

# Agile Projects Reality Check

TU Darmstadt,  
26.11.2015,  
Stefan Merten

People matter, results count.

# Agenda

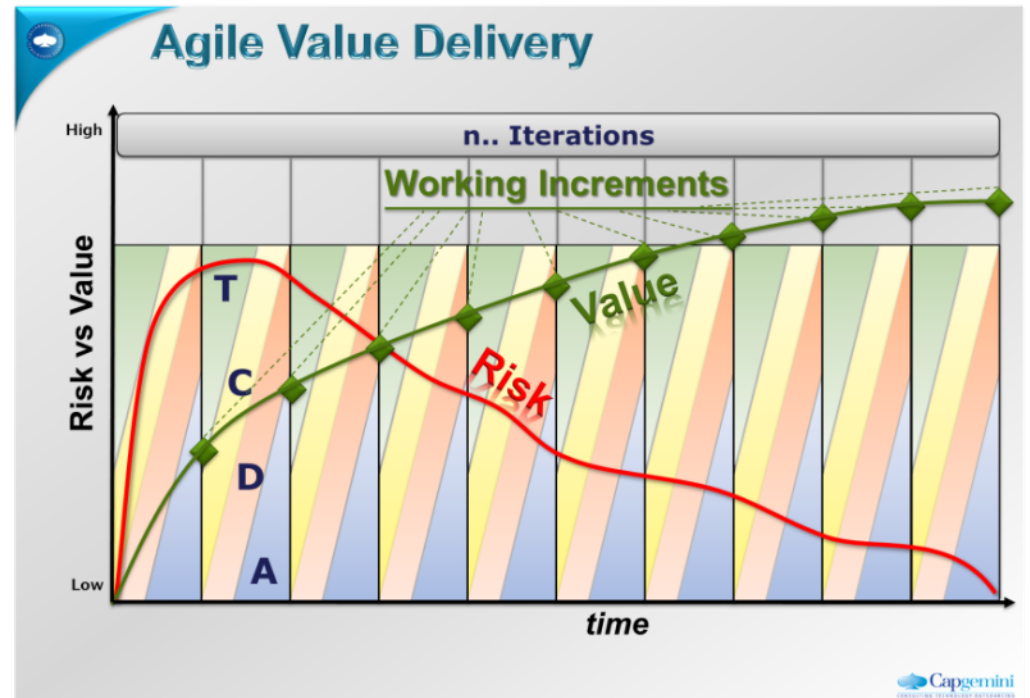
- Proposition
- Contracting
- How to get things DONE?
- Development AND Operations



# That's what we mean by Agile

The agile process is designed for flexibility

- People-centric / communicative
- Iterative
- Incremental
- Evolutionary



# Agenda

## ■ Proposition

- Contracting
- How to get things DONE?
- Development AND Operations



# Agile is never like Agile

Customer/Client



I want to sell my stuff in an online-shop. As I don't know all the requirements yet, I want to do an agile project for that.



Here is my initial (very rough) Product Backlog. I did an initial estimation on complexity. How much will that cost.



Why shall I go with you, guys?

Don't

Yihaa. That's what we want, too. Let's go.



Let's analyze one requirement and then upscale the efforts.



We are all certified scrum masters.



Do

Have you done agile projects before? What do YOU mean by agile?



As a first agile step, let's do the initial estimation together. However, it is not possible to give an exact price on an unknown product.\*)



Agility means, adapting to clients/projects needs. But we have a ready-to-go approach to start with:...





