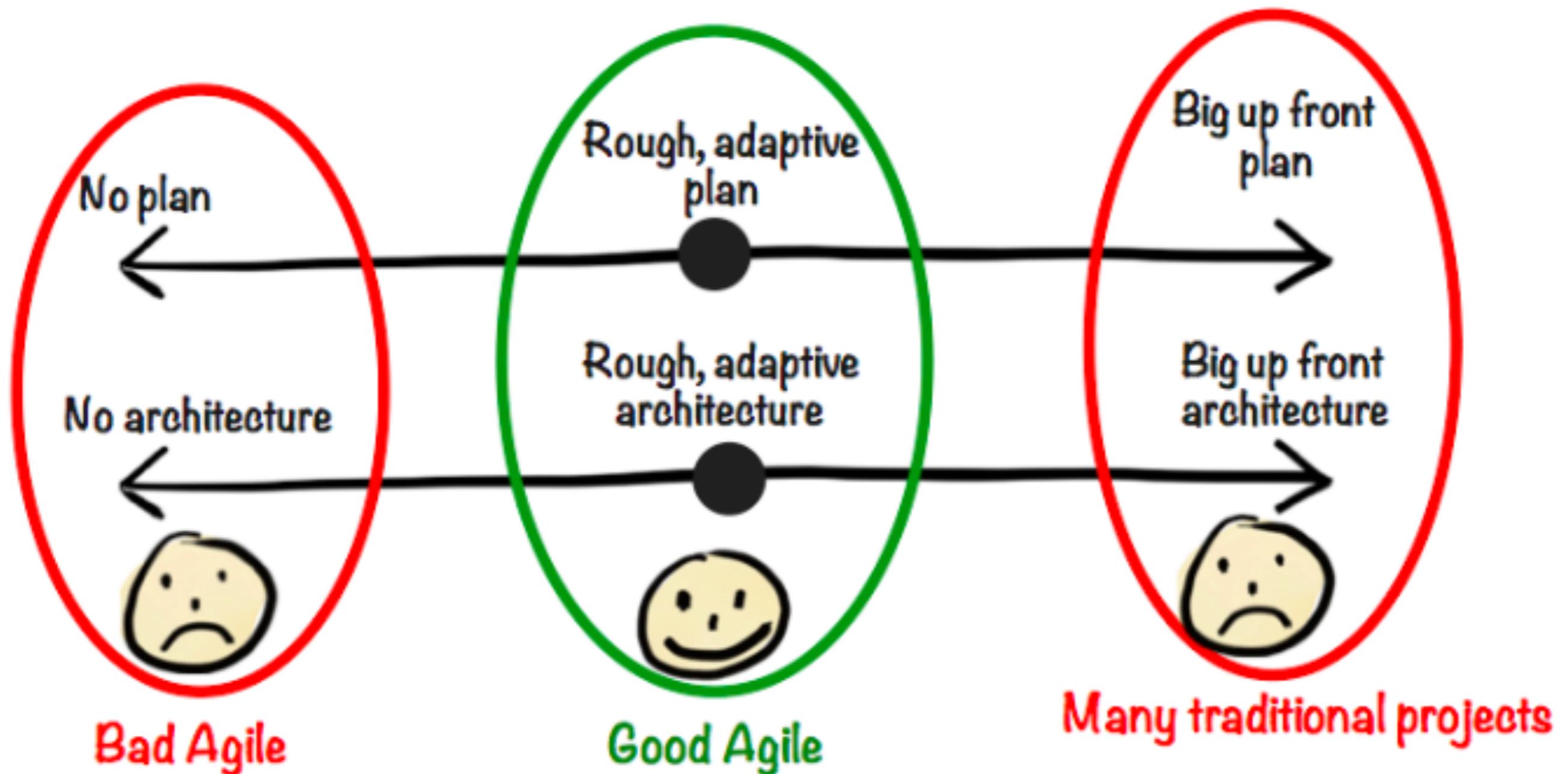


[Sommerville & Sawyer 1997]

# Expected properties vs. Descriptions



# Don't go overboard with Agile!



# Identifying Personae

Dropping the masks



# Problem Description

“Your job is to compare several pairs of poker hands and to indicate which, if either, has a higher rank.

# Minimal & Viable Product

---

- Read simple cards from the CLI (e.g., “3 5 7 2”)
- Elect the winner using the “Highest Card” rule
- That's all folks

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.println("Black:");  
    String[] line1 = scanner.nextLine().split(" ");  
    System.out.println("White:");  
    String[] line2 = scanner.nextLine().split(" ");  
    Arrays.sort(line1);  
    Arrays.sort(line2);  
    if(line1[line1.length-1].compareTo(line1[line2.length-1]) > 0 ) {  
        System.out.println("Black wins with: " + line1[line1.length-1]);  
    } else {  
        System.out.println("White wins with " + line2[line2.length-1]);  
    }  
}
```

**That's all  
folks!**



**Problems?**

**Readability?**

**Compliance with the specs?**

**Maintainability?**

**Testability?**

**Extension to fulfil the specs?**

# But first things first...

---



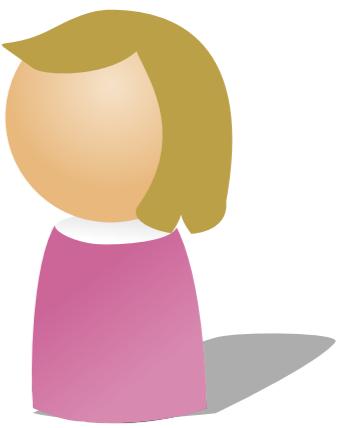
## Who will use the product?

Persona = Name + Bio + Objectives

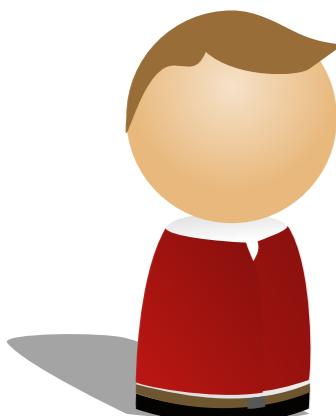
# Meet Alice & Bob



- Alice, 32yo, croupier;
- Does not like poker, do prefer blackjack and roulette;
- Fears to make careless mistake when animating a poker game and to lose her job for such a mistake.



goal: secure her job

A cartoon illustration of a person with blonde hair, wearing a pink dress, sitting at a desk with a laptop.

- Bob, 28yo, poker player (average plus);
- Plays poker at home with friends, and love to visit Las Vegas;
- Trains a lot on the Internet to prepare for a BIG tournament.

goal: improve his skills



# Writing Epics & Stories

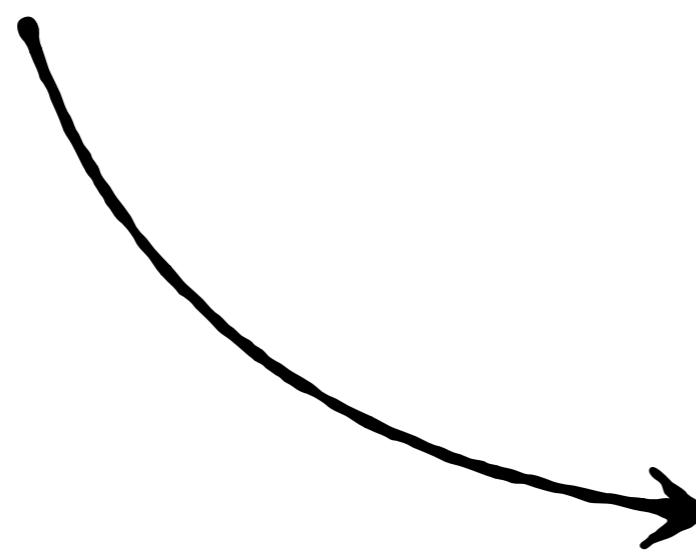
Specifying in an agile way

**How to chop  
an elephant?**



# Product

**Tasks & Deliverables**



# Maximize Value, not Output

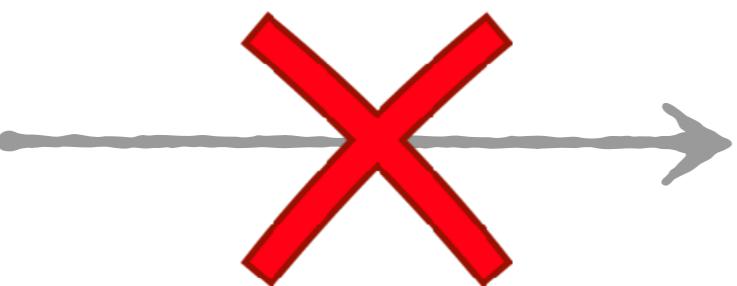


# Specifying with Stories

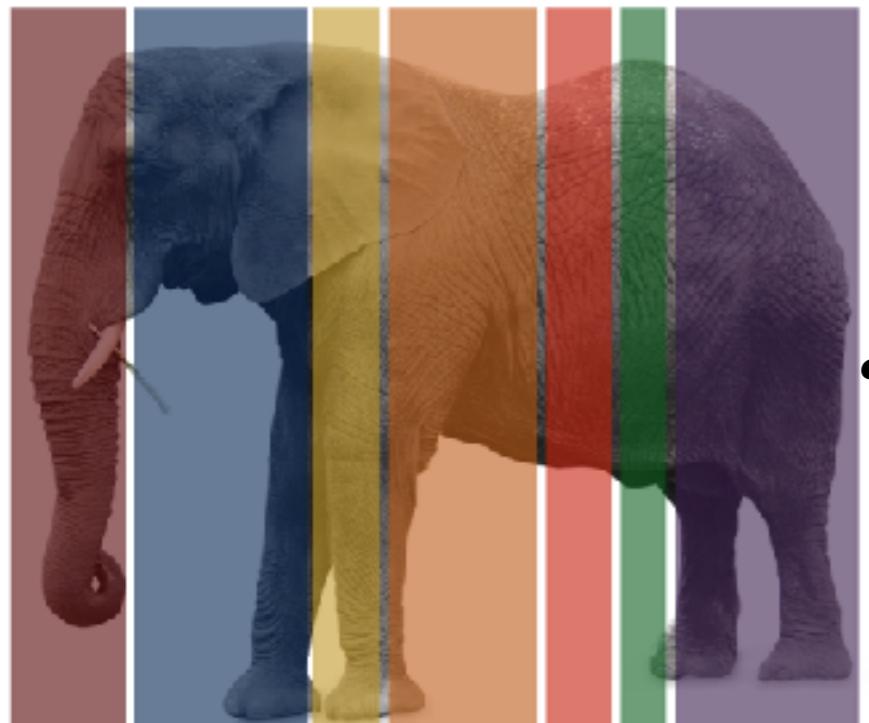
---

Epic = { Persona  
          Action  
          Benefits

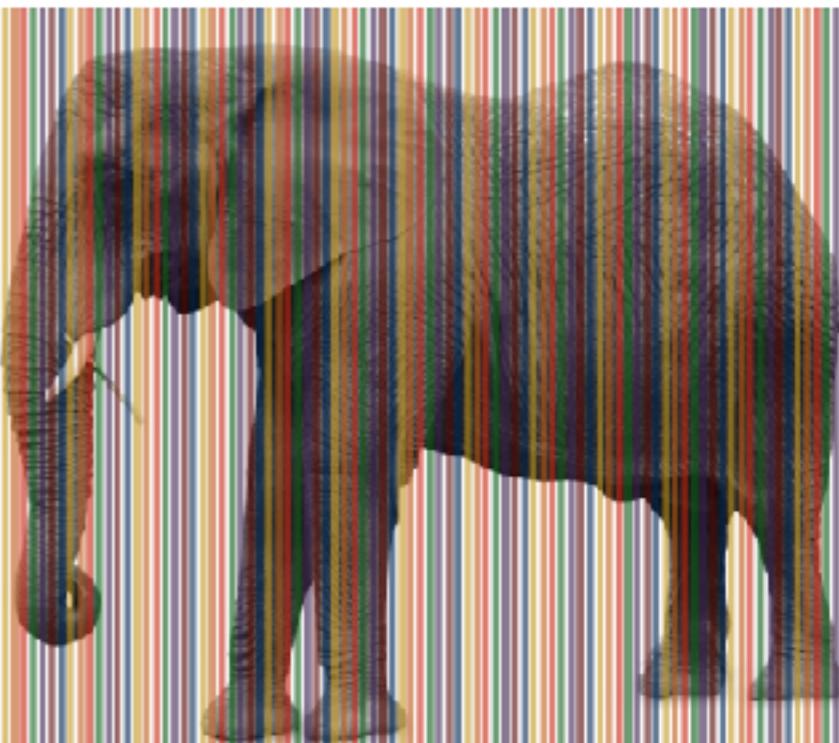
Story = Epic + { Acceptance criteria  
                  Tests, (Estimations)



**identify epics**



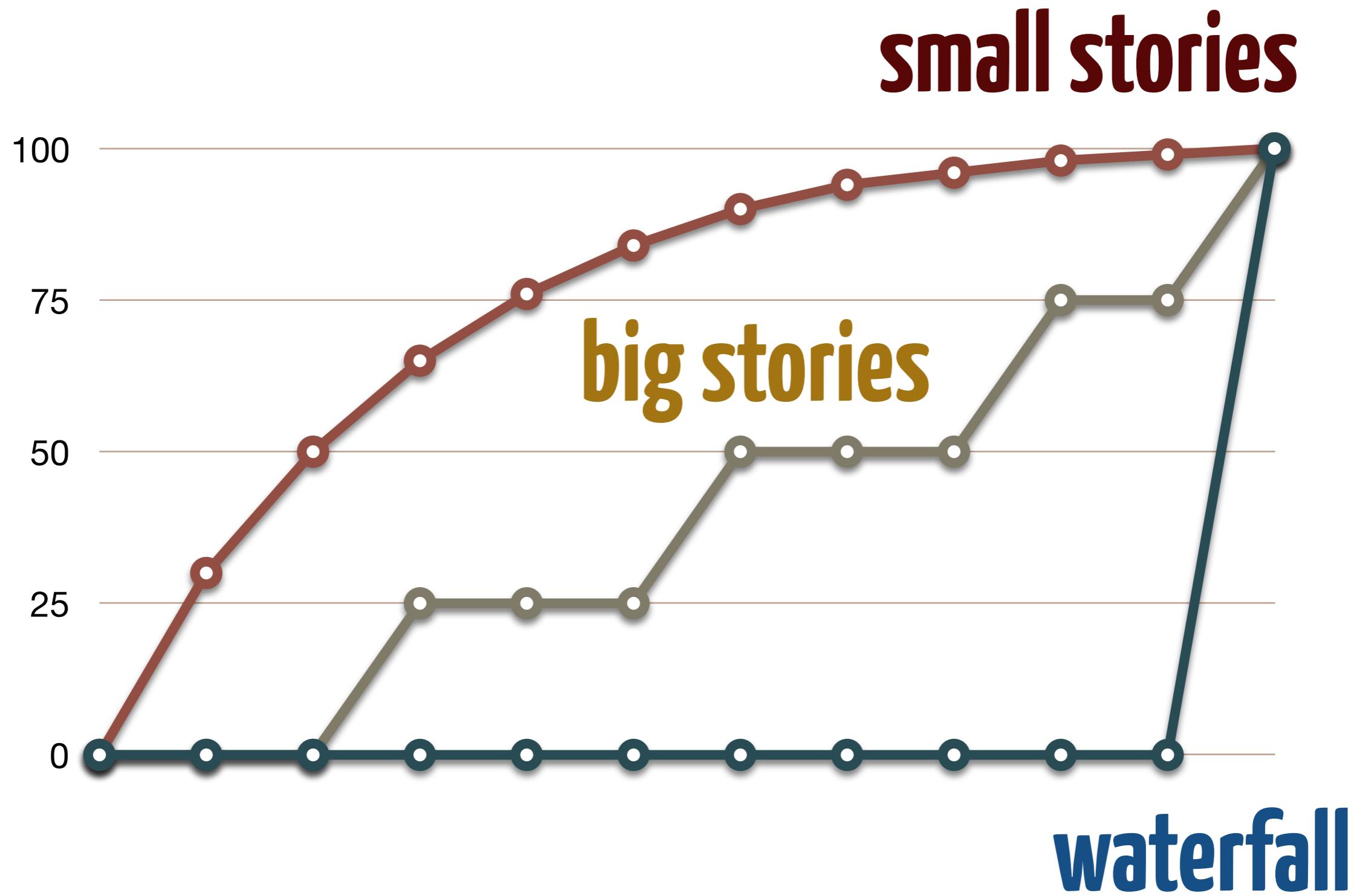
**refine  
into  
stories**



(Inspired by Alistair Cockburn's elephant carpaccio workshop)

# Cumulative value

---





Story

Epic

Story name

As

I want to

So that

Story Cards



business value



story points



ready?



Story ID

Time to practice!

---



**What are the epics associated  
to Alice and Bob?**

# Detect Cheaters

Story name



Story



Epic

As *Alice*

I want to *identify cheaters in a game*

So that *I can keep my job*



business value



story points



ready?



# Story ID

1

USER STORY #1

AS A USER, I'D LIKE TO  
BE ABLE TO USE THE  
ENTIRE FUNCTIONING  
APPLICATION.

THIS IS HARDLY ACTIONABLE  
OR TESTABLE. HAVE YOU EVER  
HEARD OF "INVEST" USER STORIES?

WE JUST LIKE TO  
KEEP THEM "VAGUE".  
VAST, ARDUOUS,  
GARGANTUAL, UNCLEAR  
AND ETERNAL.

A user story  
is an  
**INVEST**ment

I ndependent  
N egociable  
V alued  
E stimable  
S mall  
T estable

///

A **USER story** is to a **USE case**  
as a **gazelle** is to a **gazebo**.

- Alistair Cockburn



**gazelle**

**gazebo**



“A **user story** is the **title**  
of one scenario, where a  
**use case** is the **contents**  
of multiple scenarios

- Alistair Cockburn

**By adding  
value to the  
product, a  
story is by  
essence**

**VERTICAL**

# Time to practice!

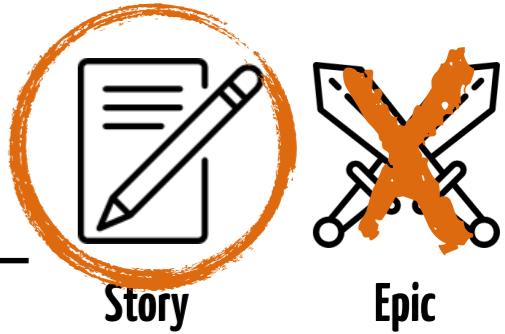
---



# Refine the epics into stories!

# Duplicated Cards

Story name



As *Alice*

I want to identify cheaters who trick  
the card deck with extra cards

So that I can report to management



business value



story points



ready?



#  
Story ID

42



# Acceptance Criteria

One enters two overlapping hands (e.g. the two hands contains a QD). The game echoes an error stating that it encountered a cheat attempts.

---



## Comments



Related to Epic #1 « Detect Cheaters »

# Estimation

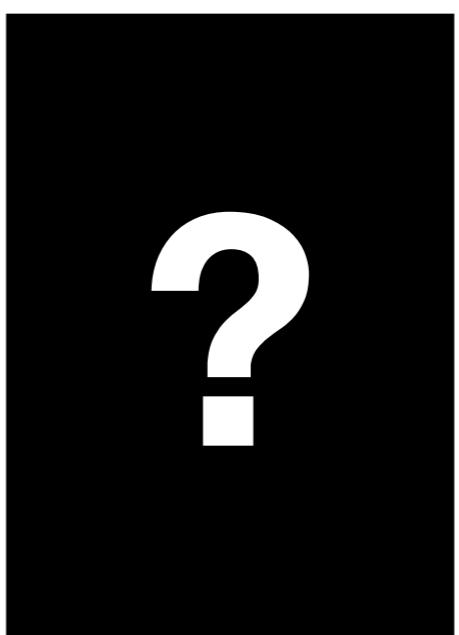


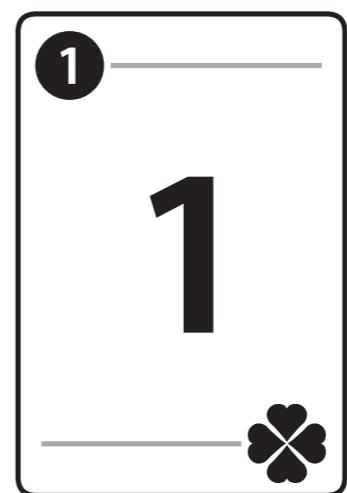
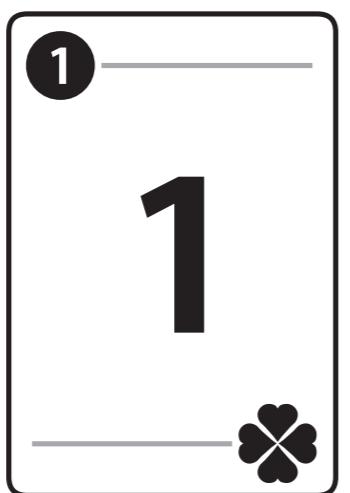
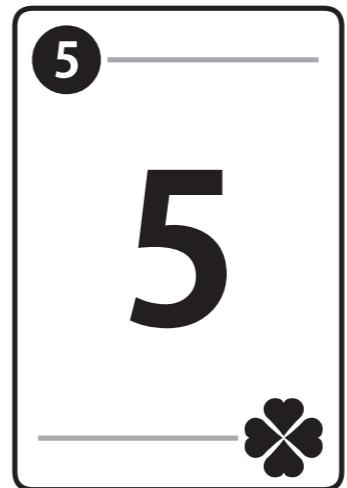
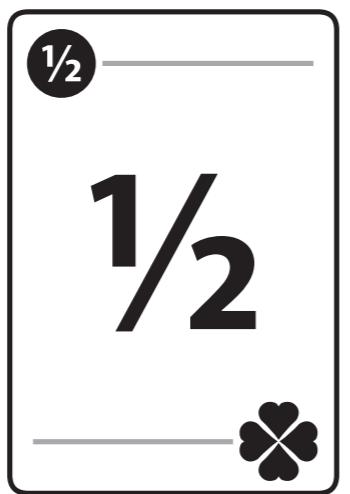
Planning Poker

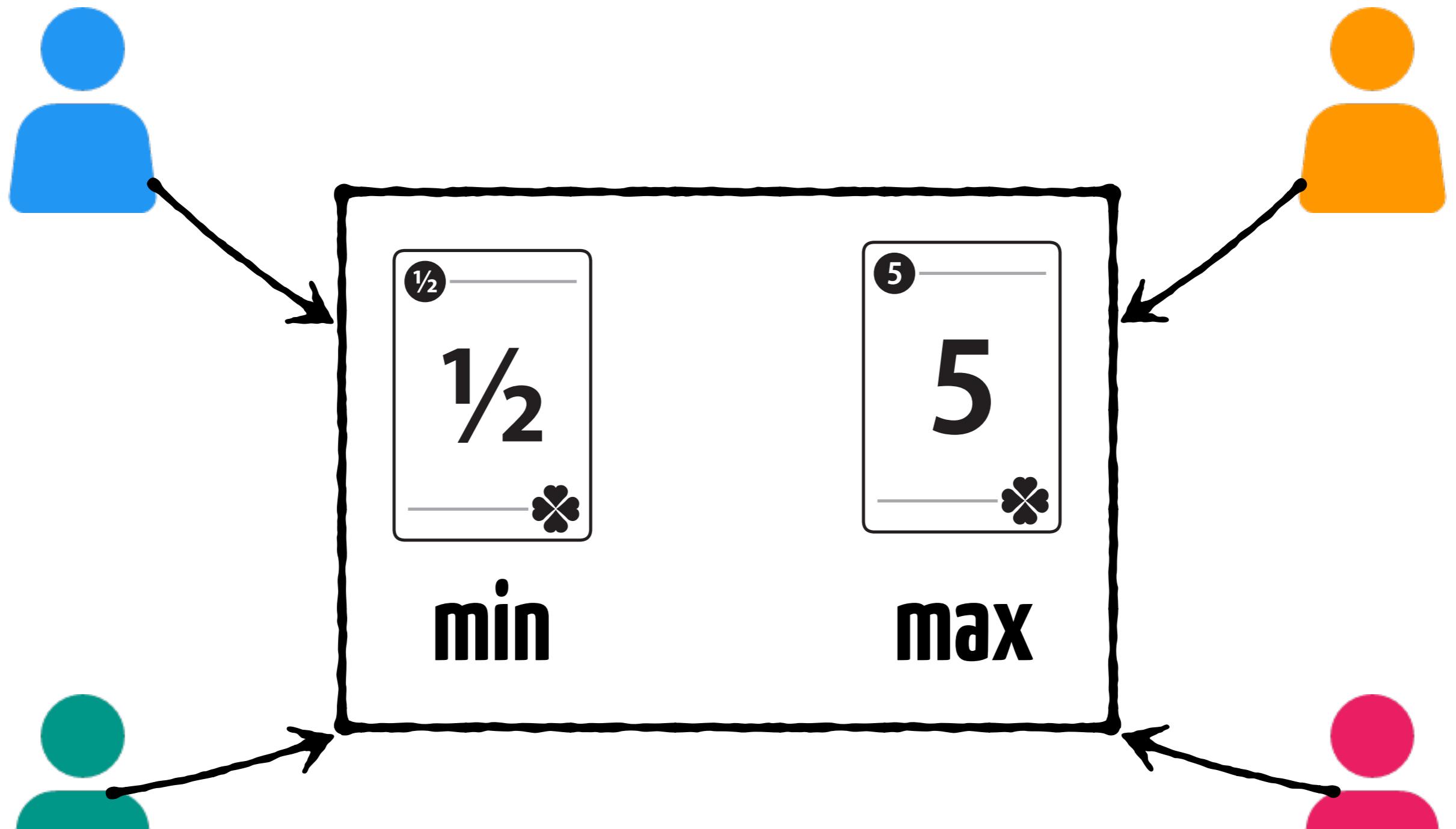


**What is the « cost » of developing  
story #42 for the team?**

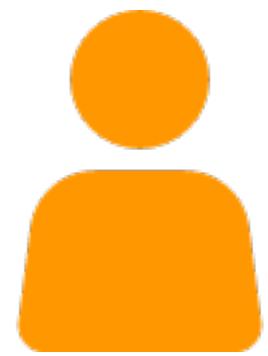
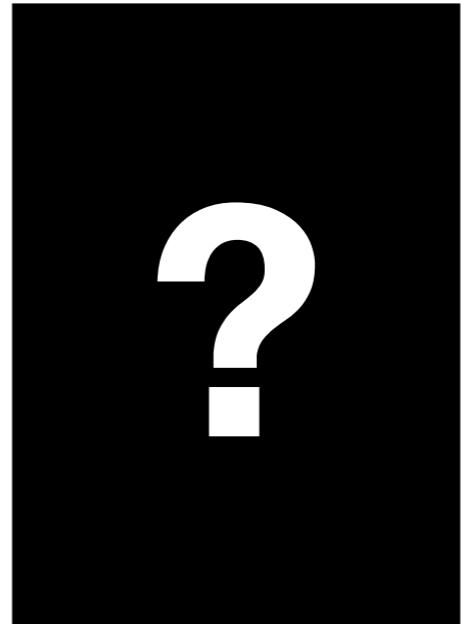
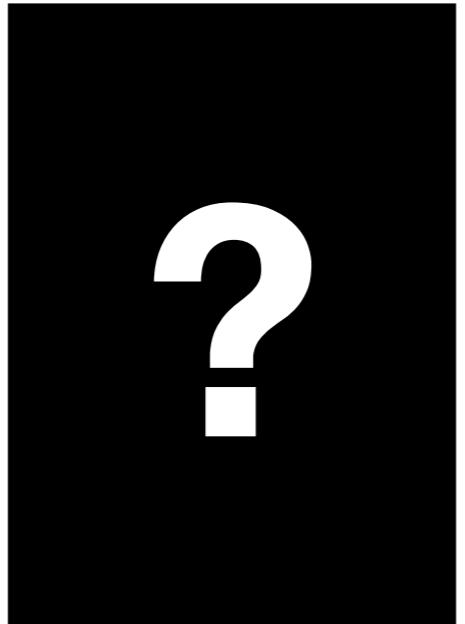


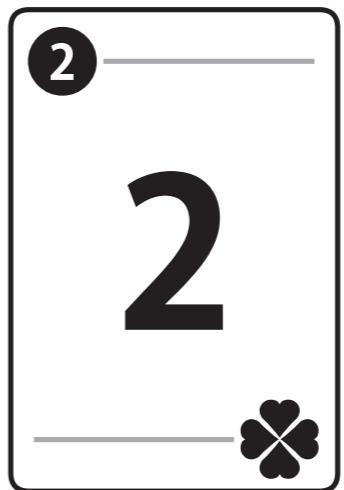






# Discussions





Time to practice!



Estimate a couple of stories!



business value

cost



# Implementing Stories & Tests

Let's craft some code!

# How to validate these 2 stories?



*As Bob, I want to enter my hand on the command line so that the game knows the contents of my hand*

?

*As Alice, I want to identify cheaters that trick the card deck so that I can report cheating attempts to management*

?