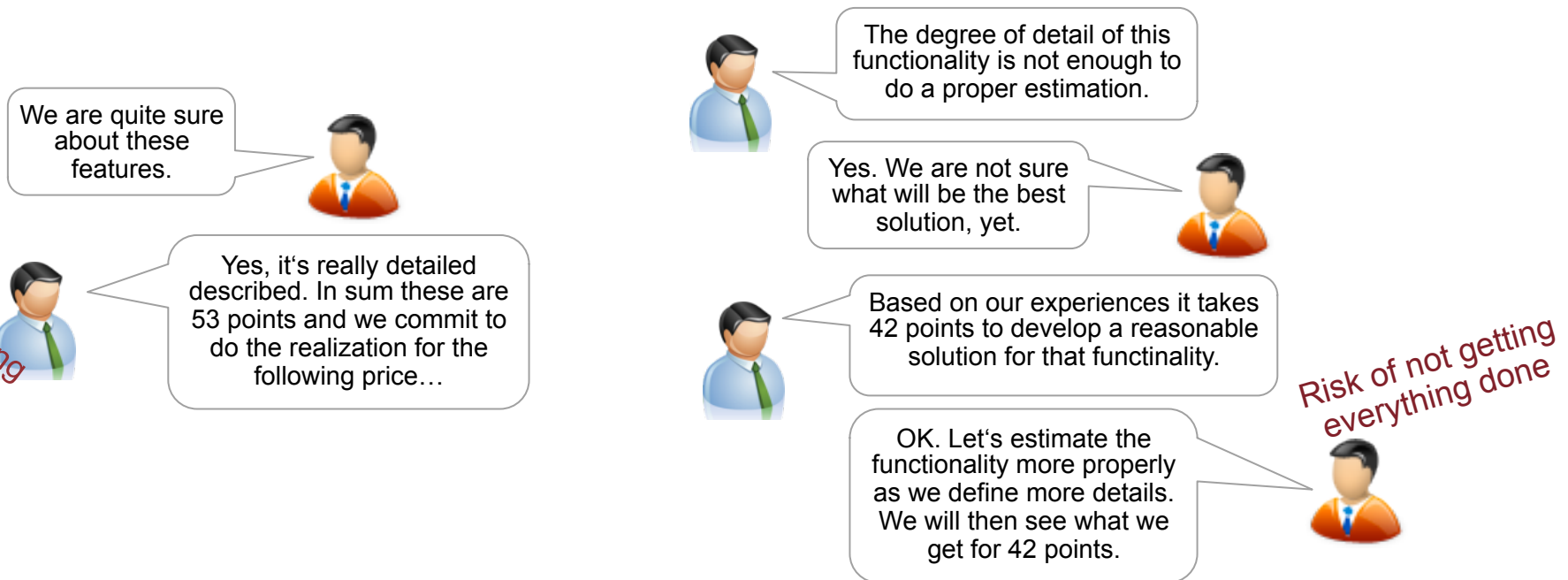
# Agenda

- Proposition
- **Contracting**
- How to get things DONE?
- Development AND Operations



# Find the initial scope (product backlog)

- The effort-estimation in agile contracts is based on abstract functional points, expressing the functional complexity. **The customer and the service provider agree on a common estimation basis and method.** They empower and trust the people they choose to do the estimation.
- Depending on the degree of detail of the provided functional specification Capgemini commits to a **piece of work** or to a **number of story points** as a scope definition. These two approaches may be combined.

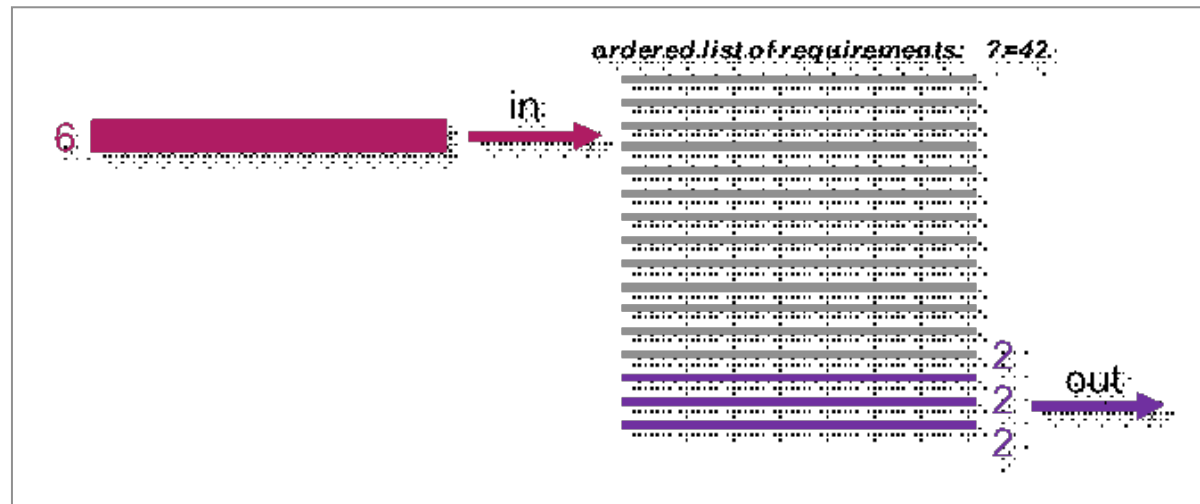


At least one part of the specification must be detailed enough, to get a basis for further estimations.

# Welcome change! Change-For-Free-Clause.

With a contract-of-work both parties agree on a fixed initial scope. To handle functional exchanges an agile contract contains a change-for-free clause:

- At the end of each sprint the customer may exchange the defined but not yet implemented scope by replacing currently in-scope items with a new higher prioritised item or by breaking down an existing item.
- The exchange does not influence the total amount of scope and velocity.



It should be transparent to all parties, changing or removing an already implemented feature is more expensive than developing it right upfront. BUT inspecting and adapting a feature step by step is sometimes the only way to determine the targeted functionality.



# I don't need more. Let's stop here!

*With the agile approach Capgemini continuously delivers business value and the customer avoids the realization of unnecessary functionality by making use of a “stop-at-business-value”-clause.*

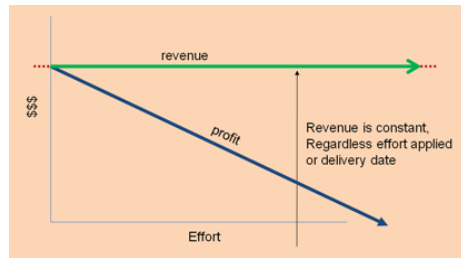
- The customer may decide that the realized functionality (=accomplished business value) is high enough. To that end the customer is allowed to quit the contract by paying the efforts occurred until the cancellation plus an previously negotiated rate of the not used rest-effort to Capgemini respecting bonus-penalty calculations.
- The decision to stop the development must be communicated to Capgemini's project manager with an adequate notice period for ramping down the project staff.



# The three (main) contract models

## Valuable software

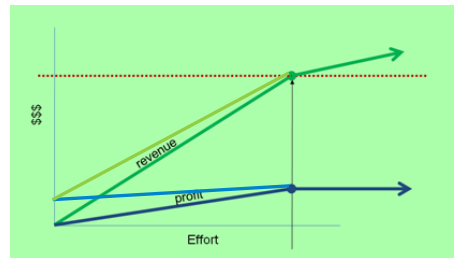
### Fixed-price-contract for work



- (Fixed) initial scope
- Fixed price, fixed time
- Stop-at-business-value
- Change-for-free
- Responsibility for the resulting product

### Fixed price

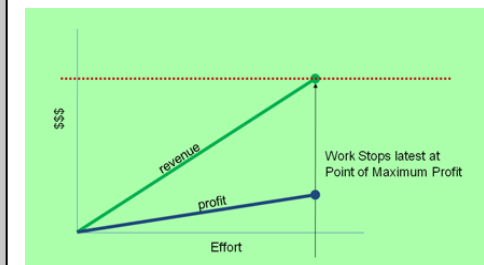
### t&m-contract for work



- (Fixed) initial scope
- Estimated price and time
- Bonus-penalty-option if estimate is not met
- Change-for-free
- Responsibility for the resulting product

### Estimated price

### t&m-contract for services



- Fixed budget or time
- Just price per effort
- Variable scope
- No responsibility for the resulting product

### Normal t&m

## Agile approach

# It's easier claiming to be agile, than being agile

## Customer obligations

- The customer needs to know what agile means and has to follow the **agile values**.
- The **Product Owner (PO) must be accepted and empowered** by all stakeholders to make project relevant decisions. On the other side the PO himself must collaborate closely with the stakeholders and the development team.
- Agile offers a high degree of transparency. The customer's duty is to **adequately use the offered information**. Project relevant information can be "pulled" at any time.
- Agile processes are built on feedback loops which means there are regular meetings between the customer and Capgemini to provide feedback on artefacts. It is recommended to **give feedback as early as possible**.
- An agile development approach has a really tight schedule. To keep the sustainable pace it is required, that **requirements are prepared in-time** according to a "definition of ready" and decisions are made within a **predefined response-time** by the customer.
- Any known **technical dependencies or restrictions** to the project must be immediately communicated at the beginning of the project or whenever they arise.
- **Changes to key persons** (especially the PO) must be avoided. At any case there must be an empowered substitute in case of illness or holidays.