```
azrael:agile-tutorial mosser$ java -jar target/poker-game.jar
Enter 1st player hand: QD QH KC KH 3S
1st: [KING of HEARTS, QUEEN of DIAMONDS, THREE of SPADES, QUEEN of HEARTS, KING of CLUBS]
Enter 2nd player hand: 7S 6H 4S 3D 2C
2nd: [FOUR of SPADES, SEVEN of SPADES, TWO of CLUBS, SIX of HEARTS, THREE of DIAMONDS]
Exception in thread "main" java.lang.UnsupportedOperationException: Cannot determine winner!
    at re.poker.Game.declareWinner(Game.java:23)
    at re.poker.Main.main(Main.java:22)
```

# Implement the HC detection!



*As Alice,  
I want to compare two hands according  
to the « Highest Card » combination  
so that I can identify the winner*

```
public String declareWinner() {  
    throw new UnsupportedOperationException("Cannot determine winner!");  
}
```

# Run the Test suite!



		411 ms
▶	poker (re)	327 ms
▶	GameTest	327 ms
✓	rejectNonLegitHands	0 ms
●	declareTheWinner	84 ms
✓	HandTest	74 ms
✓	readAPokerHandFromAStream	9 ms
✓	properlyBuildAnHand	1 ms
✓	rejectAnHandWithDuplicatedCards	0 ms
✓	rejectAnHandWithTooManyCards	0 ms
✓	rejectAnHandWithNotEnoughCards	0 ms
✓	CardTest	0 ms
✓	identifyDifferentCardsBasedOnSuits	0 ms
✓	comparisonDiffersFromEquality	0 ms
✓	identifyEqualsCard	0 ms
✓	orderCardsBasedOnValues	0 ms
✓	identifyDifferentCardsBasedOnValues	0 ms
✓	CardValueTest	0 ms
✓	checkMappingBetweenSymbolsAndValues	0 ms
✓	SuitTest	0 ms
✓	checkMappingBetweenSymbolsAndSuits	0 ms

# Test your code!



## Can you write better tests?

```
@Test  
public void thisIsATest() {  
    Game theGame = new Game();  
    theGame.submit("Bob", new Hand("AC KC QC JC TC"));  
    CardValue v = CardValue.valueOf("ACE");  
    Suit s = Suit.valueOf("CLUBS");  
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());  
    Card max = Collections.max(theGame.getByPlayer("Bob").getCards());  
    assertEquals(theCard, max);  
}
```

# It's nearly break time ...

---



**As Requirement Engineers,  
are you satisfied by your test?**

**#FoodForThoughts**

# Status update

---

- **Opening session**

1. Identifying Personae ✓
2. Writing Epics & Stories ✓
3. Implementing & Testing stories ✓

- **BREAK**

4. Automating Acceptance Scenarios
5. Deploying a Cont. Int. stack
6. From CI to Cont. Deployment

- **Closing session**





Break

# How was your coffee?

---



As Requirement Engineers,  
are you satisfied by your test?



# ACCEPTED

**Implementing  
Acceptance Scenarios**

From tests to scenarios

# What about this one?

---

**Scenario: Identify the highest card in an hand**

**Given** a new game

**When Bob** submits the following cards: **AC KC QC JC TC**

**Then Bob's** highest card is the **ACE** of **CLUBS**

```
@Test
public void IdentifyTheHighestCardInAnHand() {
    Game theGame = new Game();
    theGame.submit("Bob", new Hand("AC KC QC JC TC"));
    CardValue v = CardValue.valueOf("ACE");
    Suit s = Suit.valueOf("CLUBS");
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());
    Card max = Collections.max(theGame.getByPlayer("Bob").getCards());
    assertEquals(theCard, max);
}
```

# What about this one?

---

**Scenario: Identify the highest card in an hand**

**Given** a new game

**When Bob** submits the following cards: **AC KC QC JC TC**

**Then Bob's** highest card is the **ACE** of **CLUBS**

```
@Test
public void IdentifyTheHighestCardInAnHand() {
    Game theGame = new Game();
    theGame.submit("Bob", new Hand("AC KC QC JC TC"));
    CardValue v = CardValue.valueOf("ACE");
    Suit s = Suit.valueOf("CLUBS");
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());
    Card max = Collections.max(theGame.getByPlayer("Bob").getCards());
    assertEquals(theCard, max);
}
```

# What about this one?

---

**Scenario: Identify the highest card in an hand**

**Given** a new game

**When** Bob submits the following cards: AC KC QC JC TC

**Then** Bob's highest card is the ACE of CLUBS

```
@Test
public void IdentifyTheHighestCardInAnHand() {
    Game theGame = new Game();
    theGame.submit("Bob", new Hand("AC KC QC JC TC"));
    CardValue v = CardValue.valueOf("ACE");
    Suit s = Suit.valueOf("CLUBS");
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());
    Card max = Collections.max(theGame.getByPlayer("Bob").getCards());
    assertEquals(theCard, max);
}
```

# What about this one?

---

**Scenario: Identify the highest card in an hand**

**Given** a new game

**When** Bob submits the following cards: AC KC QC JC TC

**Then** Bob's highest card is the ACE of CLUBS

```
@Test
public void IdentifyTheHighestCardInAnHand() {
    Game theGame = new Game();
    theGame.submit("Bob", new Hand("AC KC QC JC TC"));
    CardValue v = CardValue.valueOf("ACE");
    Suit s = Suit.valueOf("CLUBS");
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());
    Card max = Collections.max(theGame.getByPlayer("Bob").getCards());
    assertEquals(theCard, max);
}
```

# IDE Integration (IntelliJ example)

The screenshot shows the IntelliJ IDE's Run tool window with the title "Run: Feature: ReadHands". The "Test Results" section is expanded, displaying a tree of test cases and their execution times. All tests are marked as successful (green checkmarks).

Test Case	Execution Time (ms)
Feature: Reading a Poker Hand	774 ms
Read a regular hand	759 ms
a new game	375 ms
Bob submits the following cards: QD TS 2C KD 3C	369 ms
Bob's hand contains 5 cards	3 ms
Bob's hand contains the QUEEN of DIAMONDS	2 ms
Bob's hand contains the TEN of SPADES	2 ms
Bob's hand contains the TWO of CLUBS	1 ms
Bob's hand contains the KING of DIAMONDS	4 ms
Bob's hand contains the THREE of CLUBS	3 ms
Reject an hand with a duplicated card	11 ms
a new game	1 ms
Bob submits the following cards: QD TS 2C KD QD	0 ms
a cheat attempt is detected!	10 ms
Detect cheaters when submitting duplicated cards	4 ms
a new game	0 ms
Bob submits the following cards: AC KC QC JC TC	3 ms
Alice submits the following cards: QD QH QS QC KD	1 ms
a cheat attempt is detected!	0 ms

# Steps auto-completion

```
Feature: Reading a Poker Hand

Scenario: Read a regular hand
  Given a new game
  When Bob submits the following cards: QD TS 2C KD 3C
  Then Bob's hand contains 5 cards
    And Bob's hand contains the QUEEN of DIAMONDS
    And Bob's hand contains the TEN of SPADES
    And Bob's hand contains the TWO of CLUBS
    And Bob's hand contains the KING of DIAMONDS
    And Bob's hand contains the THREE of CLUBS

Scenario: Reject an hand with a duplicated card
  Given a new game
  When Bob submits the following cards: QD TS 2C KD QD
    And
  Then <string>'s hand contains <number> cards
    <string>'s hand contains the <string> of <string>
    <string>'s highest card is the <string> of <string>

Scenario: Detect a cheat attempt
  Given a new game
  When Bob submits the following cards: QD OH QS QC KD
    And a cheat attempt is detected!
  Then a new game is detected!

Use ⌘⇥ to syntactically correct your code after completing (balance parentheses etc.) ➤ π
```

# Gherkin

```
Feature: Logout from application
Scenario:
  Given I am logged in
  When I click "log out" button
  Then I am informed about successful logout
  And I am redirected to login page
```

To create a new requirements description, we need to define the **Feature** which gives us the name of a new **Functionality**. Then, we go ahead with writing the **Scenario**.

# Gherkin

Feature: Logout from application

Scenario:

Given I am logged in

When I click "log out" button

Then I am informed about successful logout

And I am redirected to login page

To create a new requirement  
need to define the **F**  
the name of **E**  
go ahead and writing the **S**  
**Requirement Engineering**  
**compatible**  
**Quality**. Then, we  
cription, we

# Regular expressions as binders

---

```
@Given("^a new game$")
public void startNewGame() {
    this.theGame = new Game();
}

@When("^(.*) submits the following cards: (.*)$")
public void enterPlayerHand(String thePlayer, String data){
    this.theGame.submit(thePlayer, new Hand(data));
}