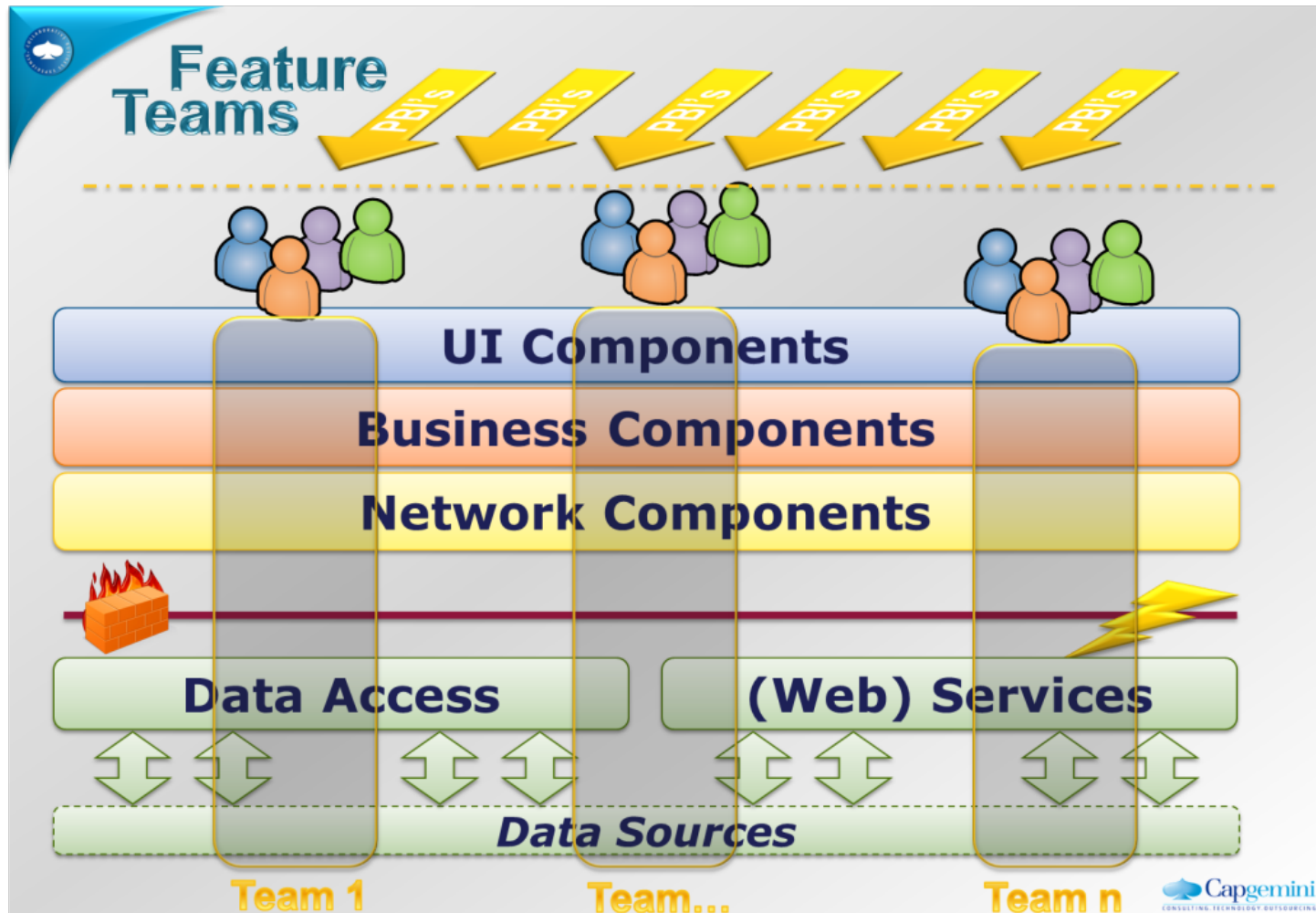


# Agenda

- Proposition
- Contracting
- **How to get things DONE?**
- Development AND Operations

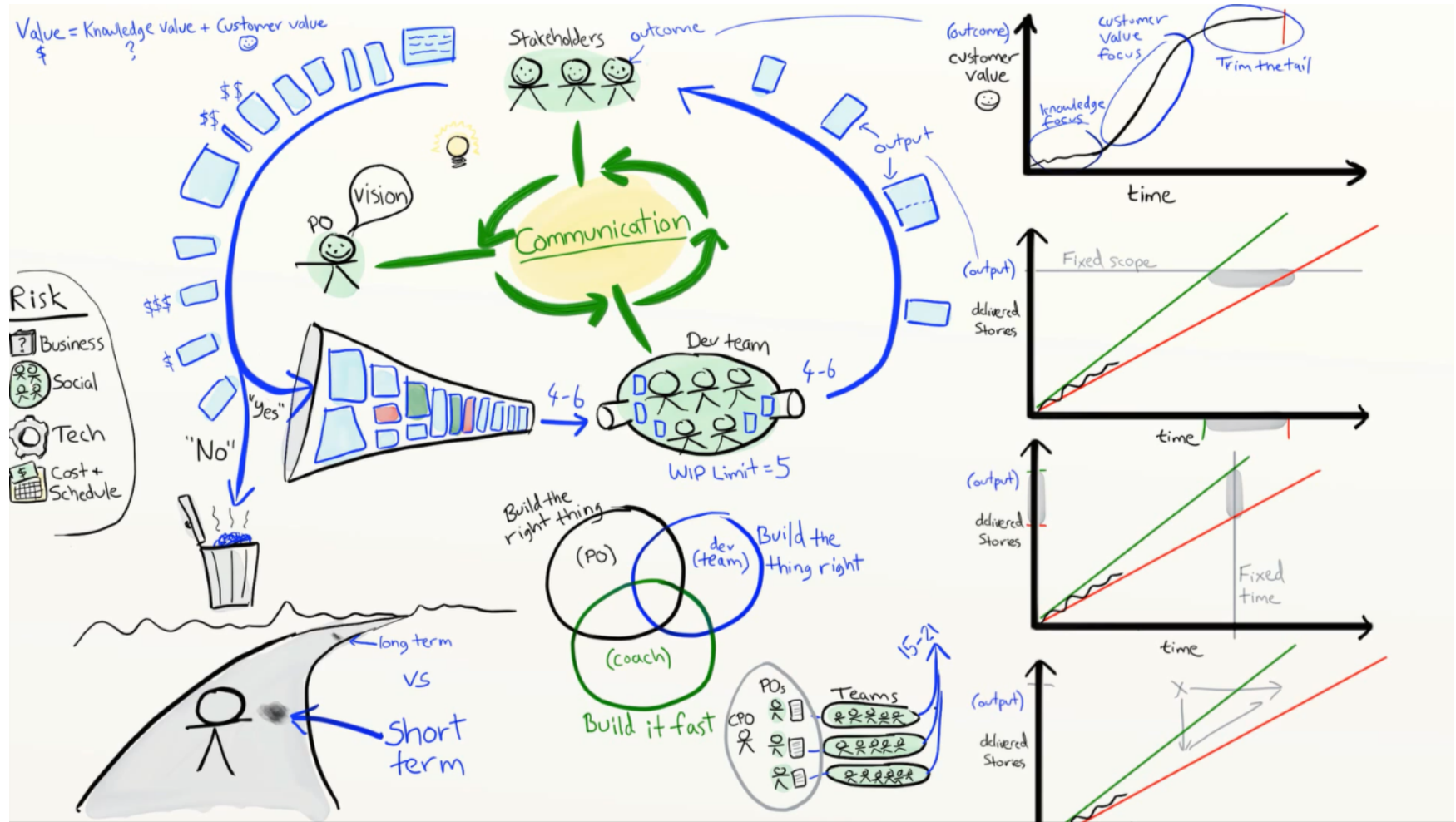


# Create cross-functional (feature-)teams