@Then("^(.*)'s highest card is the (.*) of (.*)$")
public void checkHighestCard(String thePlayer, String value, String suit) {
    CardValue v = CardValue.valueOf(value.trim());
    Suit s = Suit.valueOf(suit.trim());
    Card theCard = new Card(v.getSymbol()+" "+s.getSymbol());
    Card max = Collections.max(theGame.getByPlayer(thePlayer).getCards());
    assertEquals(theCard, max);
}
```

# **Write your own acceptance tests**

---



**As Requirement Engineers,  
are you satisfied by these test?**



# Deploying a Continuous Integration Stack

Clone, build, test, release, ...

«Integration is a **long** and  
**unpredictable** process»

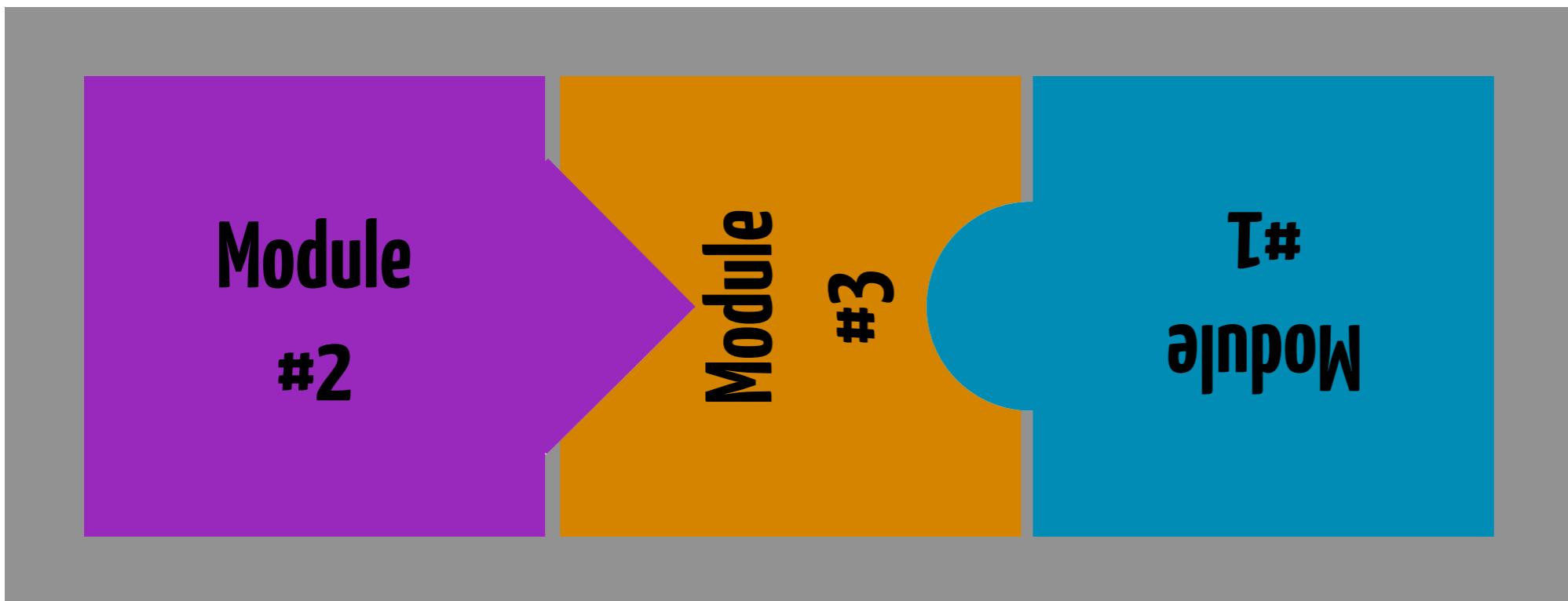
- Martin Fowler

**Module**  
**#1**

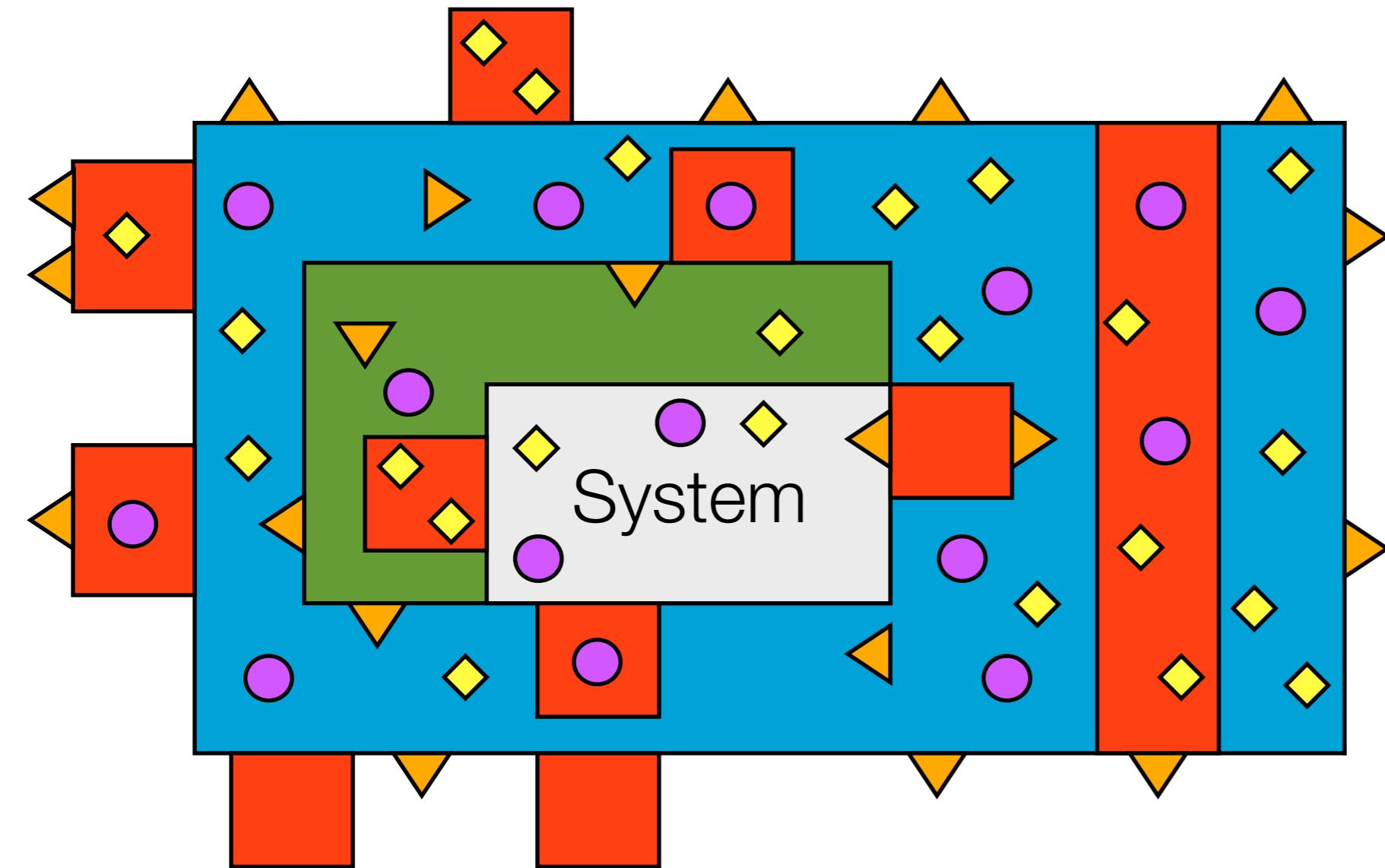
**Module**  
**#3**

**Module**  
**#2**

**modules**



# system



«real» system

# Continuous Integration

*Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage.*

---

01 May 2006



Martin Fowler

**Translations:** Portuguese Chinese  
Korean French Chinese Czech

**Tags:** popular · agile · delivery ·  
extreme programming · continuous  
integration

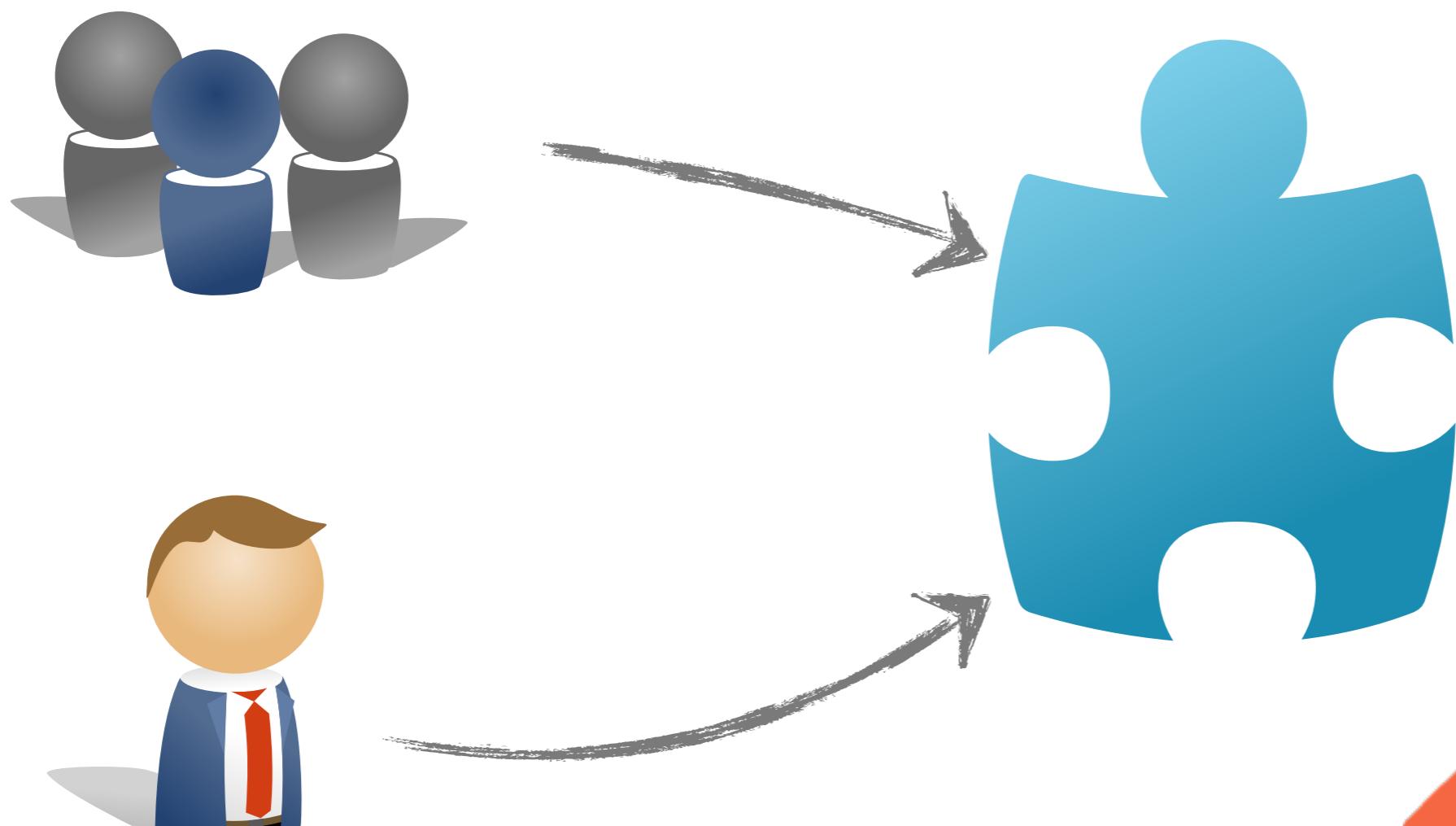
01.05.2006!

**«Integrate frequently»**

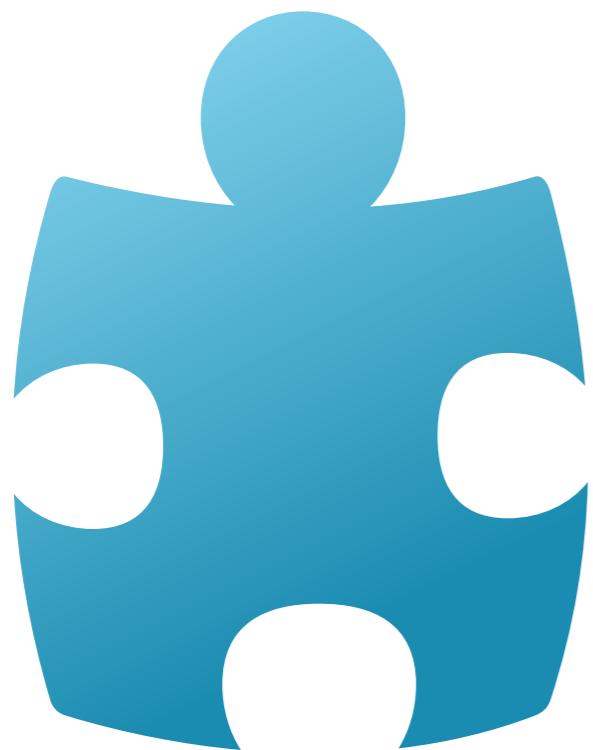
**«Integrated system is  
automatically build»**

**«Build triggers Tests»**

# «Integrate frequently»

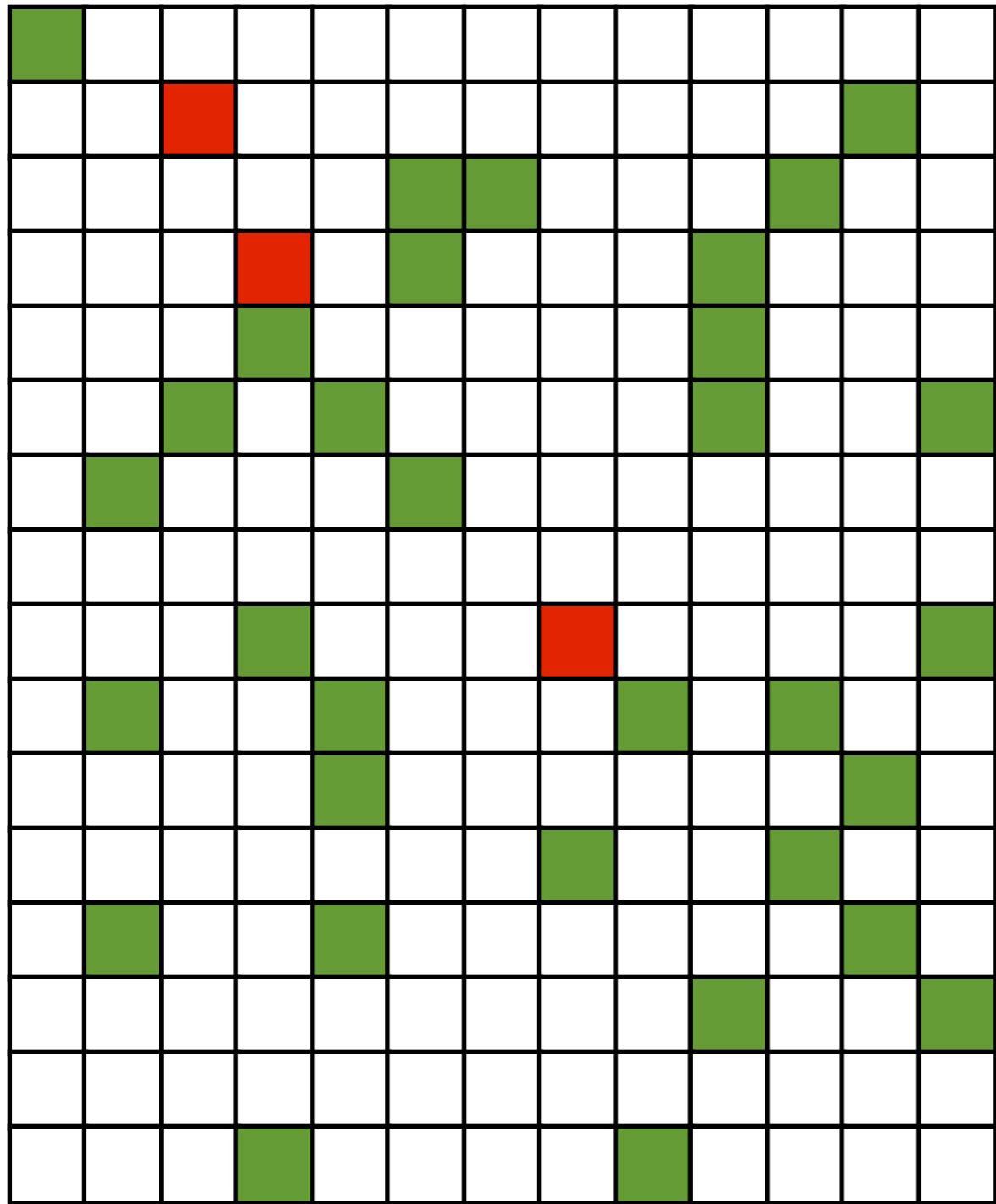


«Integrated system is  
**automatically build»**



*maven*

# «Build triggers Tests»

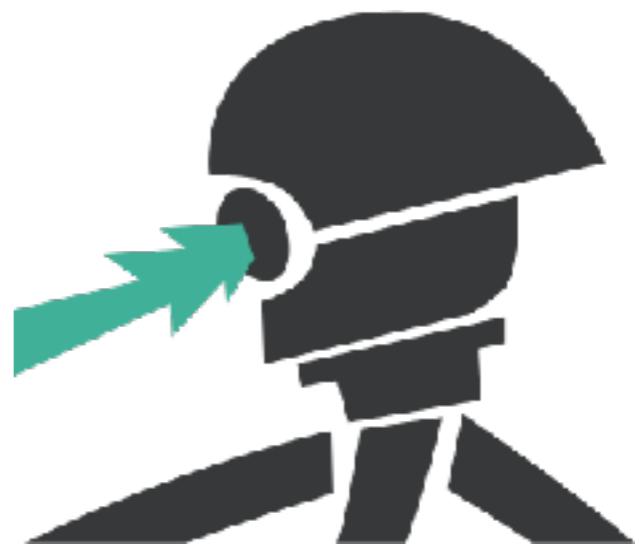


# Integration loop

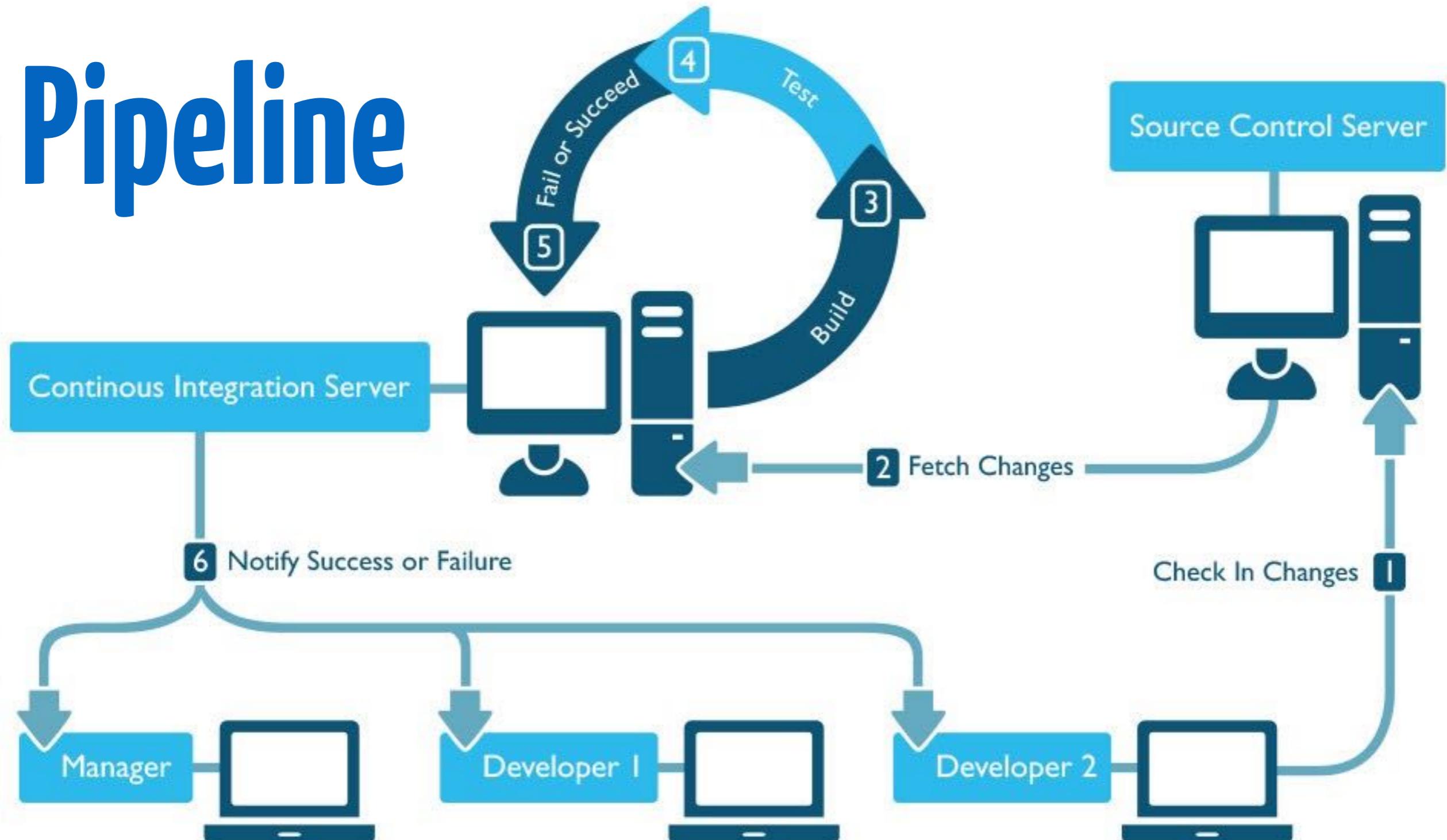


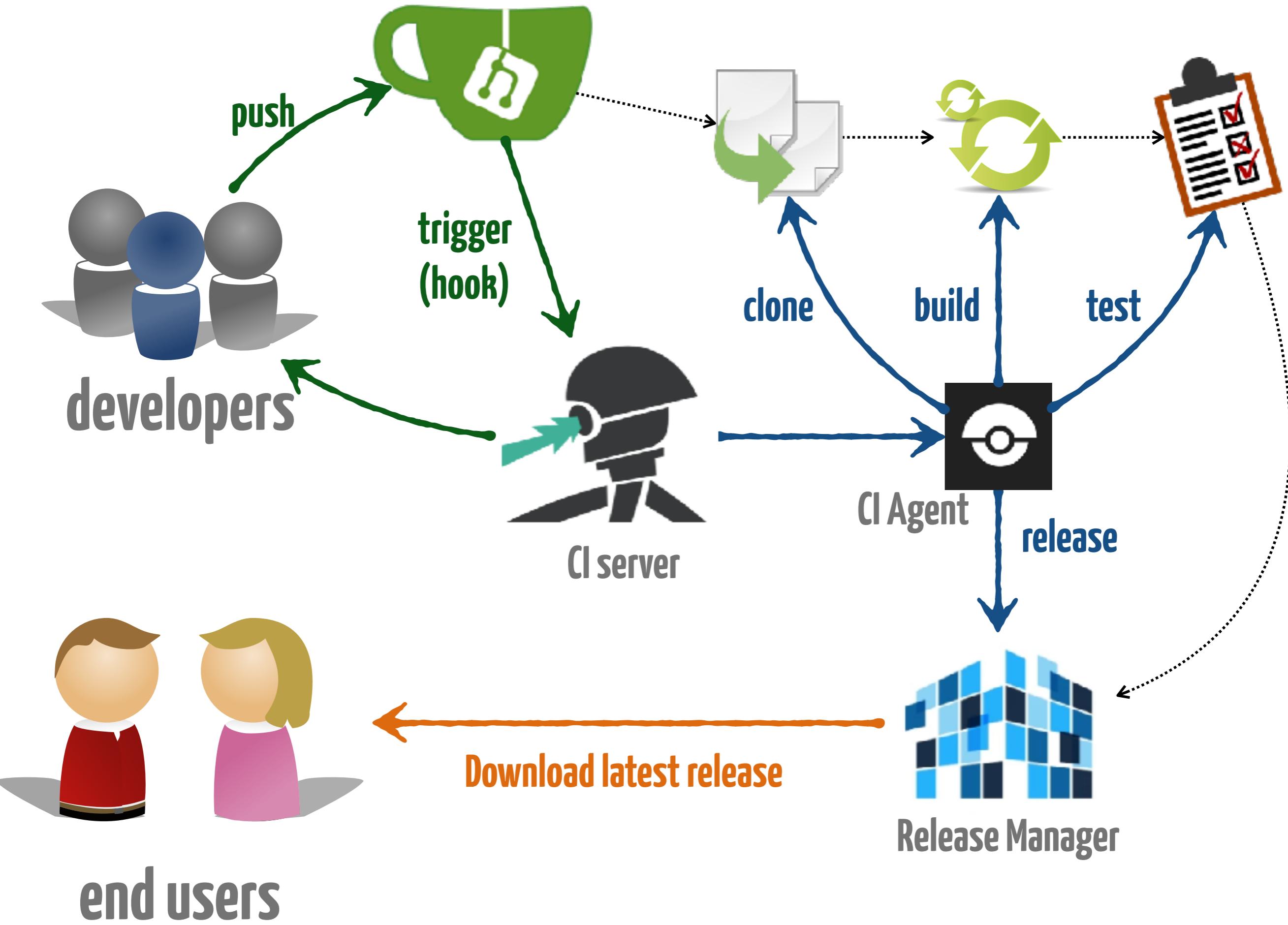
Let's integrate the  
integration loop!

drone.io



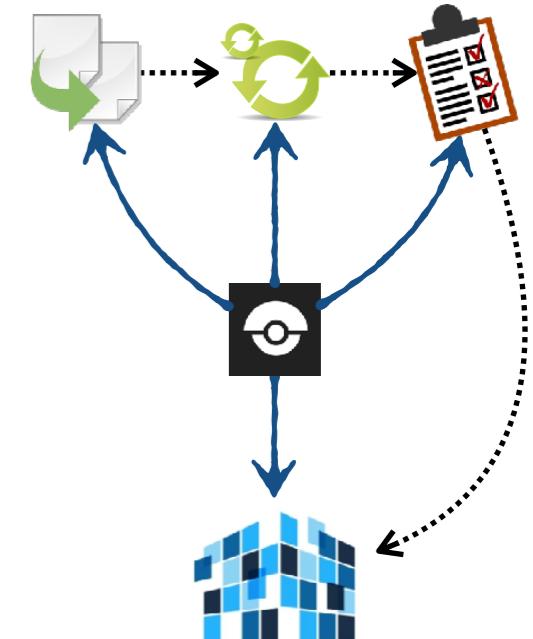
# Pipeline





# Build Pipeline model

(cloning is implicit, so can be omitted in the pipeline descriptor)



**pipeline:**

```
# Building the Poker Game using Maven
```

```
build:
```

```
  image: maven:3.5-alpine
```

```
  commands:
```

- mvn -q clean package
- mkdir releases
- cp target/poker-game.jar releases/poker-game-r\${DRONE\_BUILD\_NUMBER}.jar

**(build + tests thanks to Maven)**

```
# Publishing the last release in the release manager
```

```
scp:
```

```
  group: releasing
```

```
  image: appleboy/drone-scp
```

```
  host: release-manager
```

```
  username: root # Don't do this at home (demonstration purpose only)
```

```
  password: re18 # Don't do this at home (use secret instead)
```

```
  port: 22
```

```
  target: /usr/share/nginx/html
```

```
  source: releases/poker-game-r${DRONE_BUILD_NUMBER}.jar
```

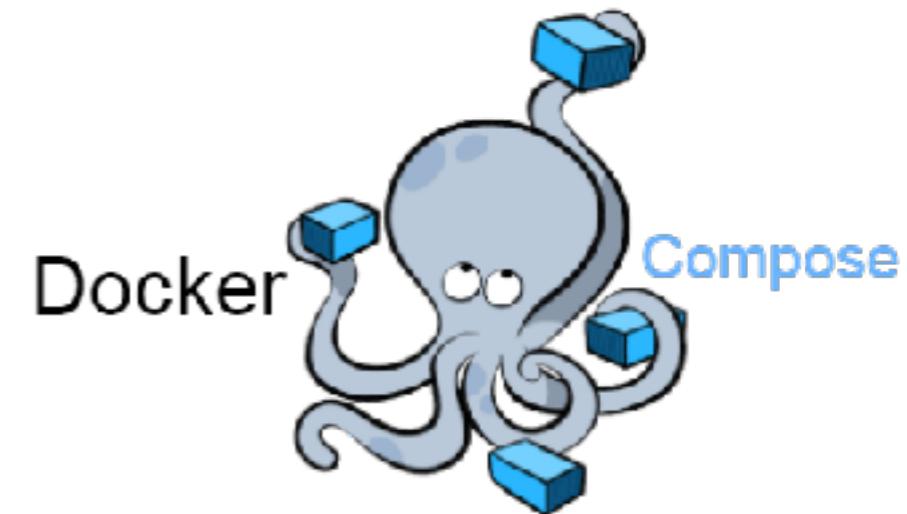
**(release to a remote web server)**

**.drone.yml**

# Deploying the stack



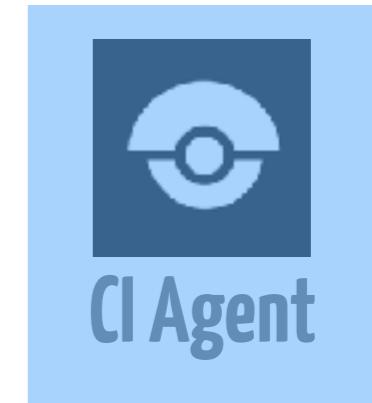
(using containers)



```
~/agile-tutorial$ cd ci
```

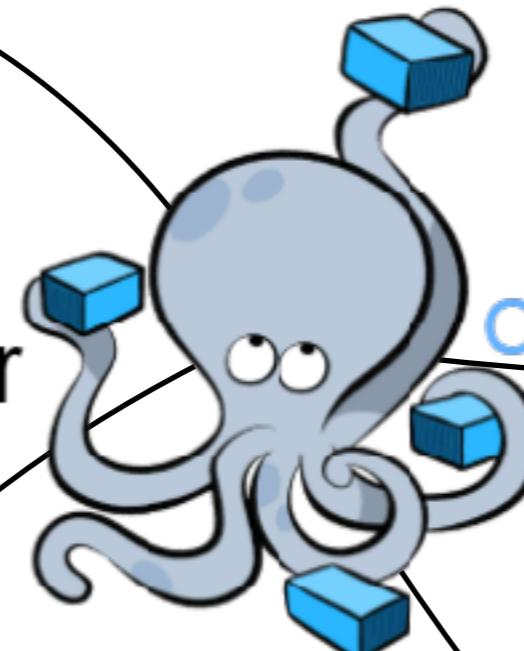
```
~/agile-tutorial/ci $ docker network create re18-network
```

```
~/agile-tutorial/ci$ docker-compose up -d
```



Docker

Compose



# docker-compose.yml



```
gitea:  
  image: gitea/gitea:1.3.2  
  restart: always  
  container_name: gitea  
  volumes:  
    - ./gitea/:/data  
  ports:  
    - "3000:3000"
```



```
drone-server:  
  image: drone/drone:0.8  
  container_name: drone-server  
  depends_on:  
    - gitea  
  ports:  
    - "8000:8000"  
  volumes:  
    - ./drone:/var/lib/drone/  
  restart: always  
  environment:  
    # Drone Config  
    - DRONE_OPEN=true  
    - DRONE_HOST=http://drone-server:8000  
    - DRONE_SECRET=agile-tutorial-re18  
    - DRONE_NETWORK=re18-network  
    # Gittea config to hook the CI to the VCS  
    - DRONE_GITEA=true  
    - DRONE_GITEA_URL=http://gitea:3000
```



# Configuring the VCS

Installation - Gitea: Git with a cup of tea

localhost:3000/install

Sébastien

### Initial configuration

If you are running Gitea inside Docker, please read the [guidelines](#) carefully before changing anything on this page.

#### Database Settings

Gitea requires MySQL, PostgreSQL, SQLite3, or TiDB.

**Database Type** \* SQLite3

**Path** \* /data/gitea/gitea.db

The file path to the SQLite3 or TiDB database.  
Please use the absolute path when you start as service.

#### General Application Settings

**Application Name** \* Gitea: Git with a cup of tea  
You can put your organization name here.

**Repository Root Path** \* /data/git/repositories  
All remote Git repositories will be saved to this directory.

**LFS Root Path** /data/gitea/lfs  
Files stored with Git LFS will be stored in this directory. Leave empty to disable LFS.

**Run User** \* git  
The user must have access to Repository Root Path and run Gitea.

**Domain** \* localhost  
This affects SSH clone URLs.



New Repository - Gites: Git with Docker

Sébastien

localhost:3000/repo/create

Dashboard Issues Pull Requests Explore

New Repository

Owner: re18

Repository Name: agile-tutorial

A good repository name is composed of short, memorable, and unique keywords.

Visibility:  This repository is Private

Description:

.gitignore: Select .gitignore templates

License: Select a license file

Readme: Default

Initialize this repository with selected files and template

Create Repository Cancel

0

Sébastien

localhost:3000/re18/agile-tutorial

Dashboard Issues Pull Requests Explore

Unwatch 1 Star 0

Code Issues Wiki Settings

Quick Guide

Clone this repository Need help cloning? Visit [Help!](#)

HTTP SSH <http://gitea:3000/re18/agile-tutorial.git>

Creating a new repository on the command line

```
touch README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin http://gitea:3000/re18/agile-tutorial.git  
git push -u origin master
```

Pushing an existing repository from the command line

```
git remote add origin http://gitea:3000/re18/agile-tutorial.git  
git push -u origin master
```

0

© Gitea Version: 1.3.2 Page: 77ms Template: 1ms

English Javascript licenses API Website Go1.9.2





# Configuring the CI

Sébastien

localhost:8000/login/form

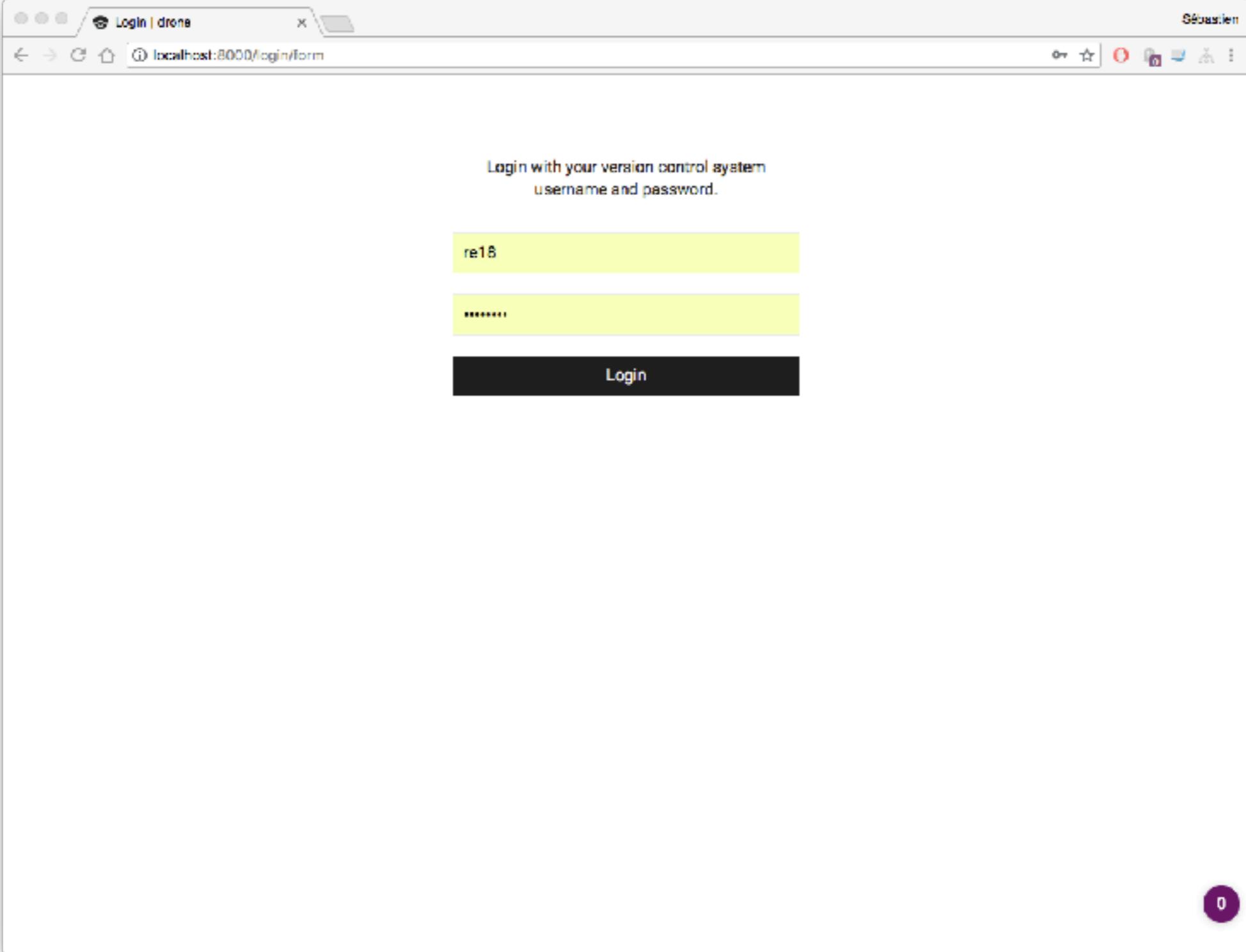
Login with your version control system  
username and password.

re18

\*\*\*\*\*

Login

0



>Welcome | drone

localhost:8000/account/repos

Sébastien

Account > Repositories

Search ...

Search ...

re18/agile-tutorial

Your build feed is empty

Successfully activated your repository

A small icon of a person wearing a helmet and holding a sword is located in the bottom right corner.

Welcome | drone      Update Webhook - Gitea: Git: x      Sébastien

localhost:3000/re18/agile-tutorial/settings/hooks/1

Webhook has been updated.

## Update Webhook

Gitea will send a POST request to the URL you specify, along with information about the event that occurred. You can also specify what data format you would like to receive upon triggering the hook (JSON, x-www-form-urlencoded, XML, etc). More information can be found in our [webhooks guide](#).

**Payload URL \***

http://drone-server:8000/hook?access\_token=eyJhbGciOiJIUzI1NiJ9.R5cG16lkpXVGJ9.eyJ0ZXh0IjoiemUxOC9hZ2lsZS10dXRvcmhhGlnR5cGUlQJob29rln0.PZAGuINTC7

**Content Type**

application/json

**Secret**

.....

**When should this webhook be triggered?**

Just the push event.

I need everything.

Let me choose what I need.

Create  
Branch, or tag created

Push  
Git push to a repository

Pull Request  
Pull request opened, closed, reopened, edited, assigned, unassigned, label updated, label cleared, or synchronized.

Repository  
Repository created or deleted

Active

Information about the event which triggered the hook will be sent as well.

**Update Webhook**   **Delete Webhook**

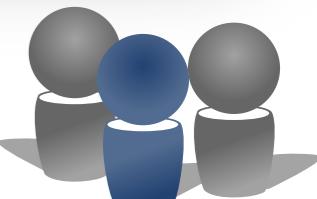
Recent Deliveries   **Test Delivery**

1

# Triggering the process



```
azrael:agile tutorial mosser$ git remote
origin
azrael:agile-tutorial mosser$ git remote add gitea http://localhost:3000/re18/agile-tutorial.git
azrael:agile-tutorial mosser$ git remote
gitea
origin
azrael:agile-tutorial mosser$ git push gitea master
Counting objects: 464, done.
Delta compression using up to 1 threads.
Compressing objects: 100% (236/236), done.
Writing objects: 100% (464/464), 1.35 MiB  14.89 MiB/s, done.
Total 464 (delta 193), reused 444 (delta 182)
remote: Resolving deltas: 100% (193/193), done.
To http://localhost:3000/re18/agile-tutorial.git
 * [new branch]    master > master
azrael:agile-tutorial mosser$
```



re18/agile-tutorial | drone X drone

localhost:8000/re18/agile-tutorial/1/3

re18 / agile-tutorial > 1

Search ...

re18/agile-tutorial ✓

organizer guide fixed ✓

① 1 minute ago  
② 50 seconds

① 1 minute ago  
② 50 seconds

efec56214 ✗

V master

clone 00:09 ✓

build 00:35 ✓

scp 00:05 ✓

**Successful**

```
1 + mvn -q clean package
2 -----
3 -----
4 T E S T S
5 -----
6 Running re.poker.GameTest
7 Tests run: 2, Failures: 0, Errors: 0, Skipped: 1, Time elapsed: 0.28 sec
8 Running re.poker.cards.SuitTest
9 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec
10 Running re.poker.cards.CardValueTest
11 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec
12 Running re.poker.cards.CardTest
13 Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec
14 Running re.poker.cards.HandTest
15 Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.057 sec
16 Running bdd.AcceptanceTest
17 Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.329 sec
18 Running bdd.RunCucumberTest
19
20 3 Scenarios (3 passed)
21 15 Steps (15 passed)
22 0m0.060s
23
24 Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.747 sec
25
26 Results :
27
28 Tests run: 20, Failures: 0, Errors: 0, Skipped: 1
29
30 + mkdir releases
31 + cp target/poker-game.jar releases/poker-game-r1.jar
exit code 0
```



Index of /releases/    X

localhost:8080/releases/    Sébastien

Index of /releases/

---

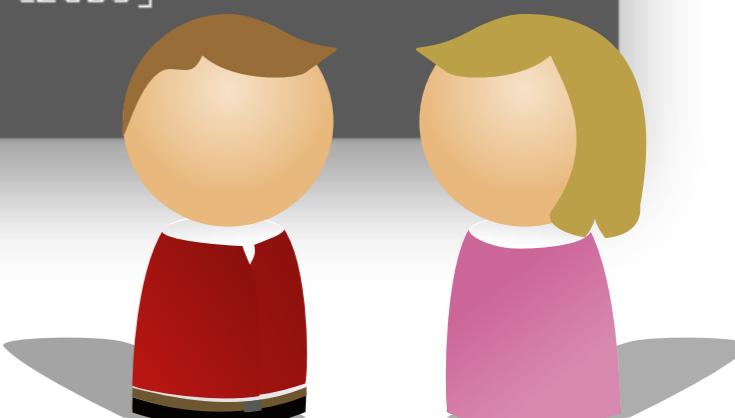
<a href="#">...</a>	17-Aug-2019 11:30	11916
<a href="#">poker-game-v1.jar</a>		

#RinseAndRepeat 0





```
[azrael:agile tutorial mosser$ curl -o product.jar http://localhost:8080/releases/poker-game-r1.jar]
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total Spent   Left  Speed
100 11816  100 11816    0     0  3846k      0 --:--:-- --:--:-- --:--:-- 3846k
[azrael:agile-tutorial mosser$ java -jar product.jar
Enter 1st player hand: AD KD QD JD TD
1st: [ACE of DIAMONDS, QUEEN of DIAMONDS, TEN of DIAMONDS, JACK of DIAMONDS, KING of DIAMONDS]
Enter 2nd player hand: 8C 7C 6C 5C 4C
2nd: [SIX of CLUBS, FOUR of CLUBS, SEVEN of CLUBS, EIGHT of CLUBS, FIVE of CLUBS]
Exception in thread "main" java.lang.UnsupportedOperationException: Cannot determine winner!
        at re.poker.Game.declareWinner(Game.java:23)
        at re.poker.Main.main(Main.java:22)
[azrael:agile-tutorial mosser$]
[azrael:agile-tutorial mosser$ curl -o product.jar http://localhost:8080/releases/poker-game-r2.jar]
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total Spent   Left  Speed
100 12420  100 12420    0     0  4042k      0 --:--:-- --:--:-- --:--:-- 4042k
[azrael:agile-tutorial mosser$ java -jar product.jar
Enter 1st player hand: AD KD QD JD TD
1st: [ACE of DIAMONDS, QUEEN of DIAMONDS, TEN of DIAMONDS, JACK of DIAMONDS, KING of DIAMONDS]
Highest card: ACE of DIAMONDS
Enter 2nd player hand: 8C 7C 6C 5C 4C
2nd: [SIX of CLUBS, FOUR of CLUBS, SEVEN of CLUBS, EIGHT of CLUBS, FIVE of CLUBS]
And the winner is: 1st
```



# When people **don't** see the value of CI

<http://itsadeliverything.com/five-things-to-do-when-people-dont-to-see-the-value-of-automation#more-5966>

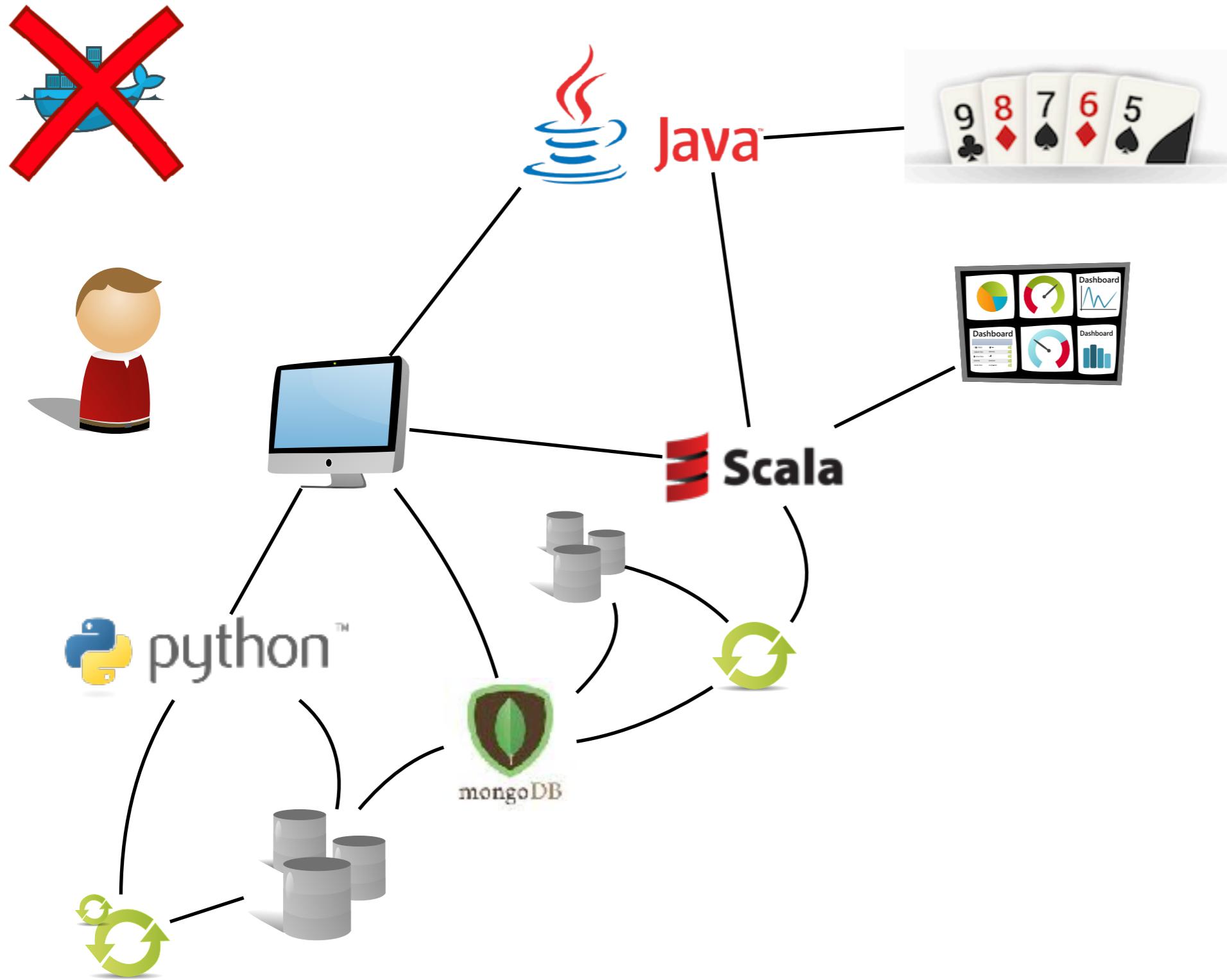
- Get an technical Ally
- Build monitor
- Pair programming
- Training
- Change the team

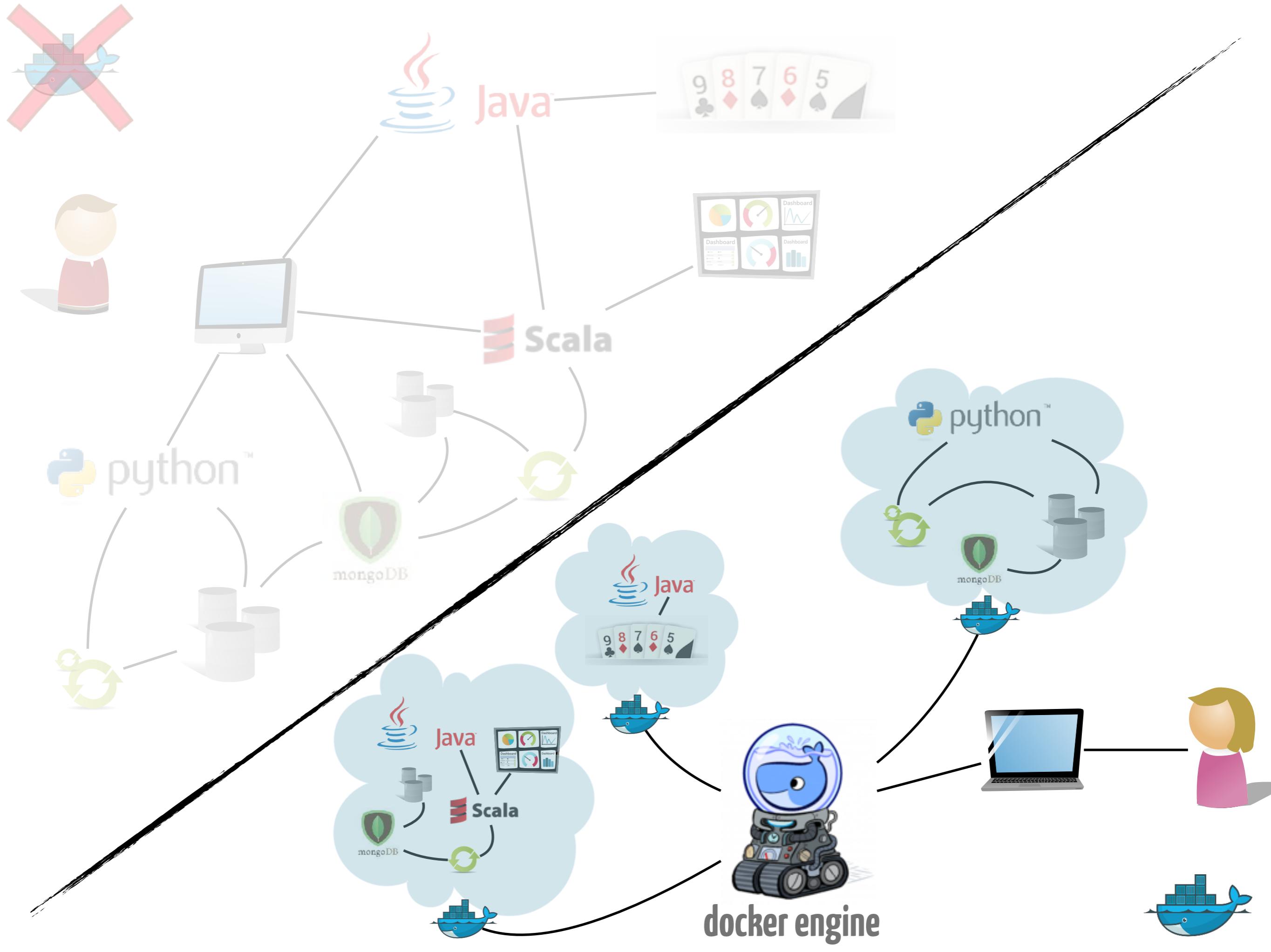




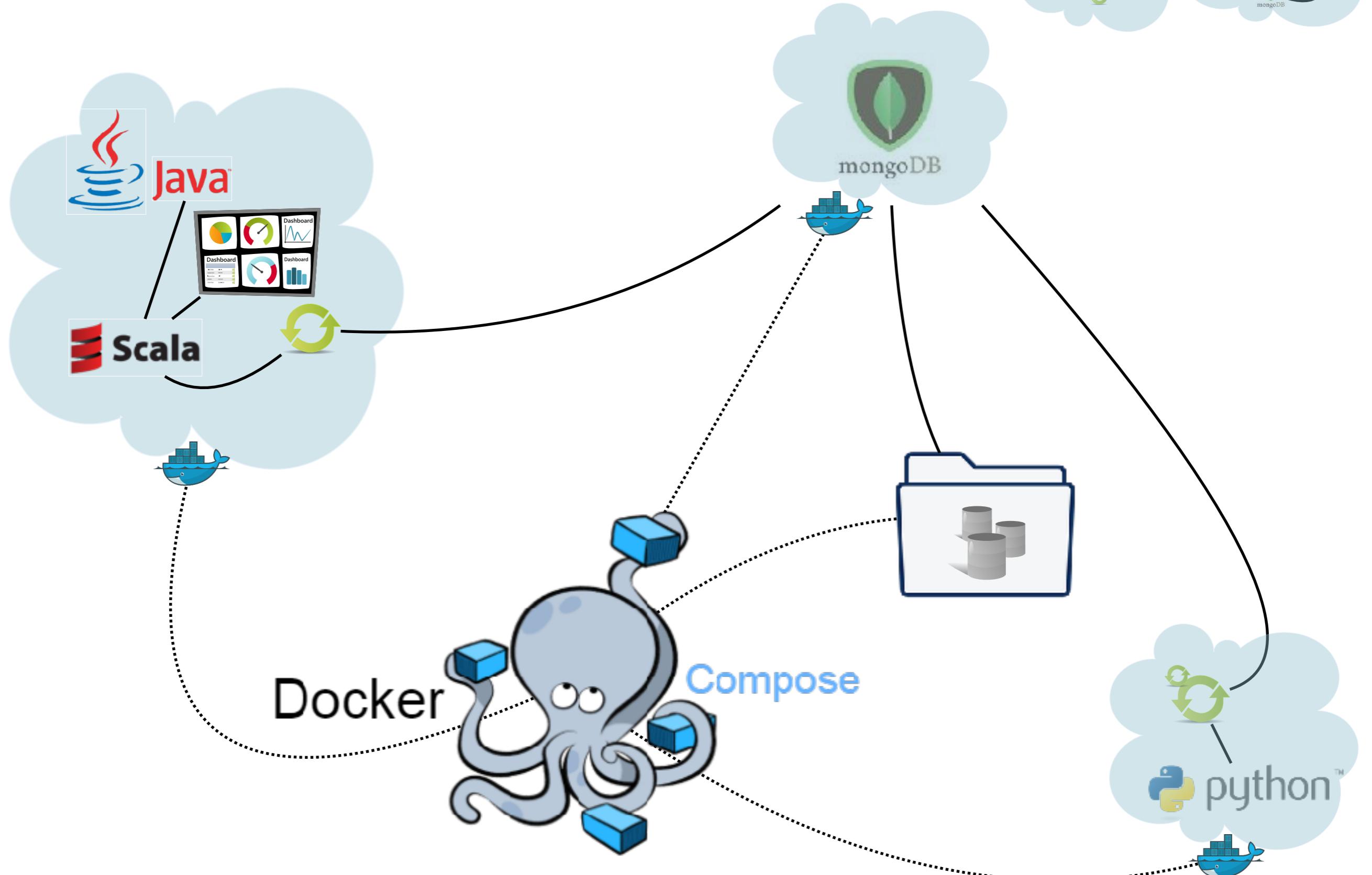
# From CI to Continuous Deployment

Let's use Docker to deploy!



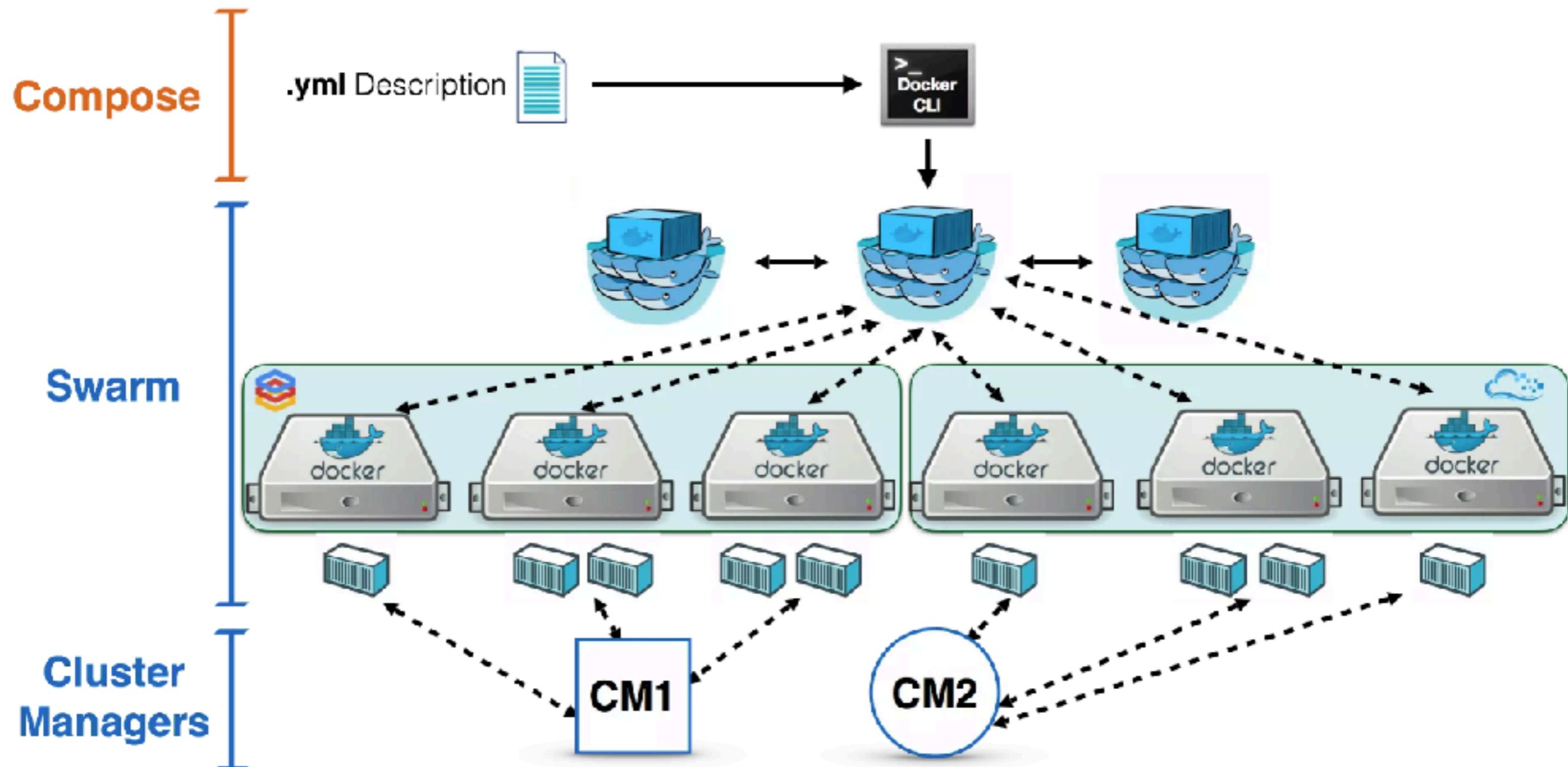


# Reusing / composing images



Remember `ci/docker-compose.yml`?

# Swarm for scalability



Out of scope for this tutorial

The screenshot shows a GitHub browser window with the URL <https://github.com/docker/labs/blob/master/README.md>. The page content is as follows:

# Docker Tutorials and Labs

At this time we are not actively adding labs to this repository. Our focus is on [training.play-with-docker.com](https://training.play-with-docker.com) where new lab and workshop oriented content is being added. We welcome fixes to existing content. For any new content you wish to contribute, please use this repository:<https://github.com/play-with-docker/play-with-docker.github.io>.

This repo contains Docker labs and tutorials authored both by Docker, and by members of the community. We welcome contributions and want to grow the repo.

**Docker tutorials:**

- [Docker for beginners](#)
- [Docker Swarm Mode](#)
- [Configuring developer tools and programming languages](#)
  - [Java](#)
    - [Live Debugging Java with Docker](#)
    - [Docker for Java Developers](#)
  - [Node.js](#)
    - [Live Debugging a Node.js application in Docker](#)
    - [Dockerizing a Node.js application](#)
- [Docker for ASP.NET and Windows containers](#)
- [Building a 12 Factor app with Docker](#)
- [Docker Security](#)
- [Docker Networking](#)
- [Hands-on Labs from DockerCon US 2017](#)

**https://github.com/docker/labs/blob/master/README.md**