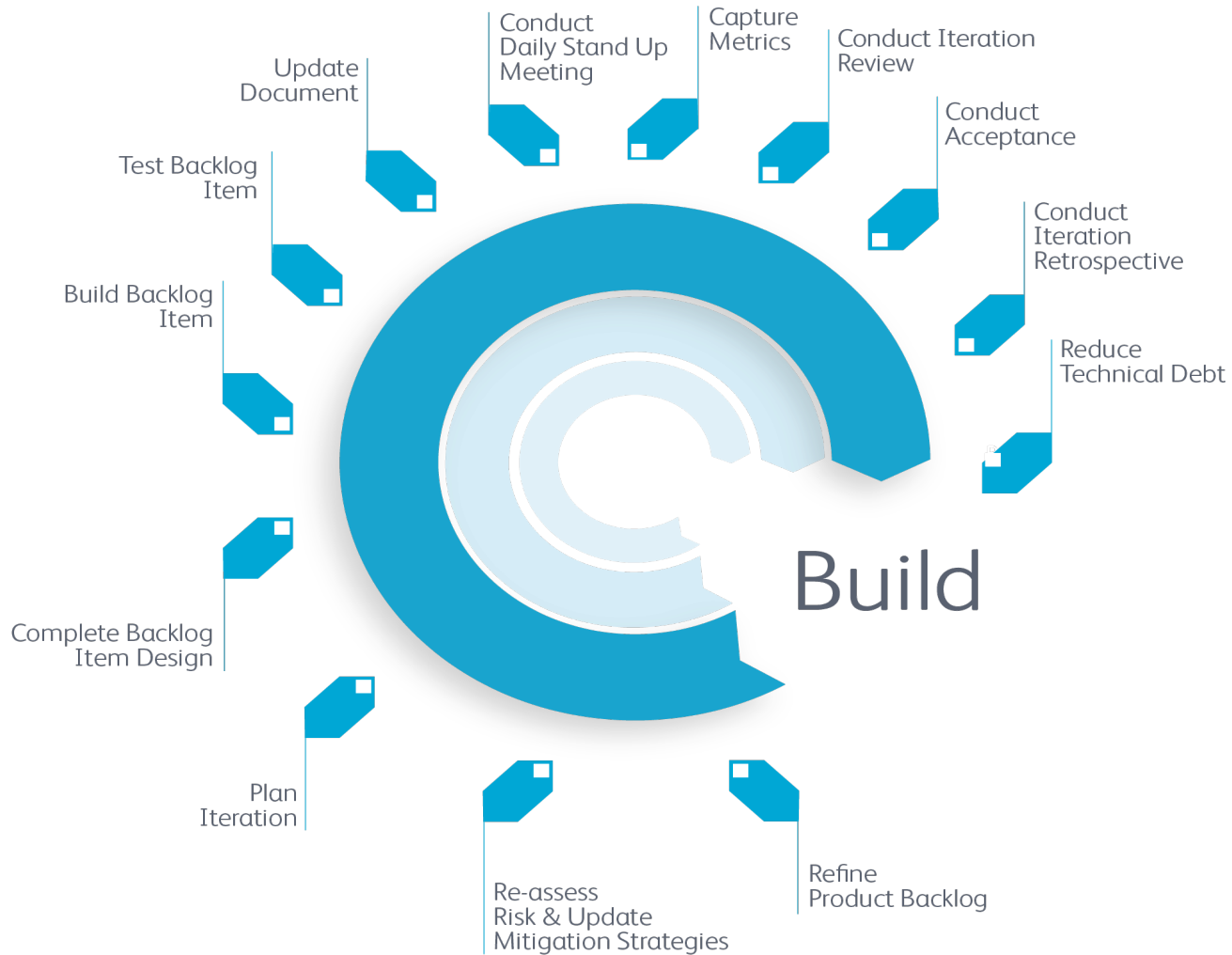
# Find a good Product Owner (team)