```
# Requiring a Java 8 JRE, based on Alpine Linux (light distribution)
FROM openjdk:8-jre-alpine

# Who is in charge of the deployment image
LABEL maintainer="Sébastien Mosser (mosser@i3s.unice.fr)"

# Copying the built JAR inside the image
COPY ./target/poker-game.jar ./poker-game.jar

# Starts the poker-game executable
CMD ["java", "-jar", "./poker-game.jar"]
```

## Dockerfile



```
azrael:agile-tutorial mosser$ docker build -t poker-game .
Sending build context to Docker daemon 2.879MB
Step 1/4 : FROM openjdk:8-jre-alpine
--> ccfb0c83b2fe
Step 2/4 : LABEL maintainer="Sébastien Mosser (mosser@i3s.unice.fr)"
 > Using cache
--> 46e18da81d63
Step 3/4 : COPY ./target/poker-game.jar ./poker-game.jar
--> e50d5f8e6171
Step 4/4 : CMD ["java", "-jar", "./poker-game.jar"]
--> Running in 5a536cbf6f49
Removing intermediate container 5a536cbf6f49
 > e43d4f2ec8c2
Successfully built e43d4f2ec8c2
Successfully tagged poker-game:latest
azrael:agile-tutorial mosser$
```

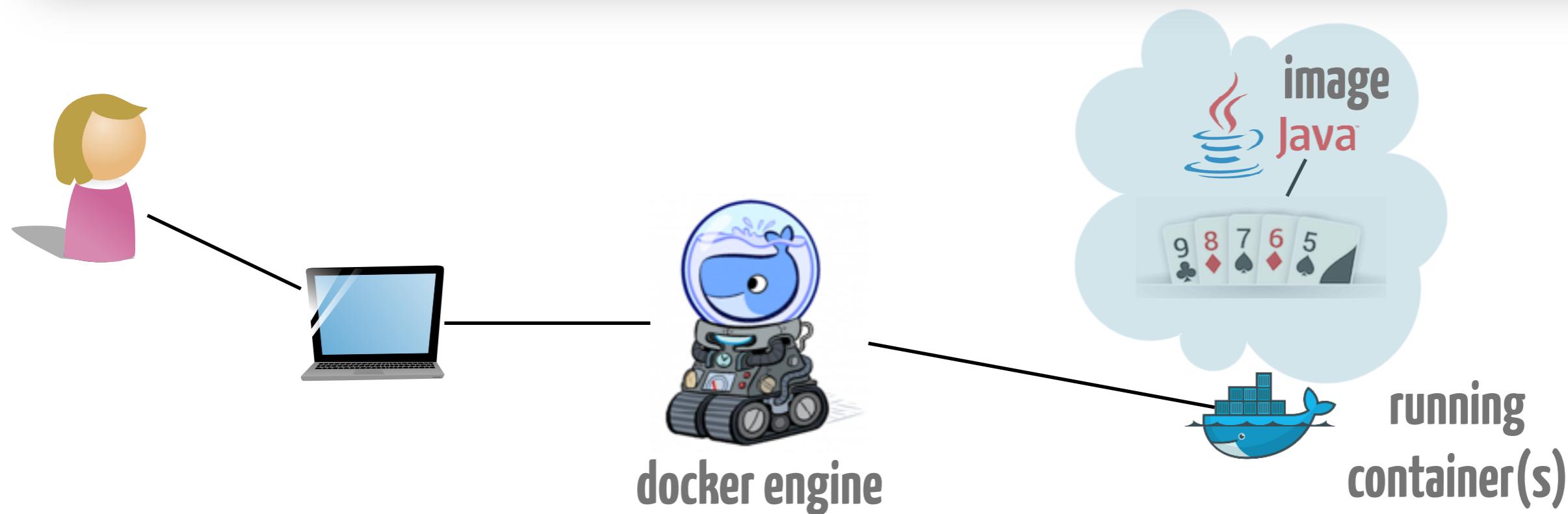
## Image

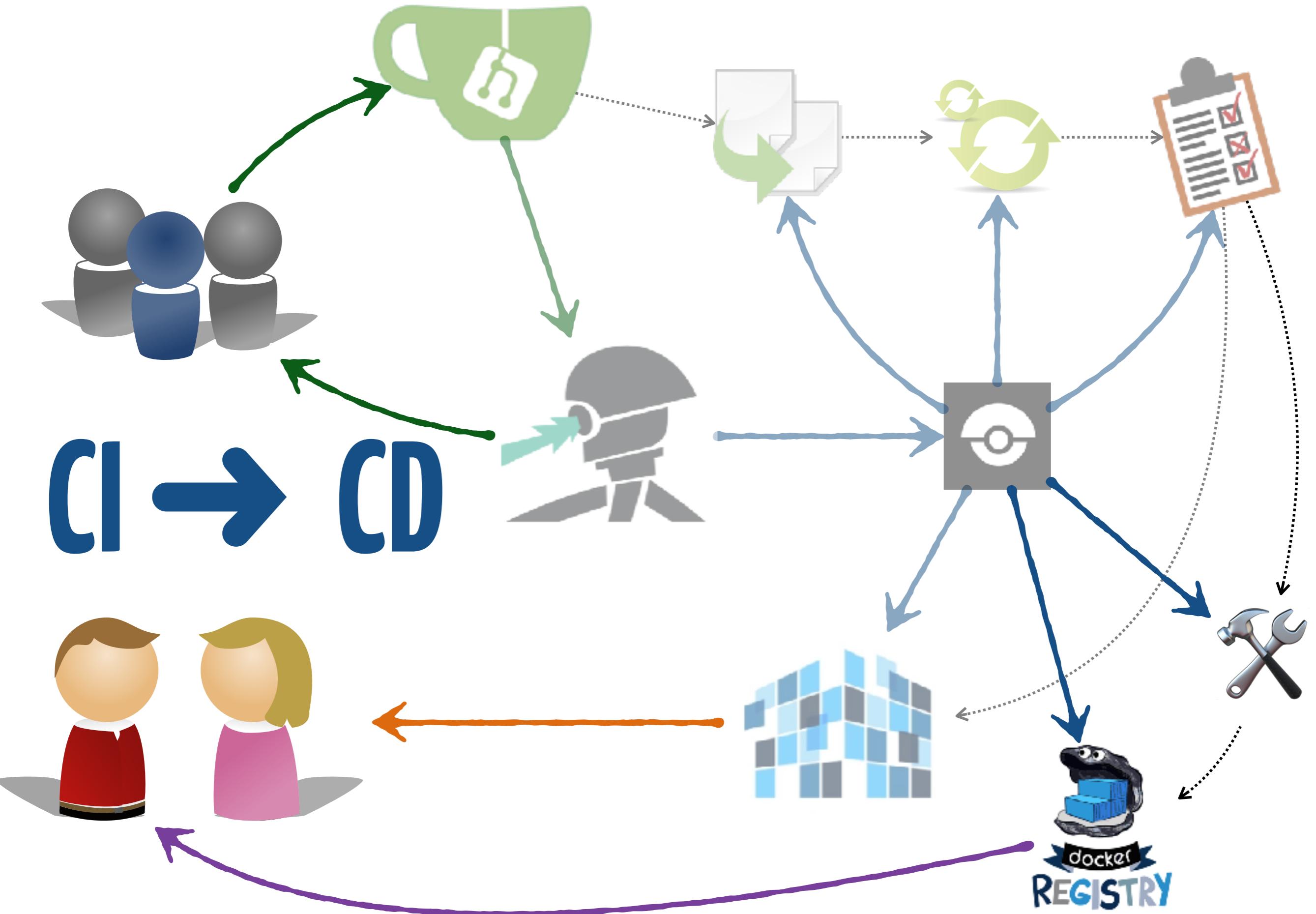


# Start a container



```
azrael:agile-tutorial mosser$ docker run -it --rm poker-game:latest
[Enter 1st player hand: AD KD QD JD TD
1st: [TEN of DIAMONDS, QUEEN of DIAMONDS, KING of DIAMONDS, JACK of DIAMONDS, ACE of DIAMONDS]
Highest card: ACE of DIAMONDS
[Enter 2nd player hand: 8C 7C 6C 5D 4C
2nd: [SIX of CLUBS, FIVE of DIAMONDS, EIGHT of CLUBS, FOUR of CLUBS, SEVEN of CLUBS]
And the winner is: 1st
azrael:agile-tutorial mosser$
```







```
# Docker registry
registry:
  image: registry:2
  container_name: registry
  restart: always
  depends_on:
    - drone-server
  ports:
    - "5000:5000"
  volumes:
    - "./registry:/var/lib/registry"
```

```
azrael:ci mosser$ docker-compose down
Stopping release-manager ... done
Stopping drone agent ... done
Stopping drone-server ... done
Stopping gitea ... done
Removing release manager ... done
Removing drone-agent ... done
Removing drone server ... done
Removing gitea ... done
Network re18-network is external, skipping
azrael:ci mosser$ docker compose up -d
Pulling registry (registry:2)...
2: Pulling from library/registry
406411dc821c: Pull complete
c12c92d1cb6a2: Pull complete
4fbc9b6835cc: Pull complete
705973b0t65f: Pull complete
3968//1a/cfa: Pull complete
Digest: sha256:51bb55f23cf7c25ac9b8313b139a8dd45baa832943c8ad8f7da2ddad6355b3c8
Status: Downloaded newer image for registry:2
Creating gitea ... done
Creating drone server ... done
Creating drone-agent ... done
Creating release manager ... done
Creating registry ... done
azrael:ci mosser$
```

# Deploying the registry

Google Agenda - X docker registry log X docker-registry.svc X faire tomber le mur X New Tab X re18/agile-tutorial X Gebäude

localhost:3000/re18/agile-tutorial/\_edit/master/.drone.yml

Dashboard Issues Pull Requests Explore

## re18 / agile-tutorial

Code Issues Pull Requests Releases Wiki Activity Settings

agile-tutorial / .drone.yml ⚡ or cancel

Edit file Preview Changes

```

1 pipeline:
2
3   # Building the Poker Game using Raven
4   build:
5     image: maven:3.5-alpine
6     commands:
7       - mvn -q clean package
8       - mkdir releases
9       - cp target/poker-game.jar releases/poker-game-r${DRONE_BUILD_NUMBER}.jar
10
11   # Publishing the last release in the release manager
12   scp:
13     group: releasing
14     image: appleboy/drone-scp
15     host: release-manager
16     username: root # Don't do this at home (demonstration purpose only)
17     password: re18 # Don't do this at home (use secret instead)
18     port: 22
19     target: /usr/share/nginx/html
20     source: releases/poker-game-r${DRONE_BUILD_NUMBER}.jar
21
22   # Building and publishing the Docker image to the local registry
23   docker:
24     group: releasing
25     image: plugins/docker
26     repo: registry:5000/poker-game
27     registry: registry:5000
28     insecure: true # Don't do this at home (demonstration purpose only, deactivating certificates validation)
29

```

 Commit Changes

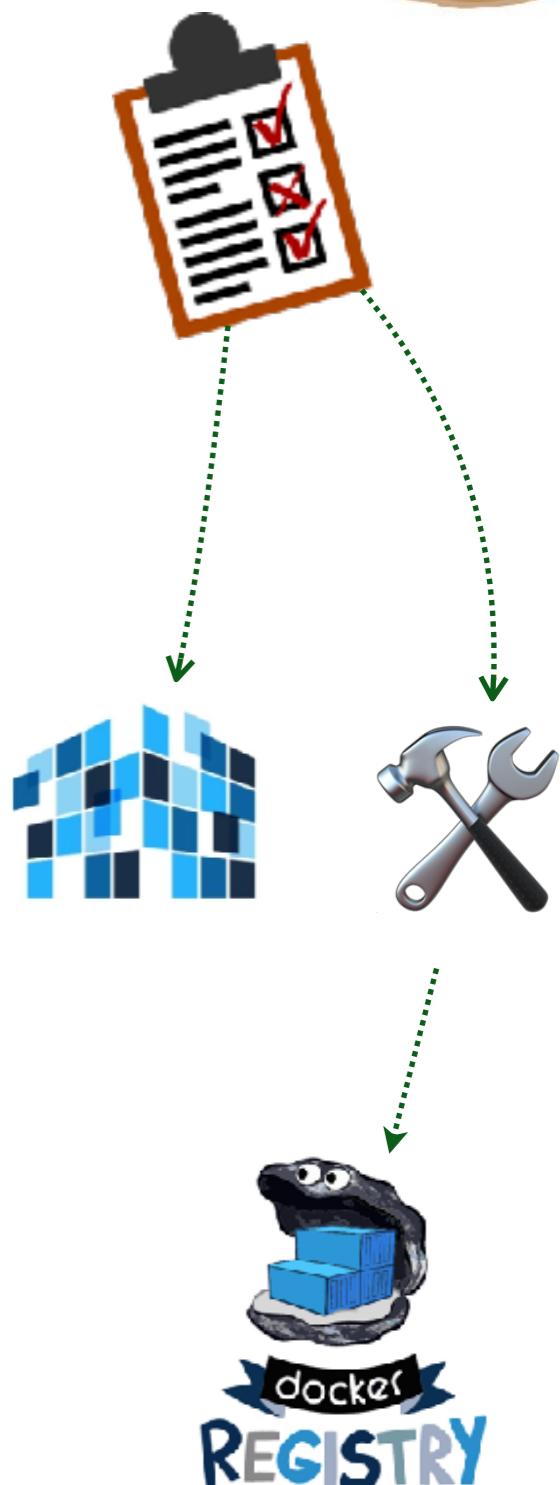
Update .drone.yml

Add an optional extended description...

Commit directly to the master branch.

Create a new branch for this commit and start a pull request.

Commit Changes Cancel



.drone.yml

rc18/agile-tutorial | drone X Sébastien

localhost:8000/re18/agile-tutorial/4

re18 / agile-tutorial > 4

Search ...

re18/agile-tutorial

⌚ 34 seconds ago ⌚ 34 seconds ago

Running

Update '.drone.yml'

⌚ 34 seconds ago ⌚ 34 seconds ago

8fdc0f11cf master

clone 00:02 ✓

build 00:25 ✓

scp 00:03 ✓

docker 00:05 ⌚

1 + git init  
2 Initialized empty Git repository in /drone/src/gitea/re18/agile-tutorial/.git/  
3 + git remote add origin http://gitea:3000/re18/agile-tutorial.git  
4 + git fetch --no-tags origin +refs/heads/master:  
5 From http://gitea:3000/re18/agile-tutorial  
6 \* branch master -> FETCH\_HEAD  
7 \* [new branch] master -> origin/master  
8 + git reset --hard -q  
8fdc0f11cfbbda179d23c1c9fae1acefde9aa4f4  
9 + git submodule update --init --recursive  
exit code 0

# Pulling the image



```
[azrael:agile-tutorial mosser$ docker pull localhost:5000/poker-game:latest
latest: Pulling from poker-game
8c3ba11ec2a2: Already exists
311ad0da4533: Already exists
391a6a6b3651: Already exists
95903d23a647: Pull complete
Digest: sha256:e22fd617d01a5fa3a0e387f7ab94dfae7a6d49a7fa15cb75b0e2263615b58b09
Status: Downloaded newer image for localhost:5000/poker-game:latest
[azrael:agile-tutorial mosser$ docker images | grep poker
poker game          latest      648e5d50c853    45 seconds ago   83MB
localhost:5000/poker-game  latest      e13d4f2ec8c2    27 minutes ago   83MB
[azrael:agile-tutorial mosser$ docker run -it --rm localhost:5000/poker-game:latest
[Enter 1st player hand: AD KD QD JD TD
1st: [TEN of DIAMONDS, QUEEN of DIAMONDS, KING of DIAMONDS, JACK of DIAMONDS, ACE of DIAMONDS]
Highest card: ACE of DIAMONDS
[Enter 2nd player hand: 8C 7C 6C 5C 4C
2nd: [SIX of CLUBS, EIGHT of CLUBS, FIVE of CLUBS, FOUR of CLUBS, SEVEN of CLUBS]
And the winner is: 1st
azrael:agile-tutorial mosser$
```



# Closing session

time to say goodbye!

# What we did together

- **Opening session**

1. Identifying Personae
2. Writing Epics & Stories
3. Implementing & Testing stories



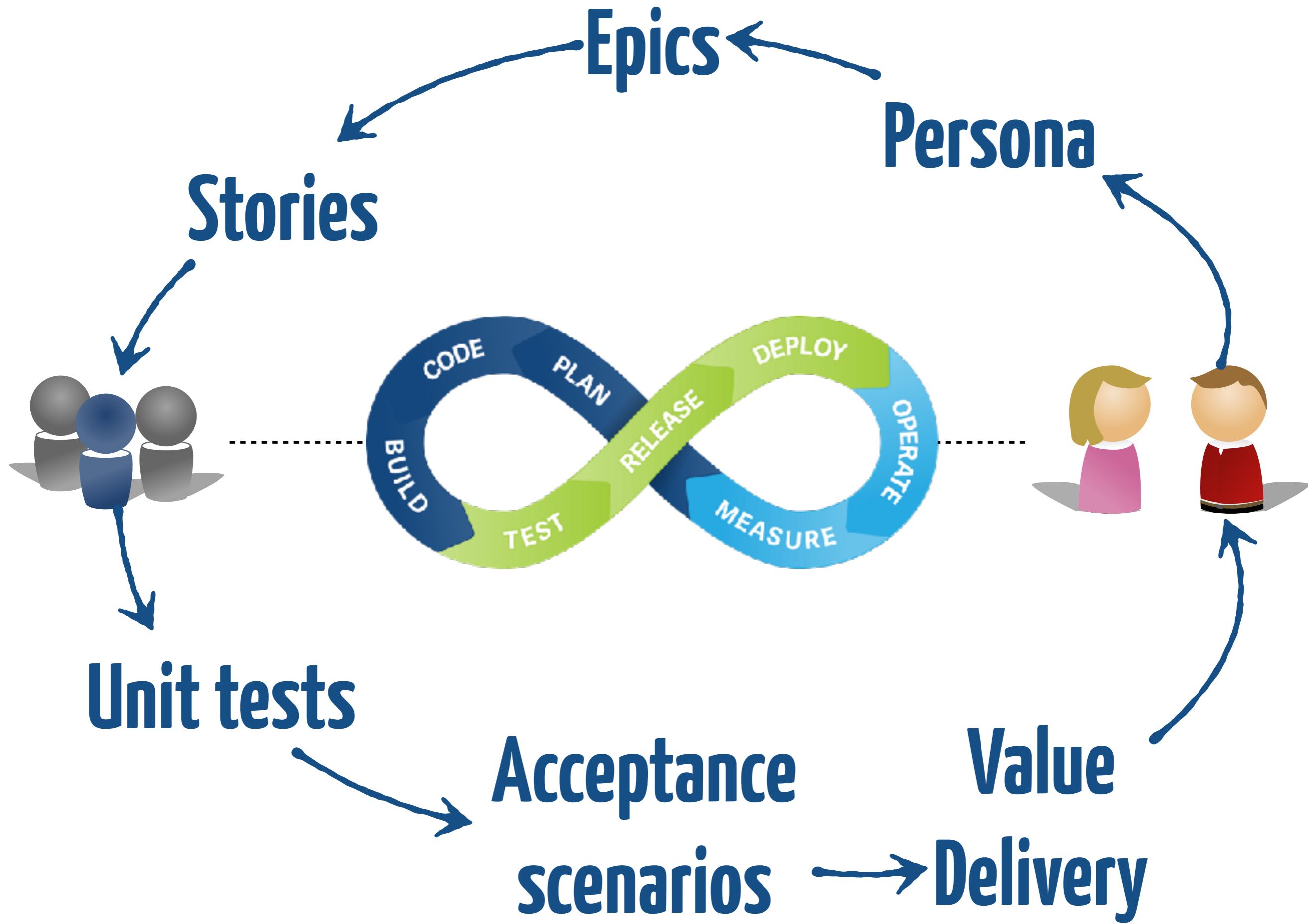
- **BREAK**

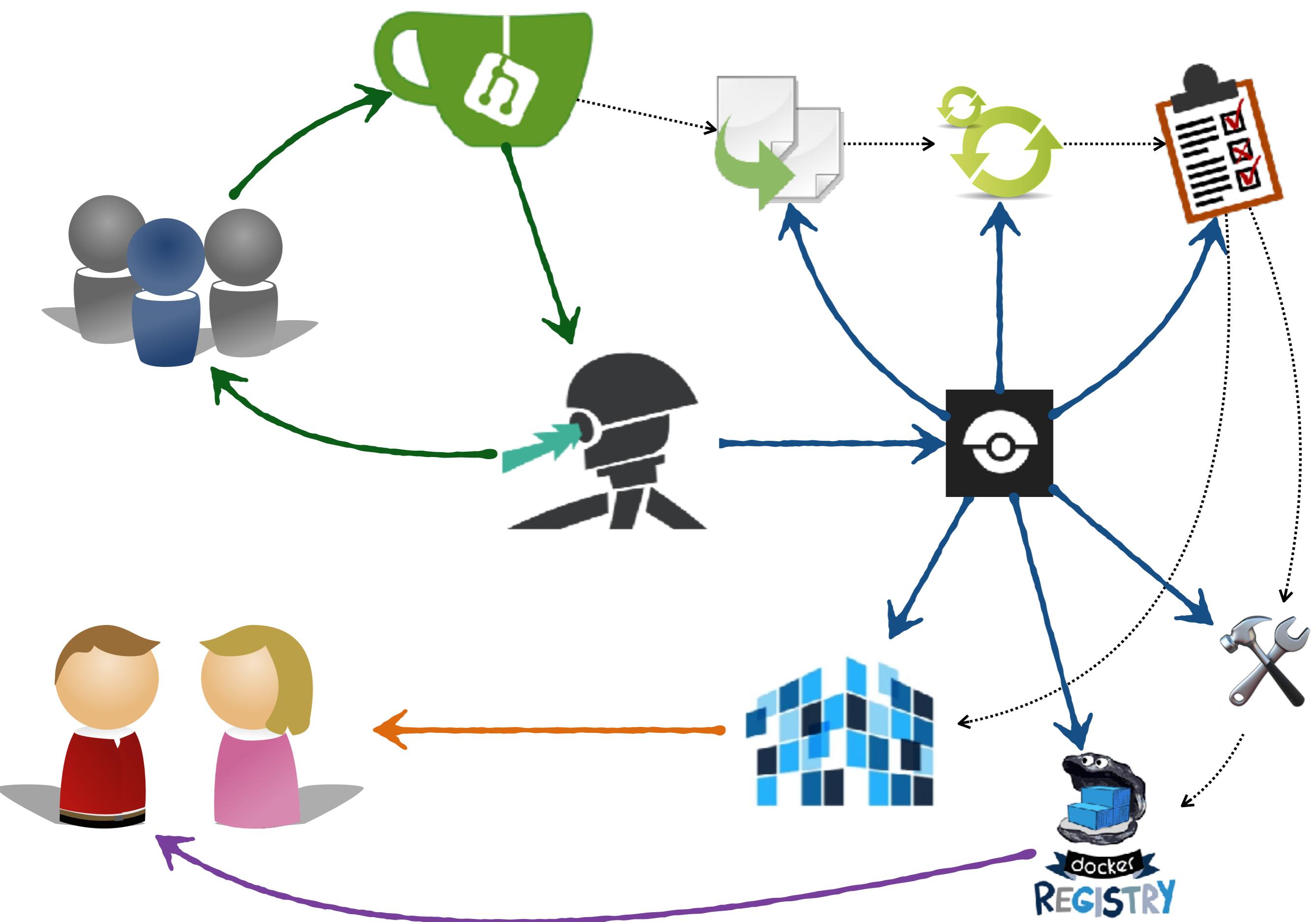
4. Automating Acceptance Scenarios
5. Deploying a Cont. Int. stack
6. From CI to Cont. Deployment



- **Closing session**









**What did you  
learn today?**

**What was difficult?**

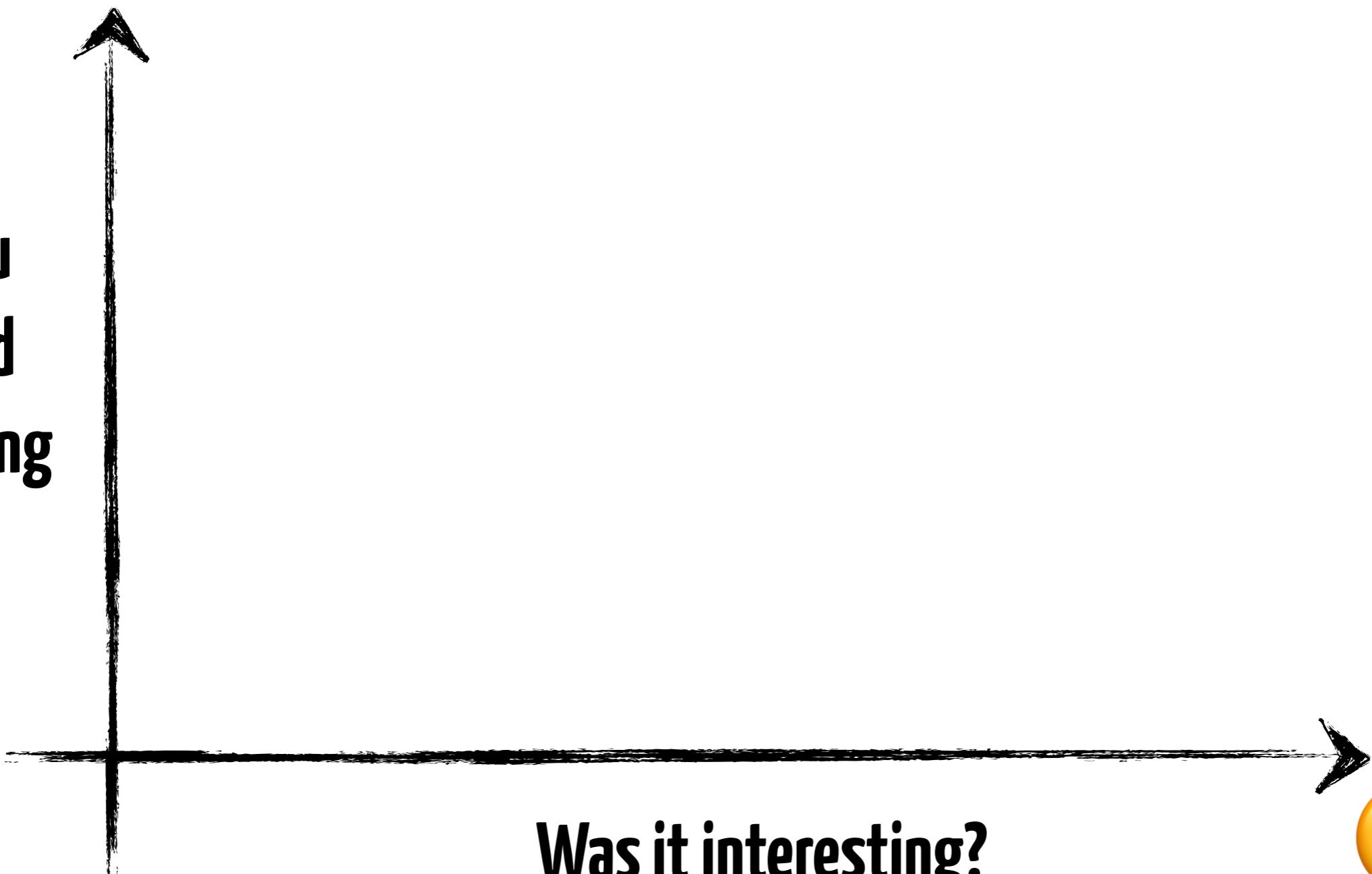
Can you adopt such an  
approach in your lab?

# Do not forget to fill the RoTI board!

---



Did you  
learned  
something  
new?



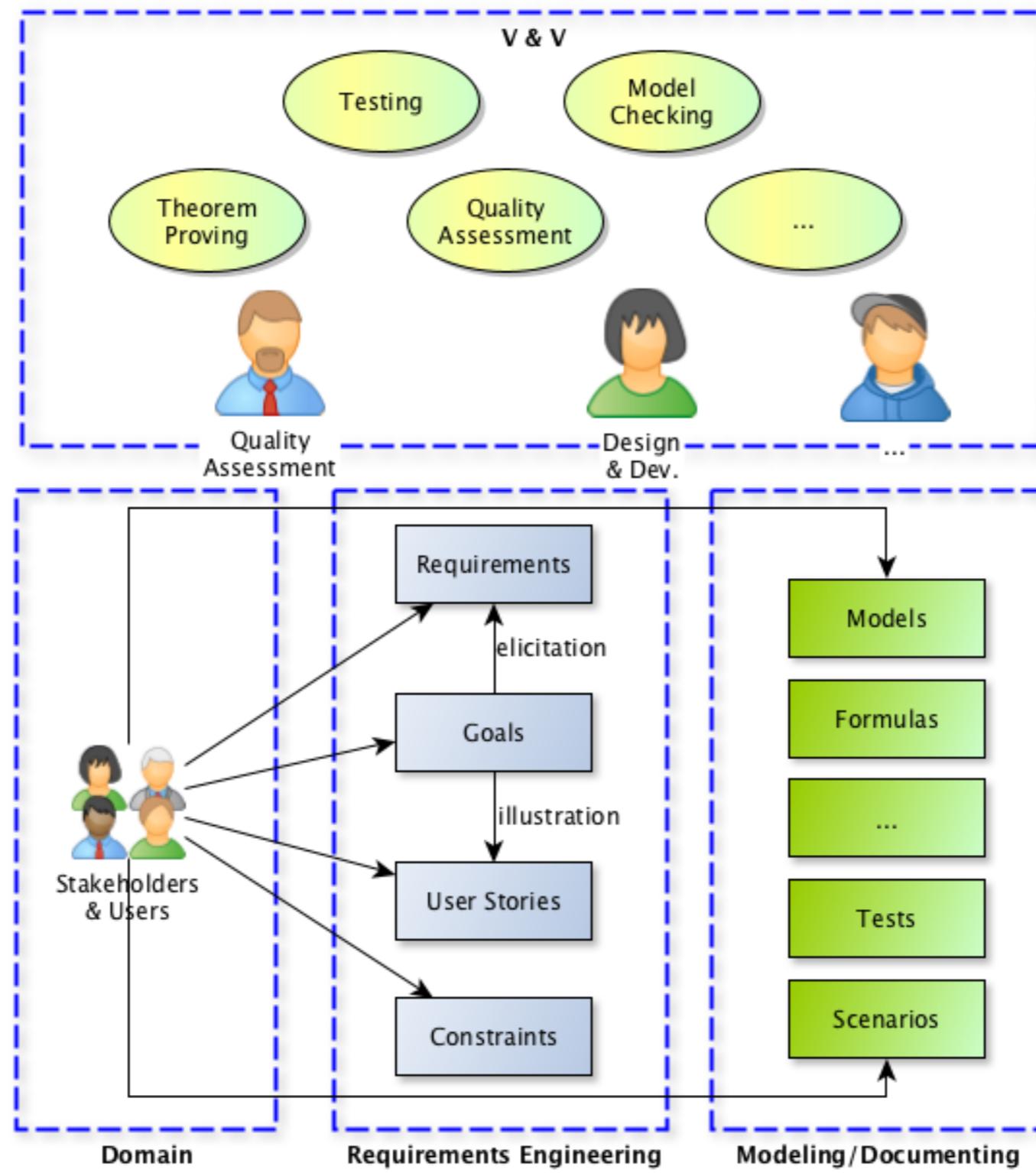
Was it interesting?



OLD

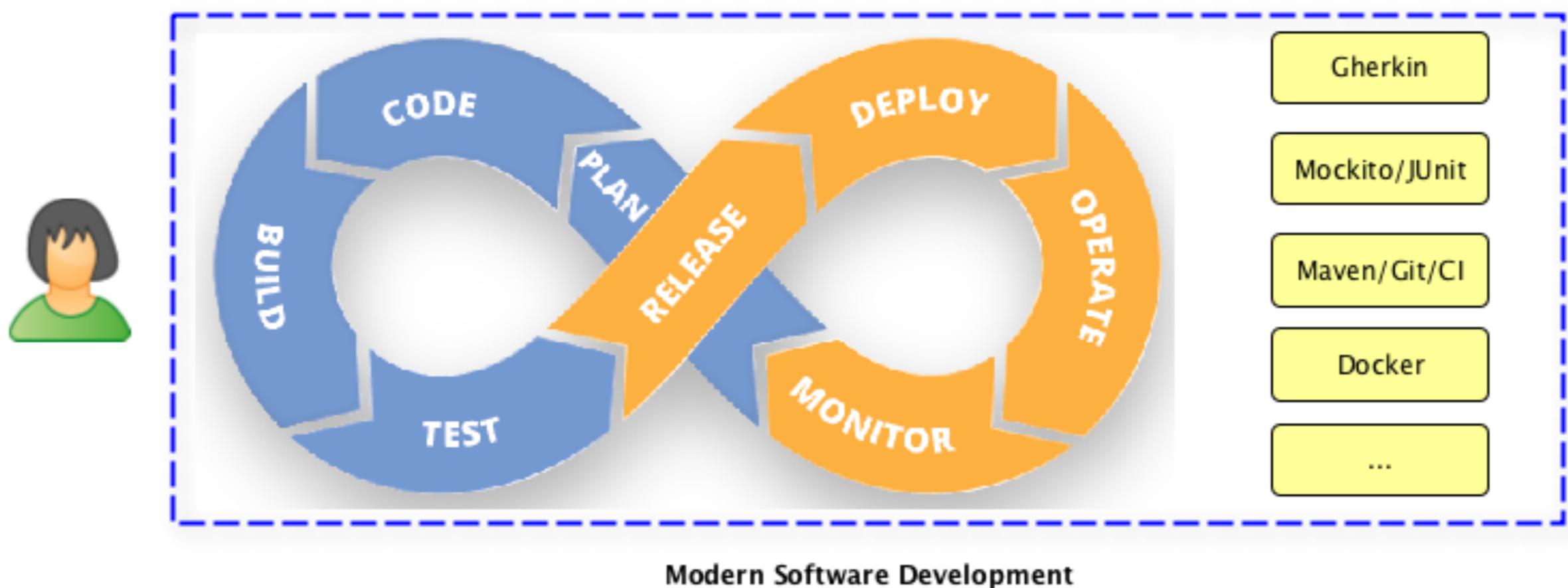
**So, what's about  
this tutorial?**

# Big Picture

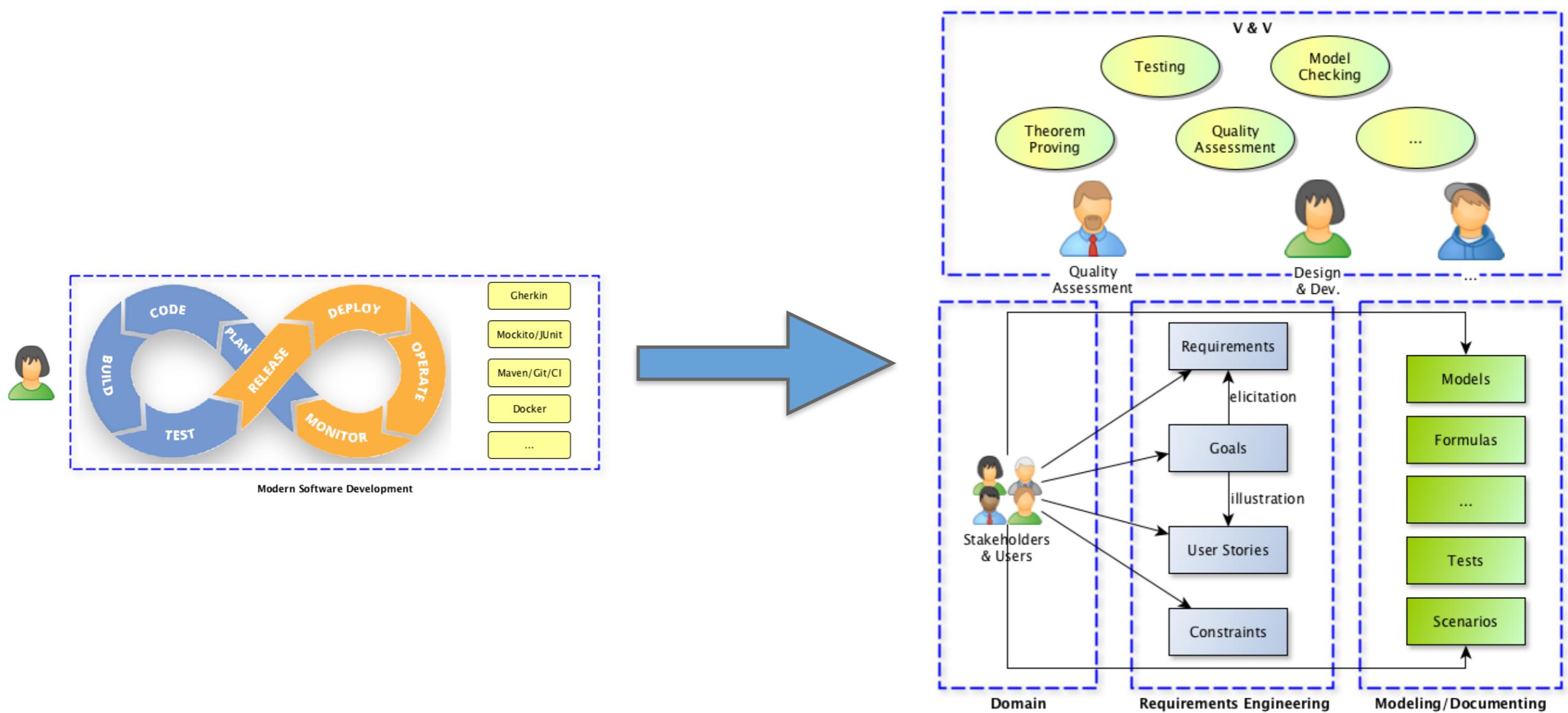


# Modern software development: **DEVOPS**

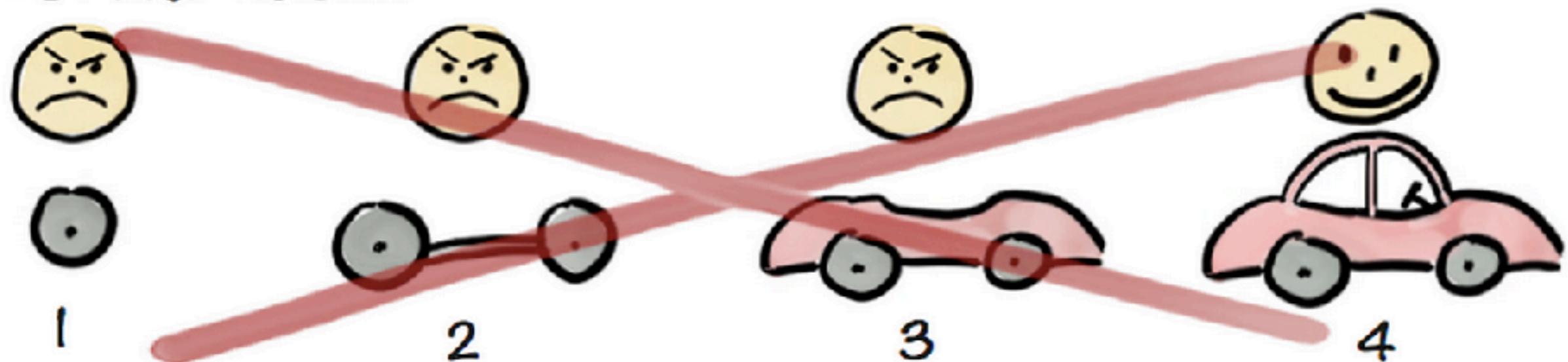
---



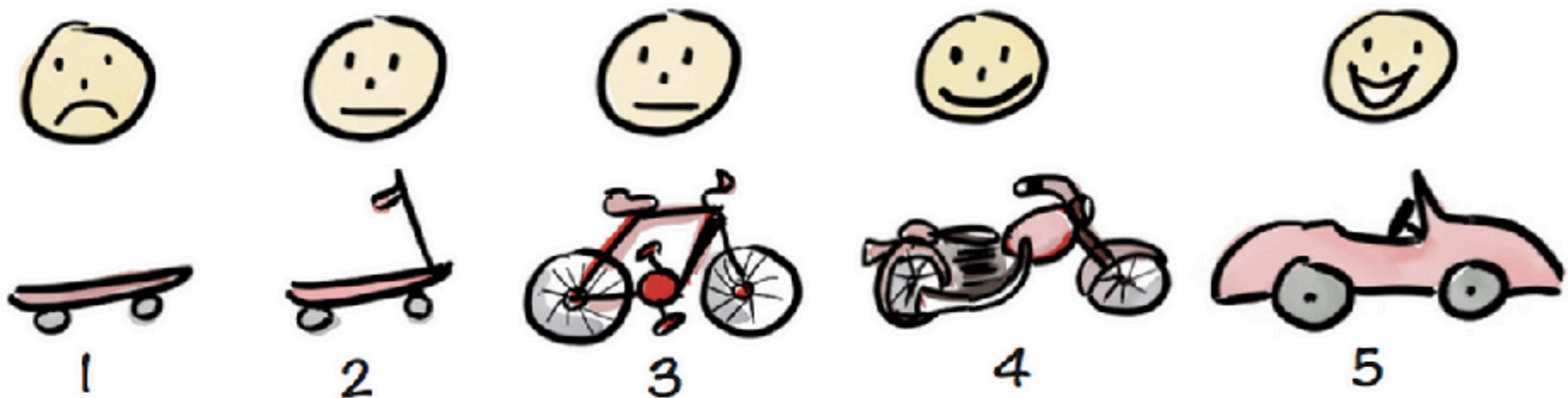
# Apply good practice to RE



**Not like this....**



**Like this!**



by Henrik Kniberg

Not like this....



1



3



4

Like this



1



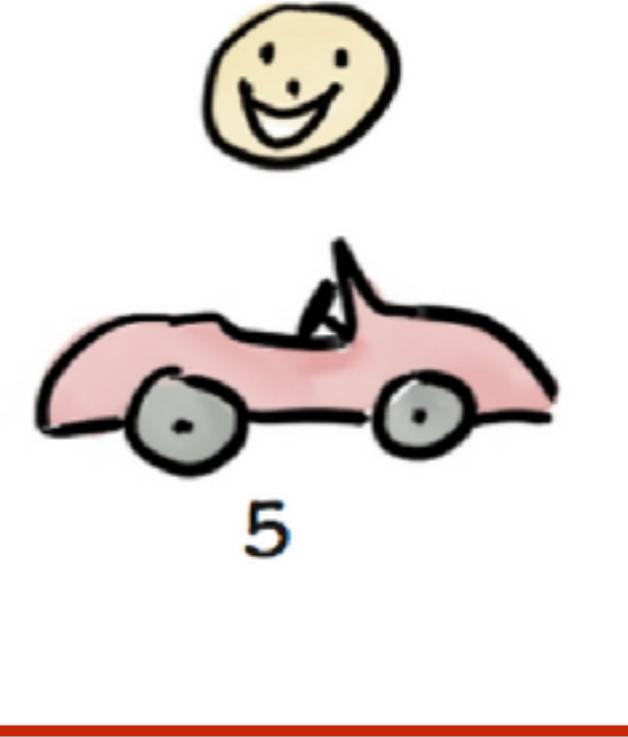
2



3

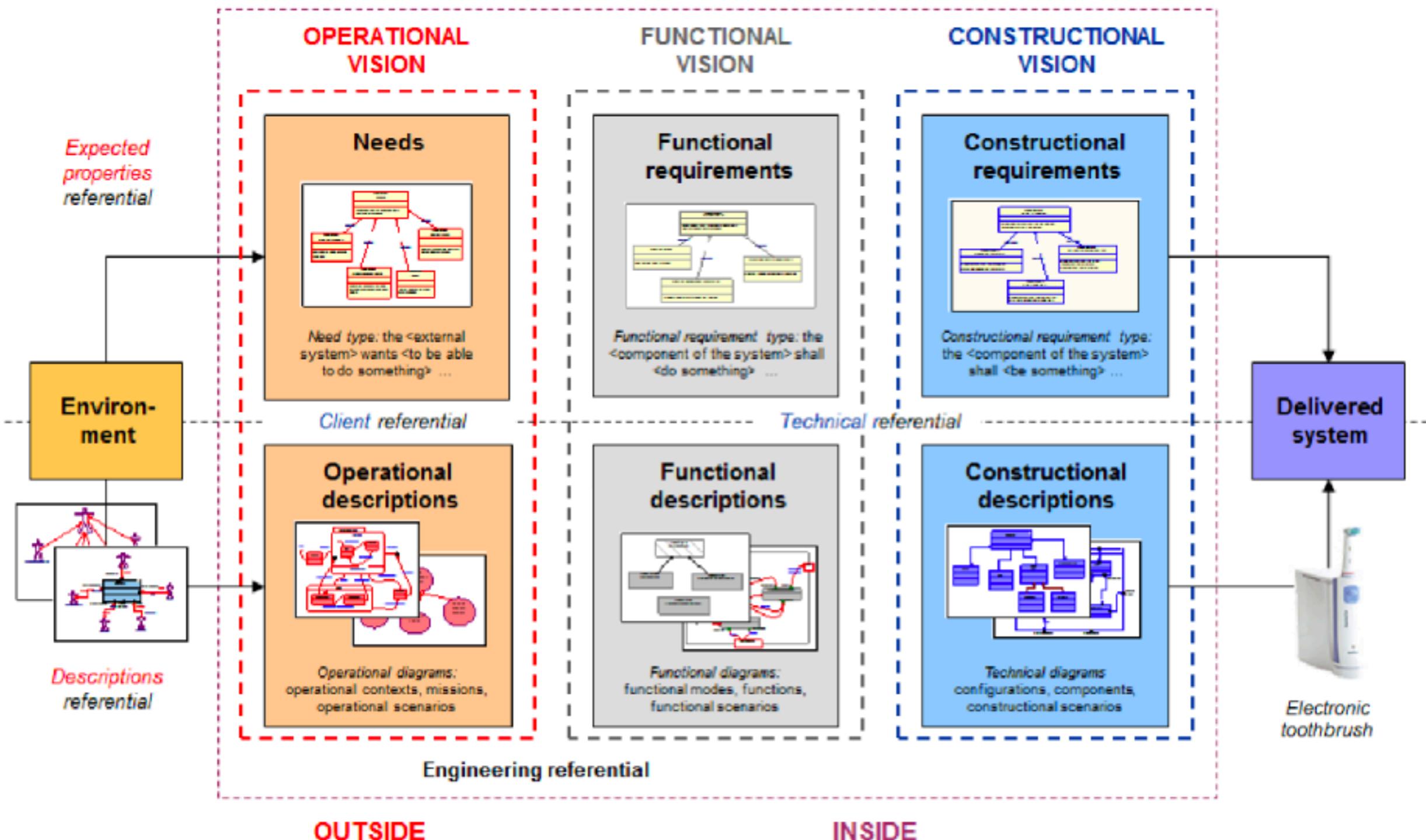


4

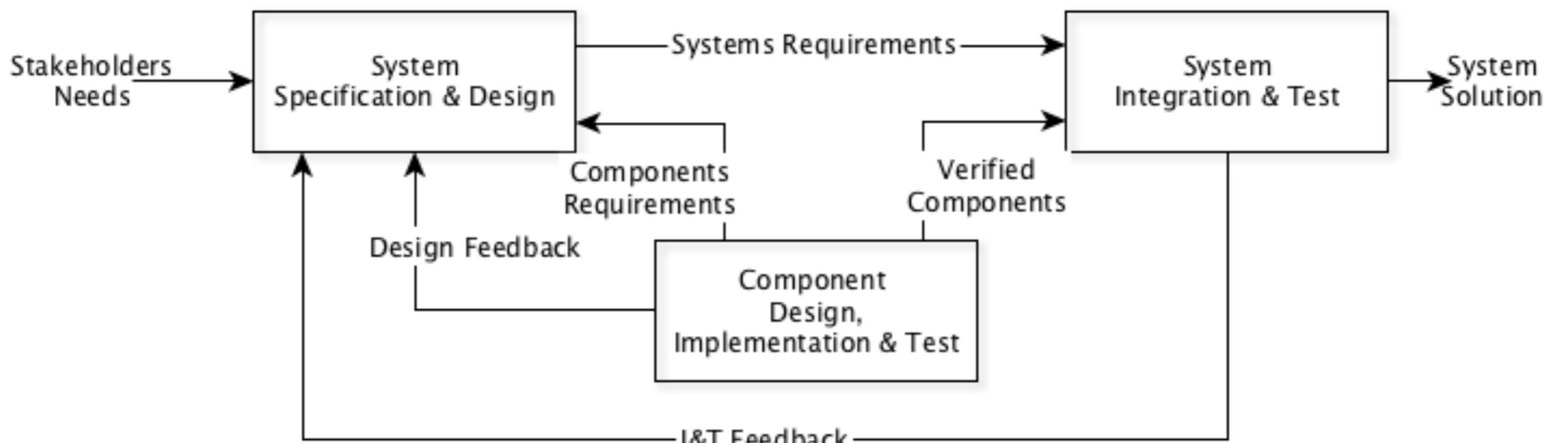


5