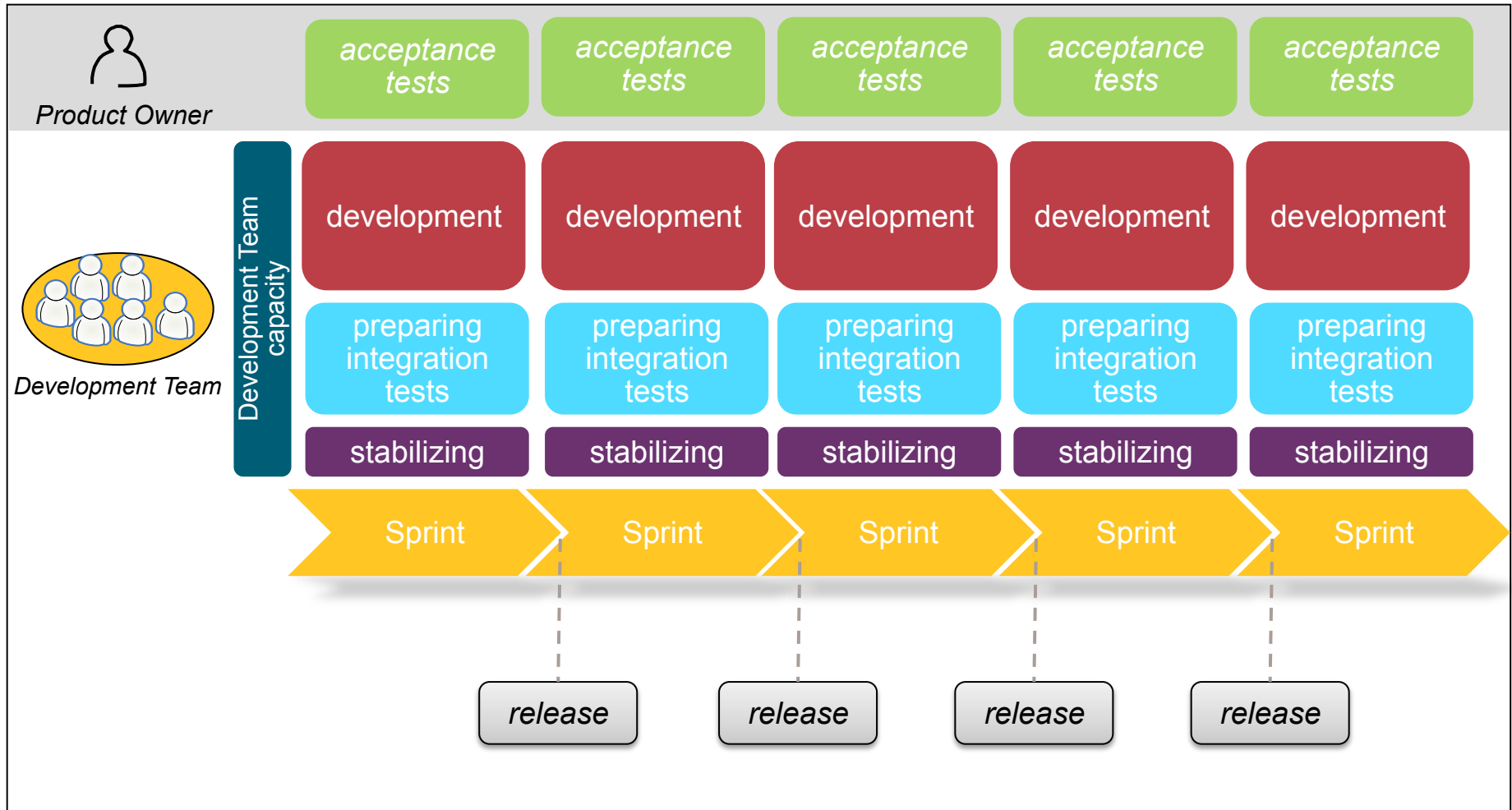
<https://www.youtube.com/watch?v=502ILHjX9EE>