# Expected properties vs. Descriptions

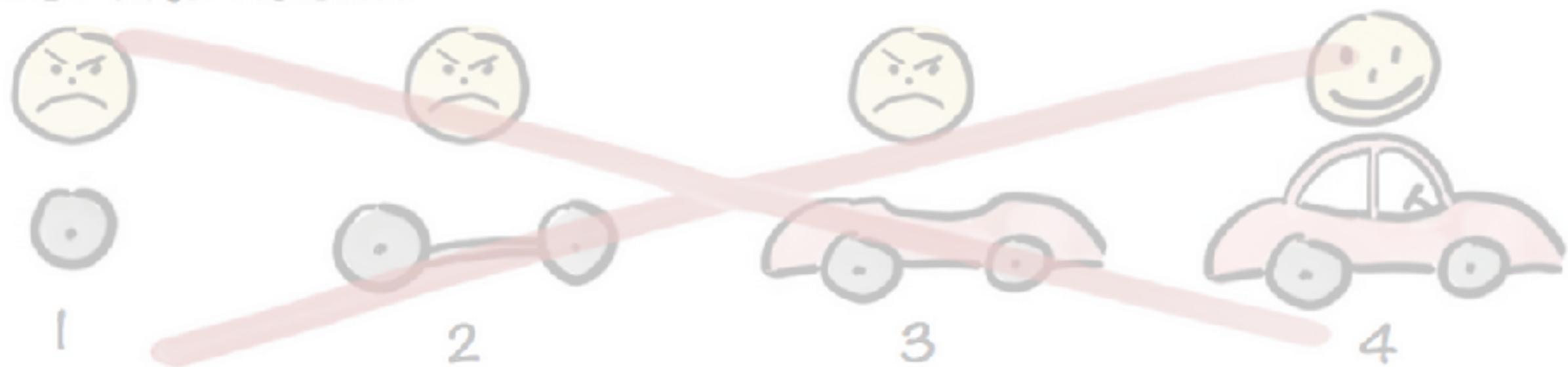


# Systems Engineering (simplified) Process

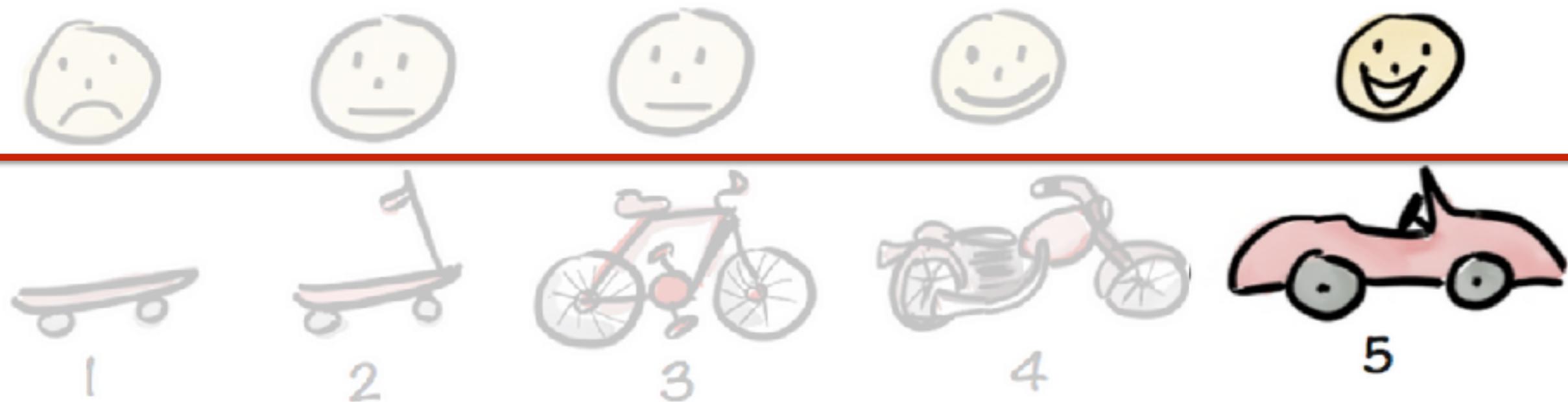


[Friedenthal 2016]

Not like this....



Like this!



by Henrik Kniberg

Not like this....



1

# Requirements Validation



3



4

Like this!



1



2



3



4



5

# Example of ArgoSim Stimulus



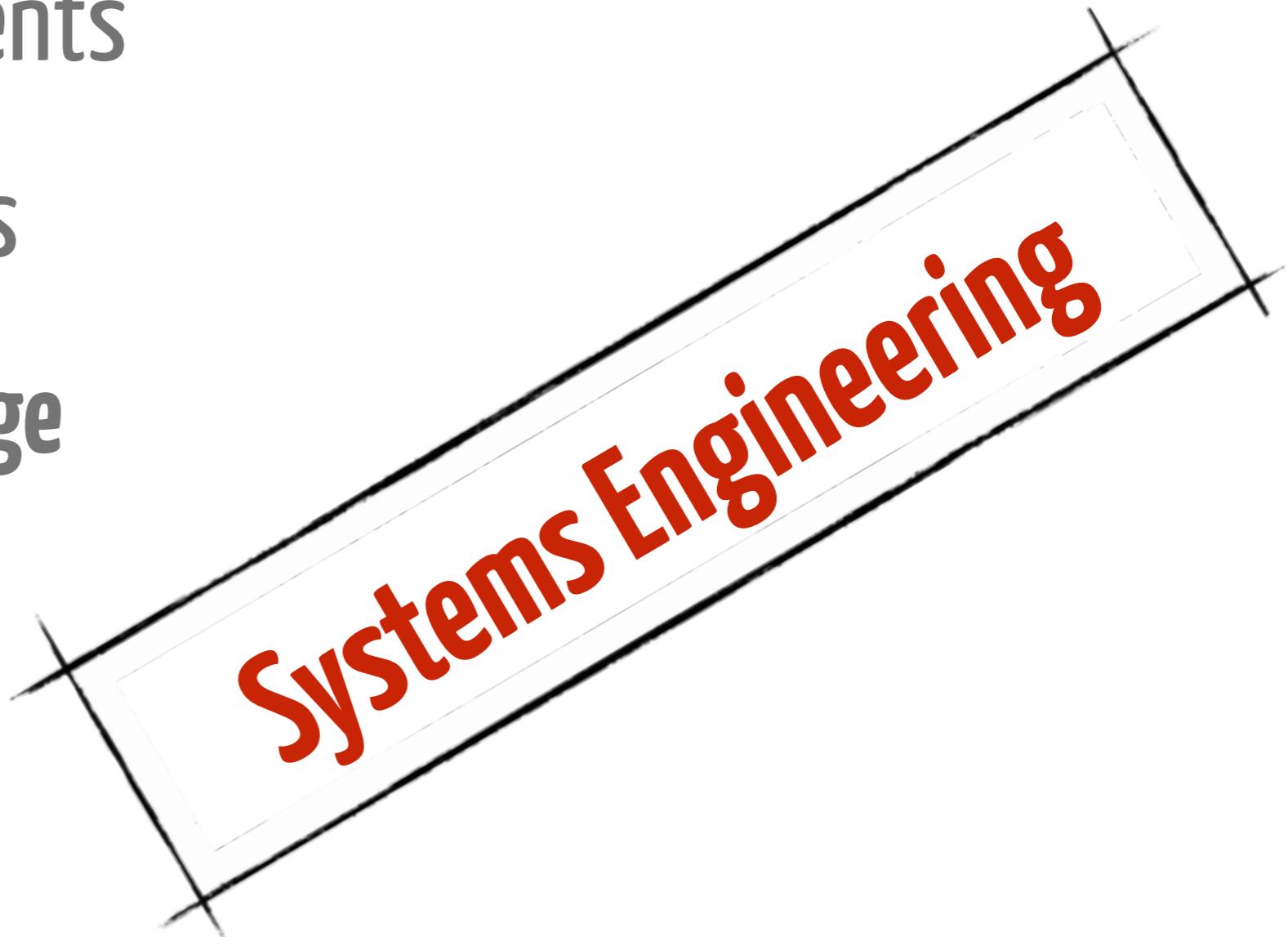
<https://www.argosim.com/home/product-overview/>

- **Conflicting Requirements**
- **Missing Requirements**
- **Requirements coverage**

# Example of ArgoSim Stimulus



- Conflicting Requirements
- Missing Requirements
- Requirements coverage



# Conflicting Requirements

The screenshot shows the Argosim interface with several tabs at the top: Start page, FML\_import, TestSuite, LLR\_v3, World, and LLR\_v2. The LLR\_v2 tab is active.

The main area displays two sets of requirements:

**Initially**

- If lightIntensity is less than or equal to 70[percent] then headLight shall be 'ON'

**afterwards**

- When lightIntensity is less than or equal to 70[percent], headLight shall be 'ON' (highlighted in red)

**REQ\_003.4**

**Initially**

- If lightIntensity is greater than 70[percent] then headLight shall be 'OFF'

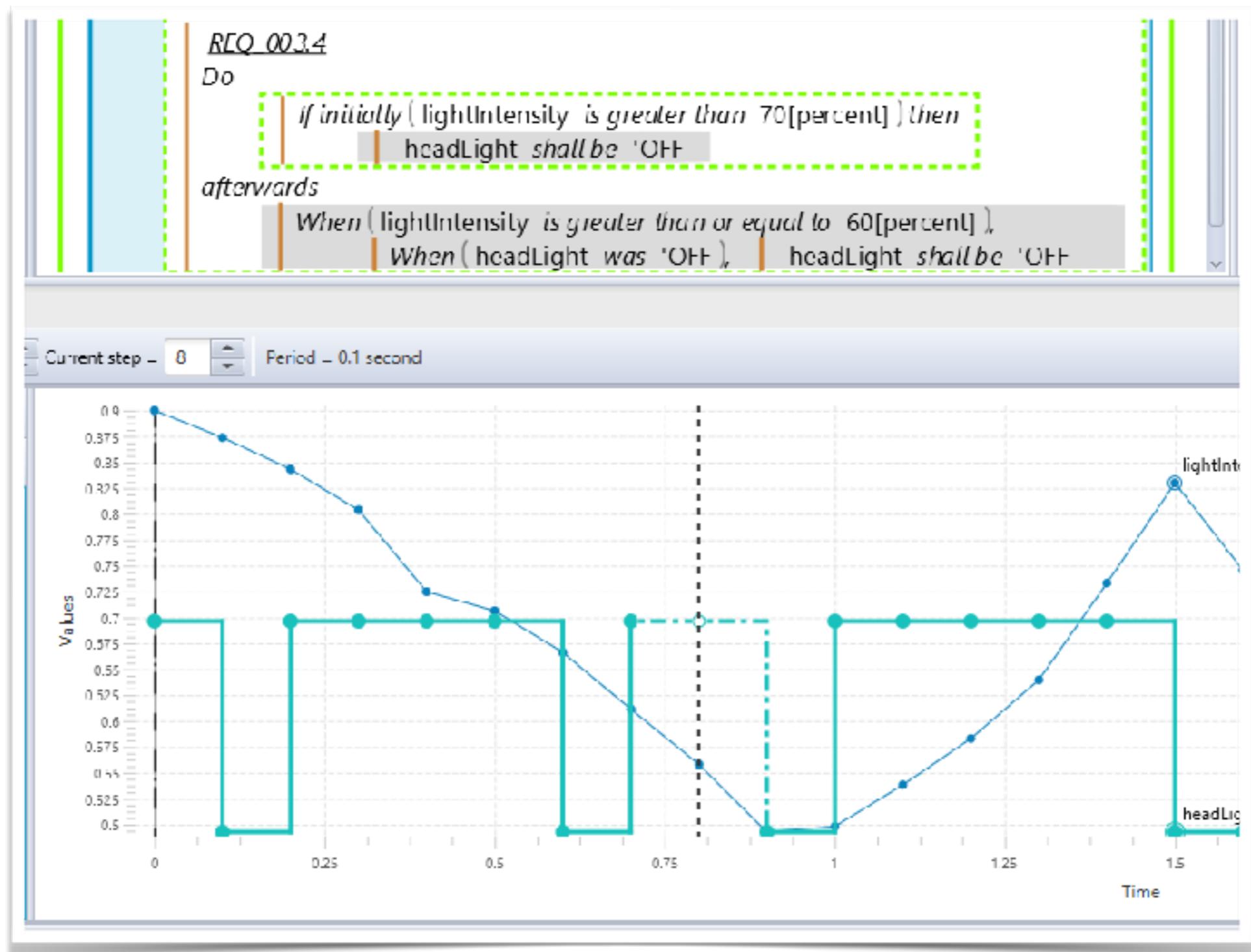
**afterwards**

- When lightIntensity is greater than or equal to 60[percent], headLight shall be 'OFF' (highlighted in red)

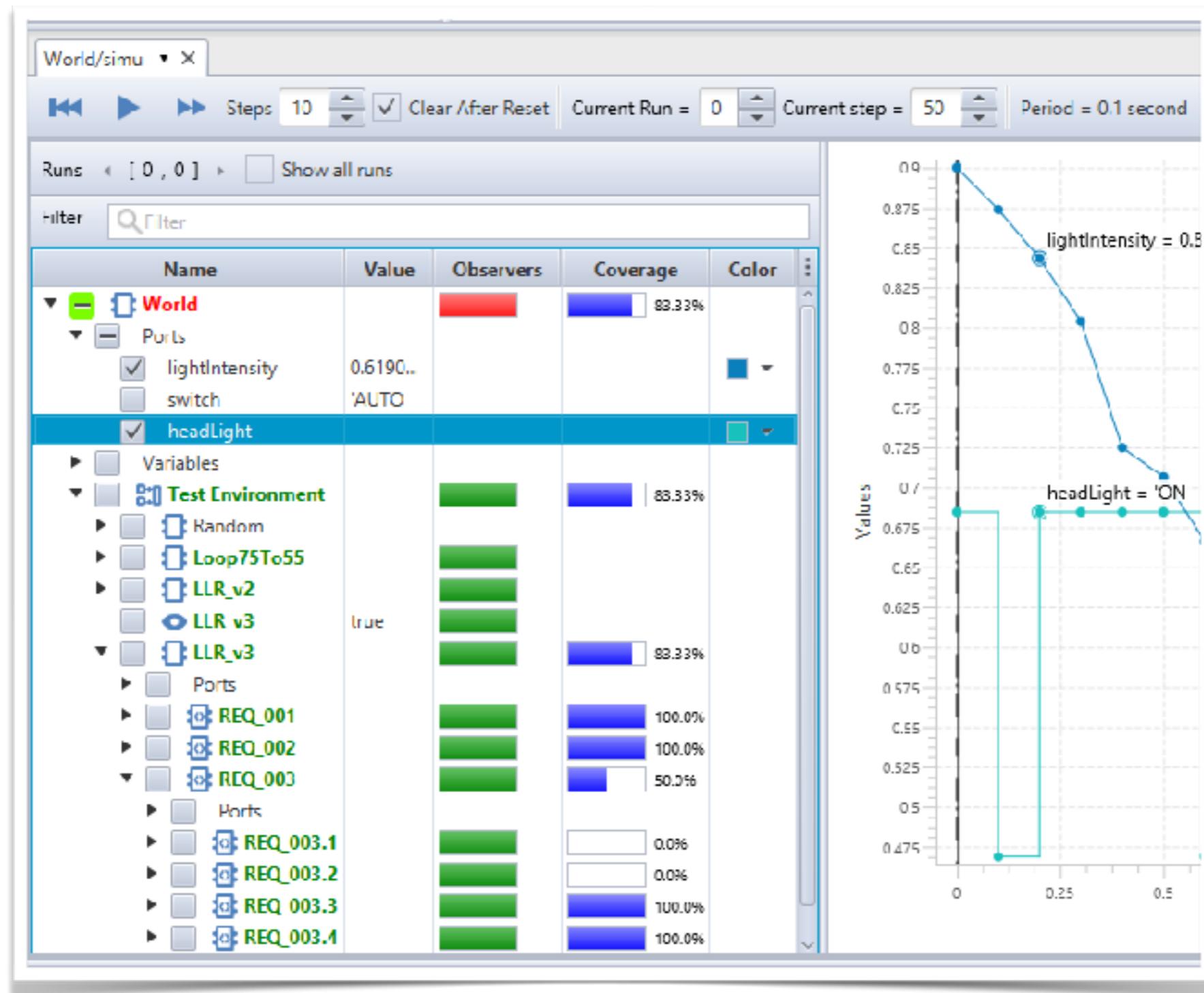
At the bottom, there is a toolbar with buttons for Run After Reset, Current Run (0), Current step (50), and Period (0.1 second). Below the toolbar is a table:

Error Level	Message	Run / Step
Error	Conflict in LLR_v2 inside LLR_v2 on variable headLight no values for variable headLight can satisfy the following constraints: headLight = 'ON' (1) from headLight shall be 'ON' headLight = 'OFF' (0) from headLight shall be 'OFF'	0 / 50

# Missing Requirements



# Requirements coverage

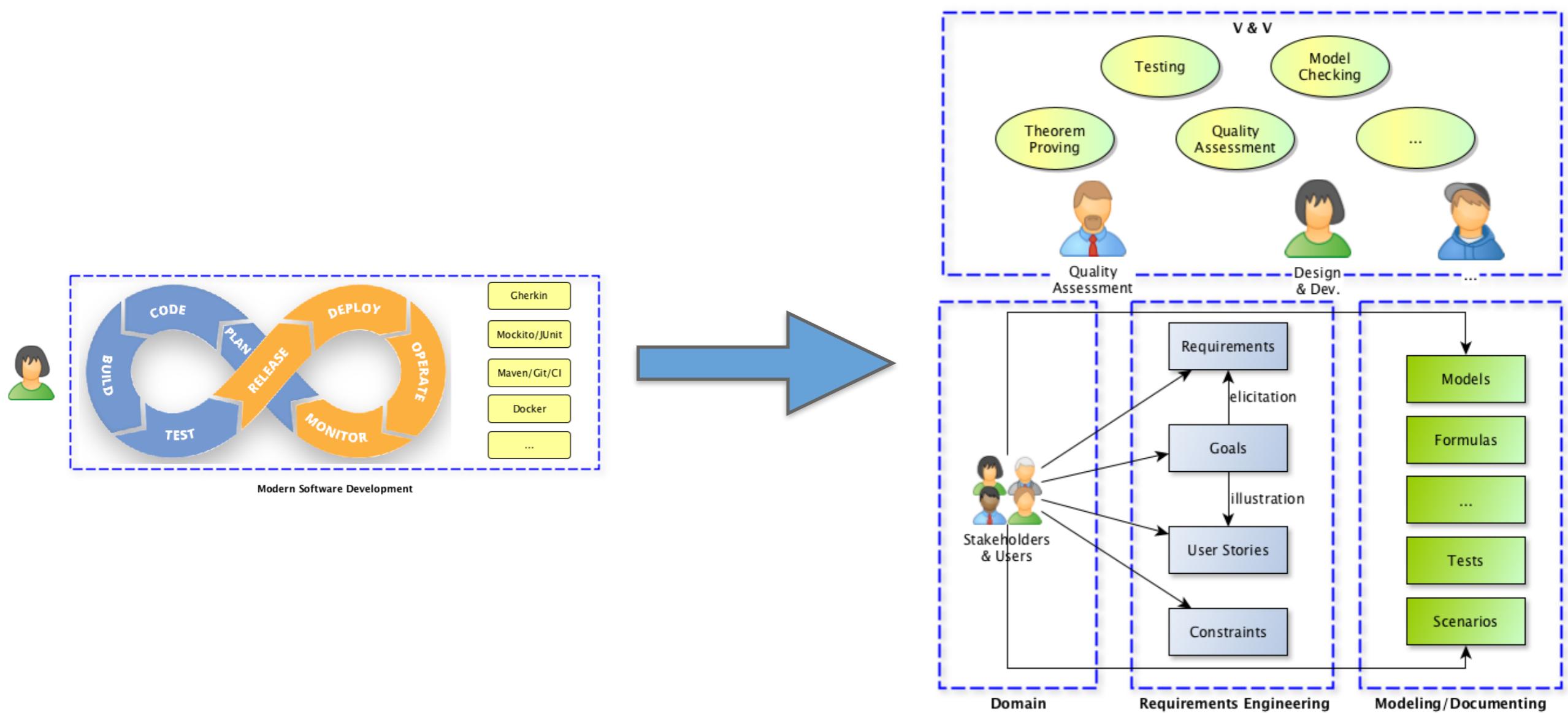


<https://www.argosim.com/home/product-overview/>

OK

what do we do  
now?

# Apply good practice to RE



**Let's do it!**

# Gherkin

**Feature:** Logout from application

**Scenario:**

Given I am logged in

When I click "log out" button

Then I am informed about successful logout

And I am redirected to login page

unambiguous  
explicit

???

behavior only

# Gherkin

```
Feature: Logout from application
Scenario:
  Given I am logged in
  When I click "log out" button
  Then I am informed about successful logout
  And I am redirected to login page
```

To create a new requirements description, we need to define the **Feature** which gives us the name of a new **Functionality**. Then, we go ahead with writing the **Scenario**.

# Gherkin

```
Feature: Logout from application
Scenario:
  Given I am logged in
  When I click "log out" button
  Then I am informed about successful logout
  And I am redirected to login page
```

To create a new requirements description we need to define the **Feature** with the name of a new **Feature**. In this case, we go ahead with **Logout from application**.

**Requirement Engineering** is a discipline that deals with the process of eliciting, specifying, and refining requirements for software systems. It involves understanding the needs of stakeholders and translating them into a formal language that can be used for development and testing.

Human  
resistance

# When people **don't** see the value of CI

<http://itsadeliverything.com/five-things-to-do-when-people-dont-to-see-the-value-of-automation#more-5966>

- Get an technical Ally
- Build monitor
- Pair programming
- Training
- Change the team



# Notes en vrac

# System

---

« an open set of complementary,  
**interacting parts**, with properties,  
capabilities and behaviours **emerging**,  
both from the parts and from their  
interactions, to synthesize a unified  
whole »

Derek Hitchins

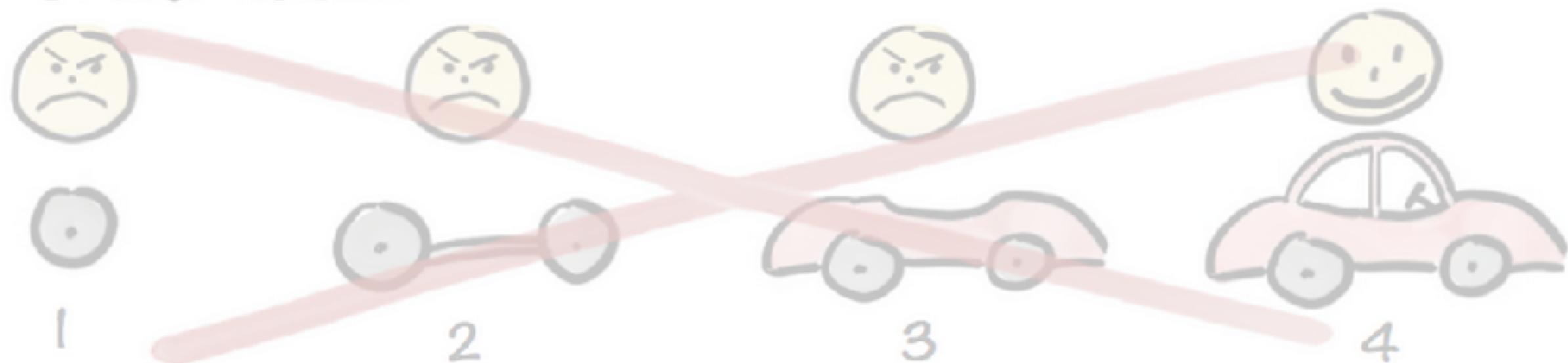
# Open System

---

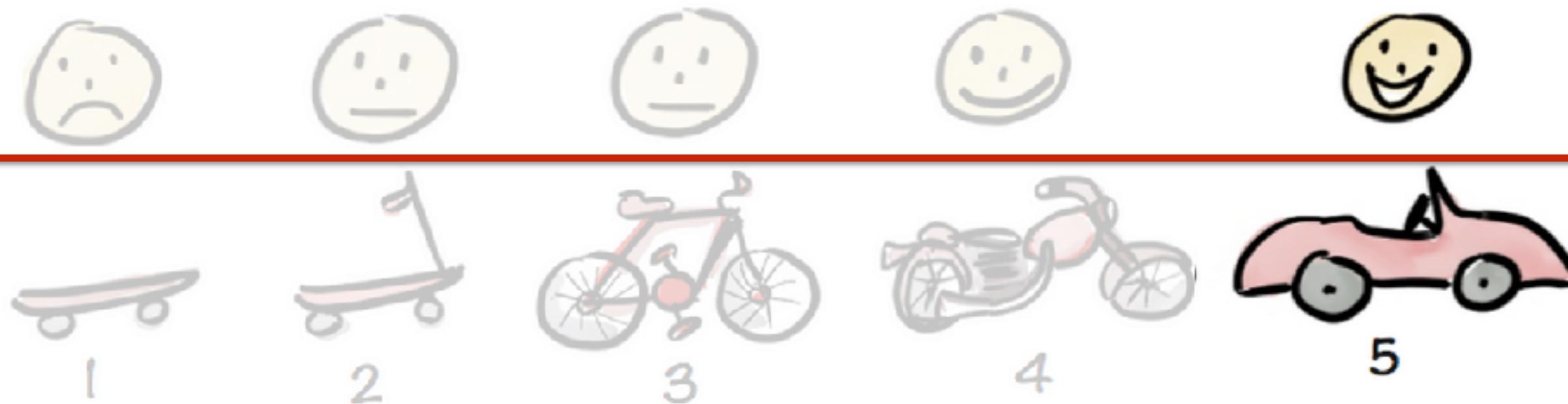
**Context and environment** should be described too

# Model Validation

Not like this....



Like this!



by Henrik Kniberg

Not like this



1



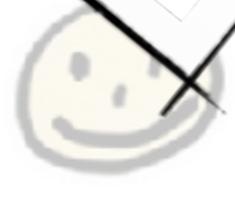
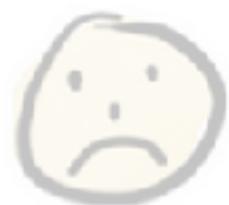
Required  
Models Validation

Validation



4

Like this!



1



2



3



4



5

# Model Quality Criteria

---

[Friedenthal 2016]

- Is the model's purpose well defined?
- Is the model's scope sufficient?
- Is the model complete?
- Is the model well-formed?
- Is the model consistent?
- Is the model understandable?
- Are modeling conventions documented and used consistently?
- Is the model self-documenting?
- Does the model accurately reflect the domain of interest?
- Does the model integrate with other models?

# Model-Based METRICS

---

[Friedenthal 2016]

- What is the quality of the design?
- What is the progress of the design and development effort?
- What is the estimated effort to complete design & development?

# Kata

# Poker Kata SE approach

---

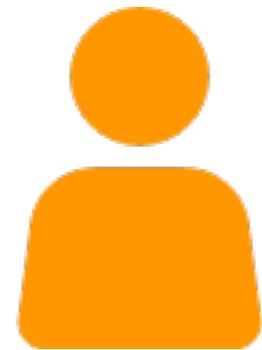
« Make a program that can create, evaluate,  
and compare 5-card poker hands »

# Stakeholders identification

---



User



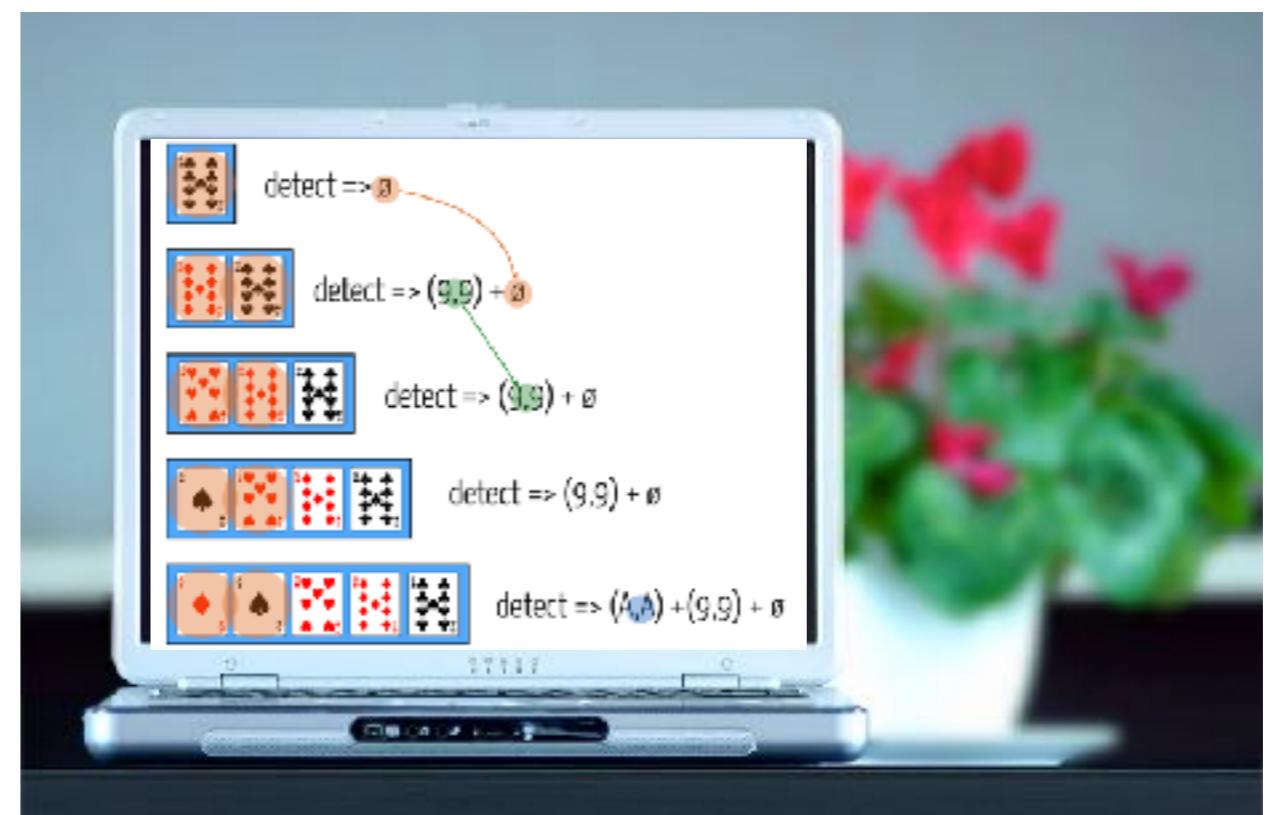
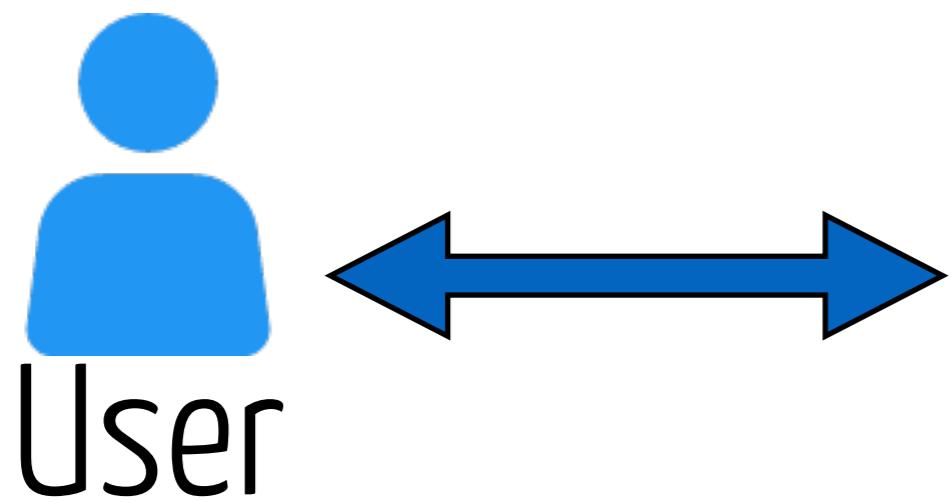
Poker Rules



Client

# System environment

---



# Requirements Elicitation

---

- Use Scenarios to elicit Requirements

# Requirements Analysis & Negotiation

---

- Needs => Requirements
- Prioritize Requirements
- Classify Requirements
- Rules => Invariants (boolean properties that should always be true)
- Assess Requirements Risks

# Requirements list

---

Reqs Priority / Stakeholders	Client 	Poker Rules 	User 
Must	Comparing Poker Hands	See Rules*	Comparing Poker Hands
Should	Create hands		Provide stats
Could	Provide stats		Select card games (style,...)
Will	Have an API for game plugin		Have the menu in French

# Requirements Risk Assessment

---

Reqs Priority / Stakeholders	Client 	Poker Rules 	User 
Must	Comparing Poker Hands	See Rules*	Comparing Poker Hands
Should	Create hands		Provide stats
Could	Provide stats		Select card games (style,...)
Will	Have an API for game plugin		Have the menu in French

# Functional decomposition

---

- The program should **manipulate** Poker Hands (PH)
  - It should be able to **Read** PH: **F<sub>1</sub>**
  - from keyboard
  - from various file format (JSON, XML,...)
- It should be able to **validate** a PH **F<sub>2</sub>**
- It should **score** a PH **F<sub>3</sub>**
  - It should **detect combinations** **F<sub>4</sub>**
- At any time only valid PH should be considered

# Requirements Validation

---

- Propose Requirements Test Cases

# Requirements Documentation

---

- Automate documentation generation

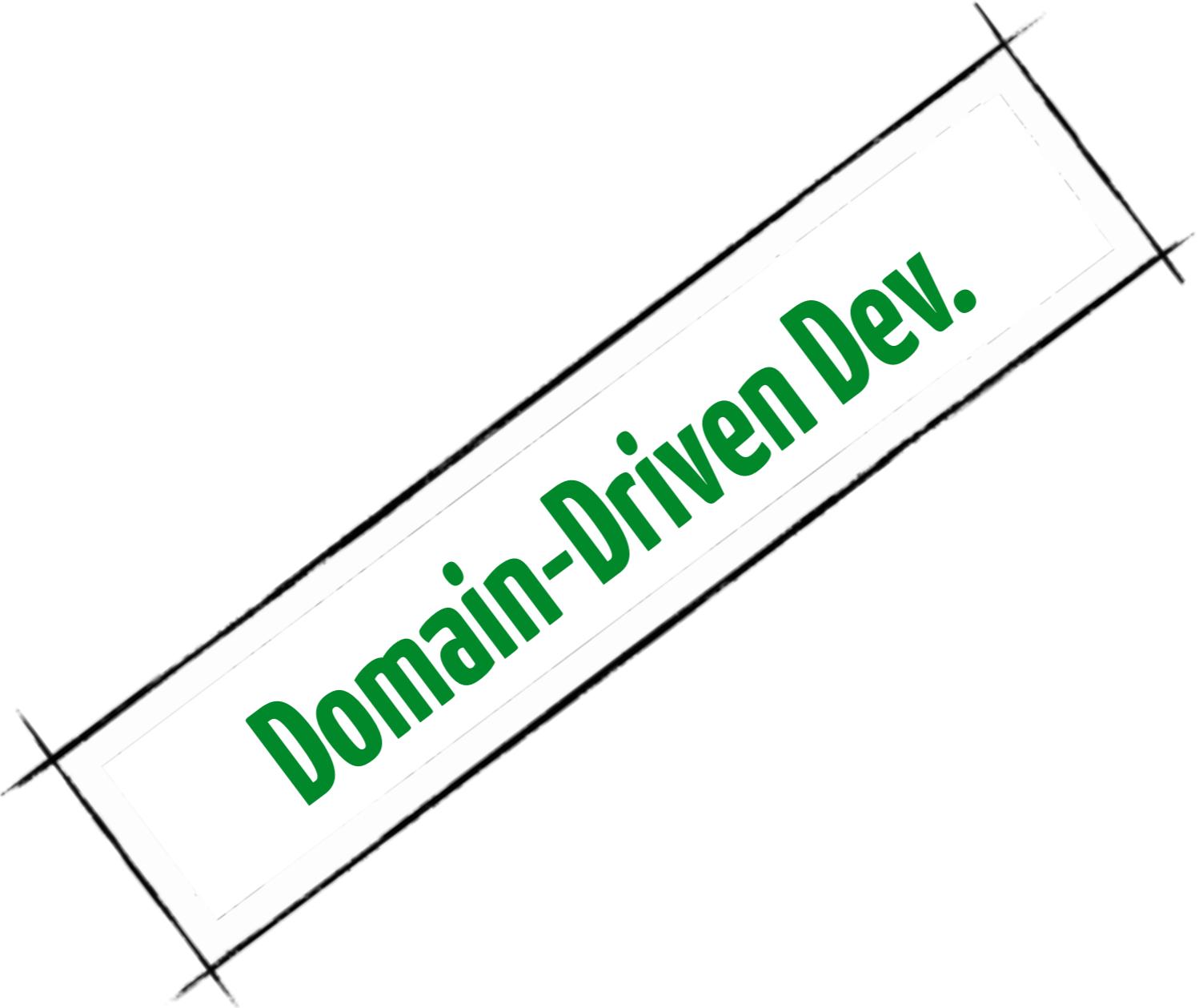
# Requirements Management

---

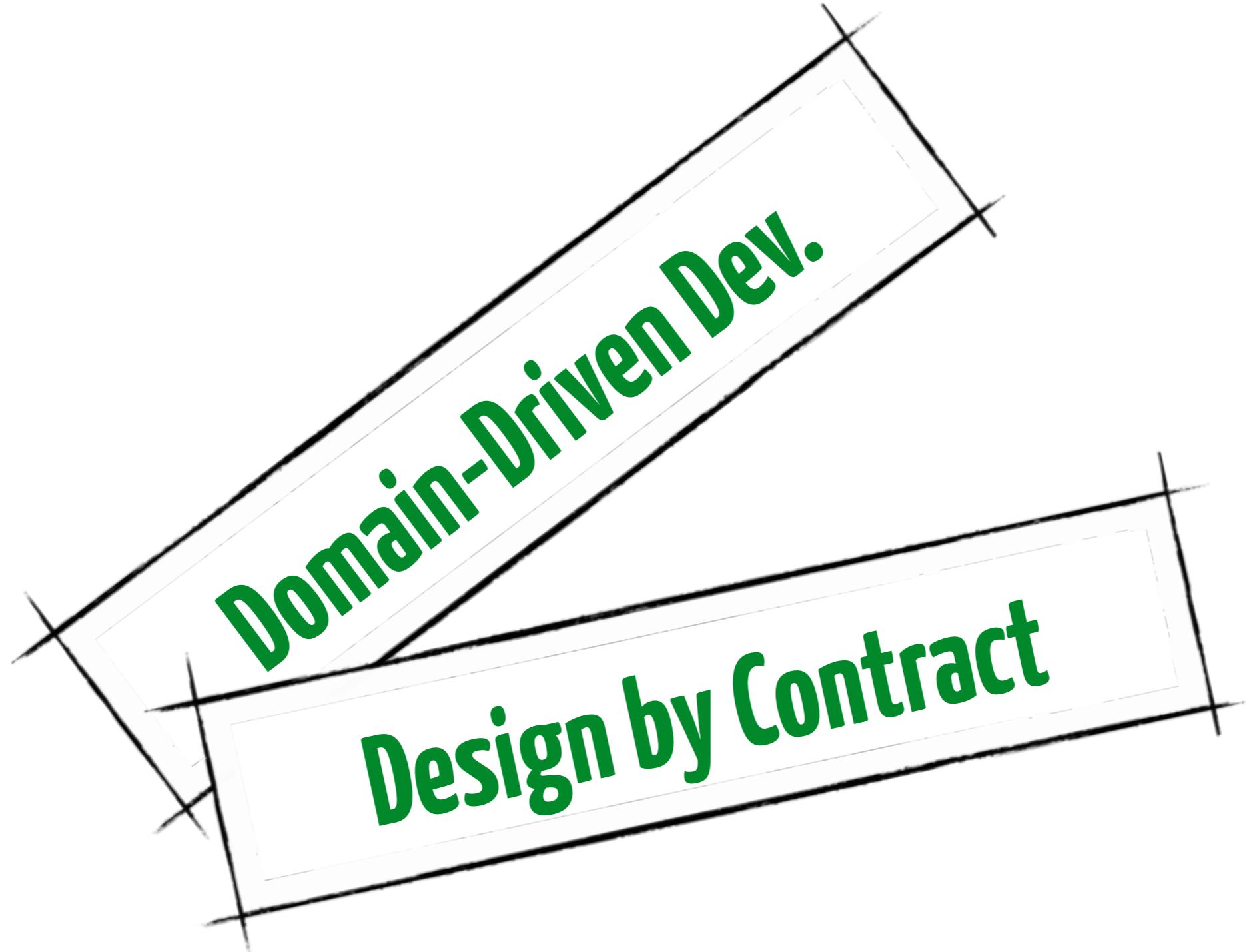
- Use requirements as tickets/features in a Project Manager

We haven't talk  
about...





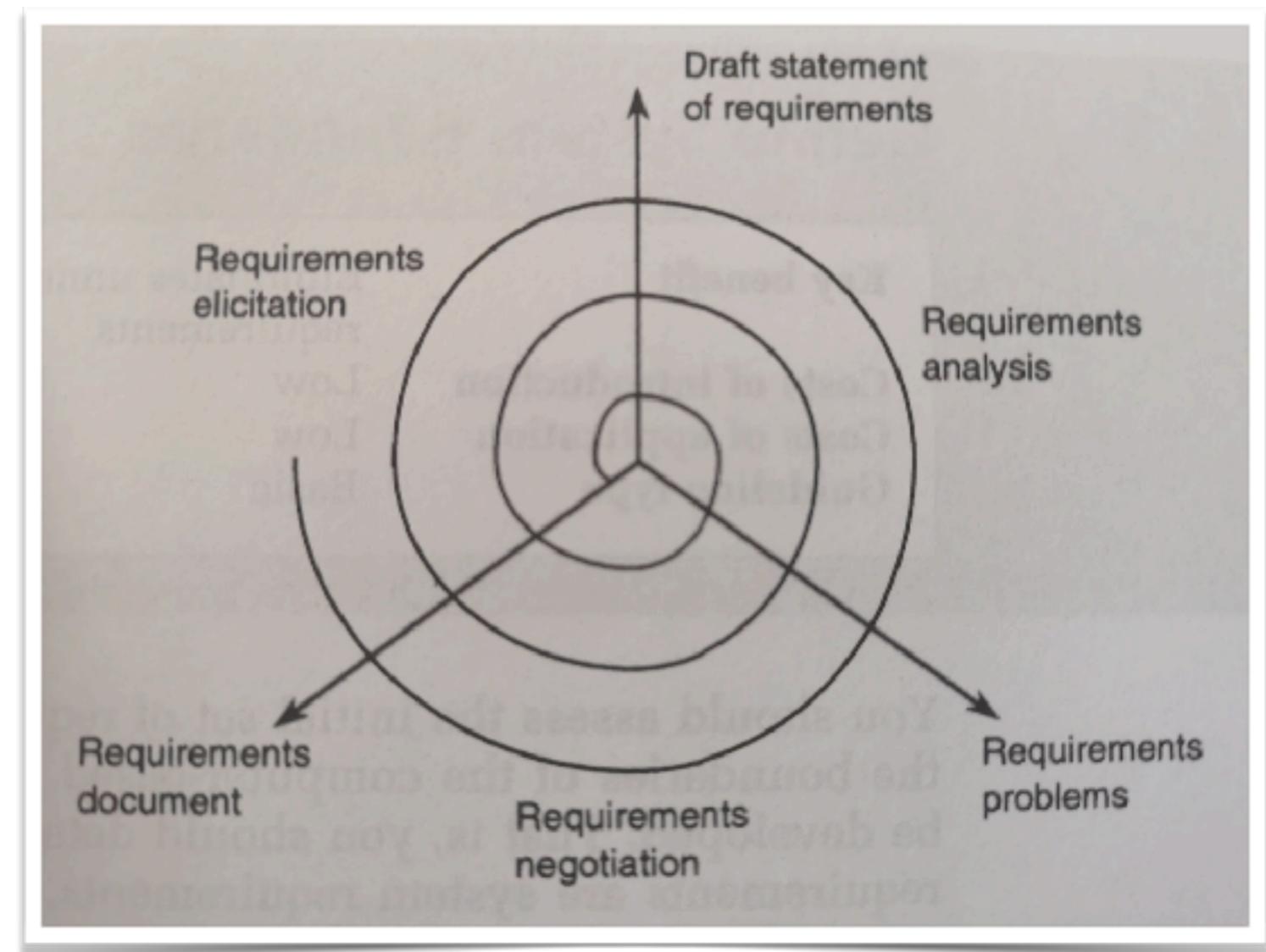
Domain-Driven Dev.



# RE Process

# Requirements Engineering Process

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



# Requirements Elicitation

---

- Use Scenarios to elicit Requirements

# Requirements Analysis & Negotiation

---

- Prioritize Requirements
- Classify Requirements
- Assess Requirements Risks

# Requirements Validation

---

- Propose Requirements Test Cases

# Requirements Documentation

---

- Automate documentation generation

# Requirements Management

---

- Use requirements as tickets/features in a Project Manager

# BTW...