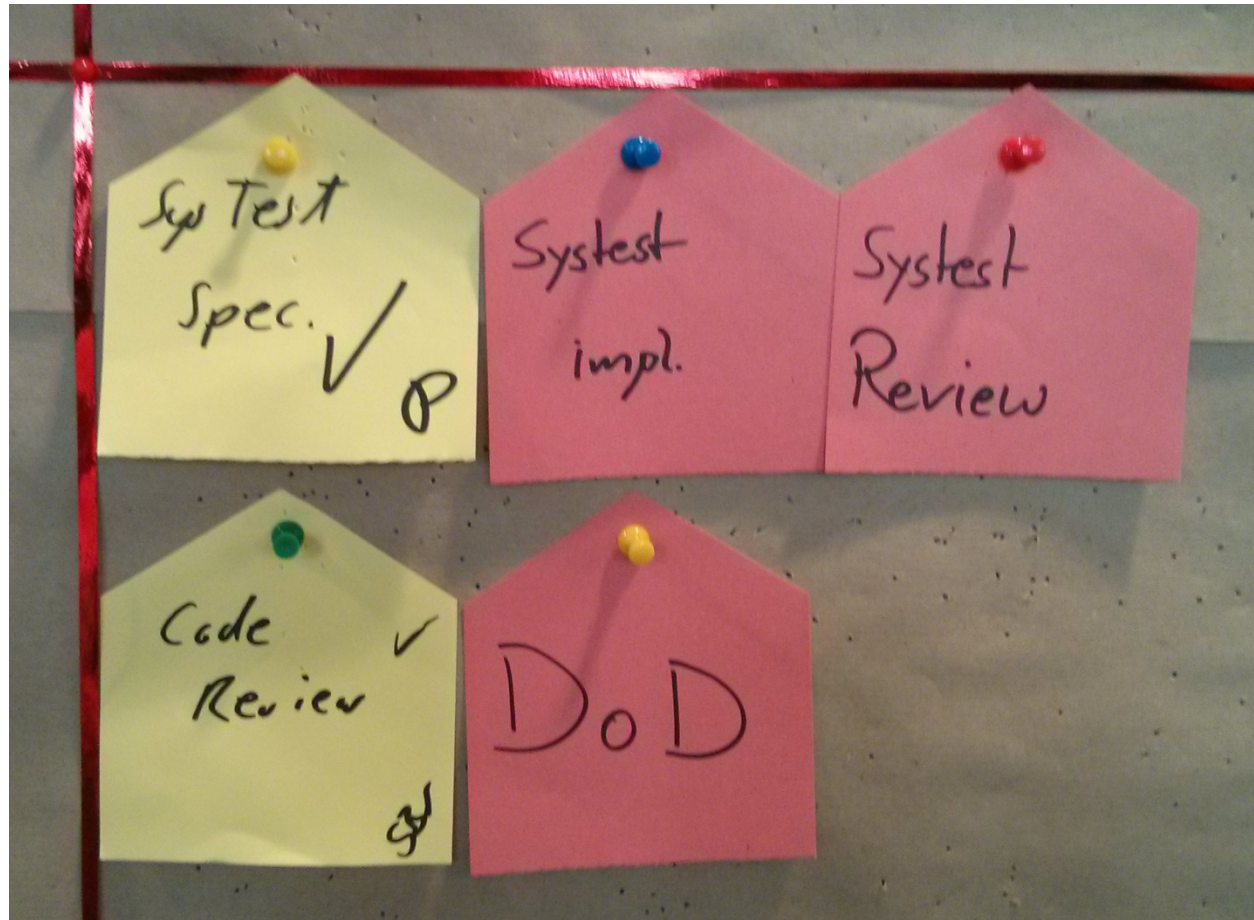
# There is more than just coding



# There are only sprints. Release with every sprint.

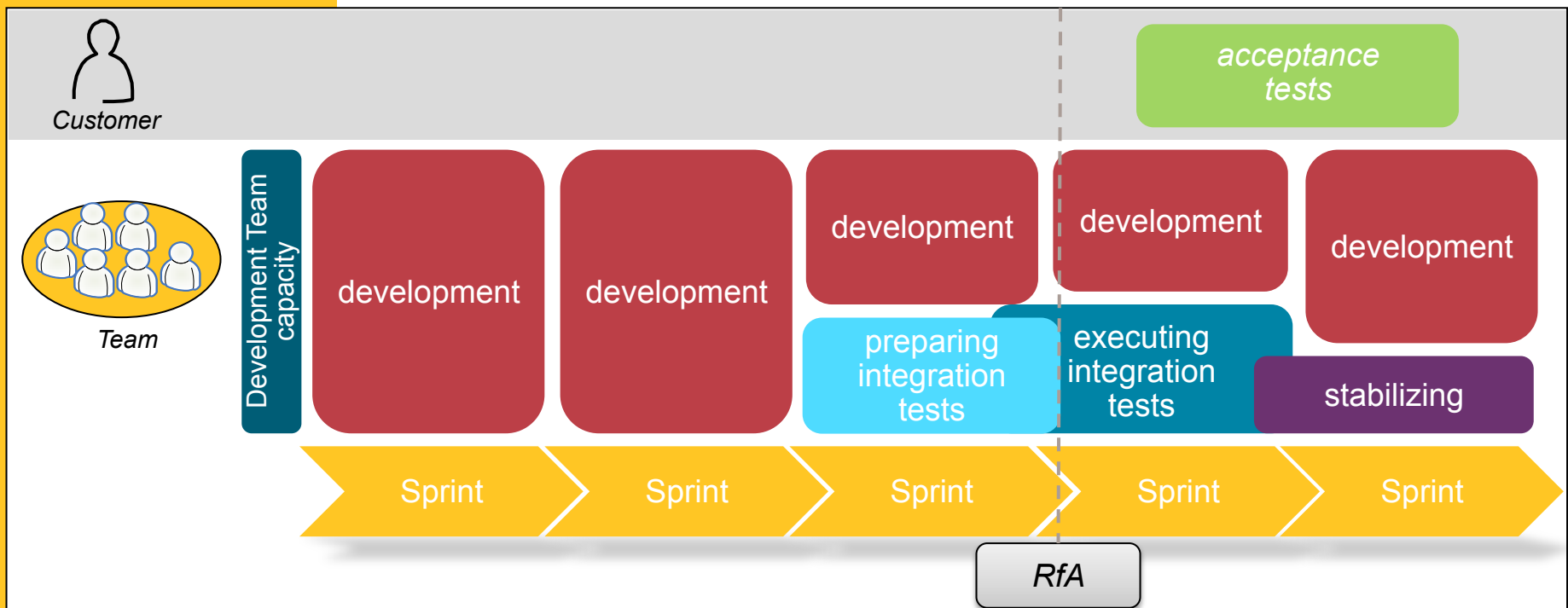


# Make QA visible in the Sprint Backlog

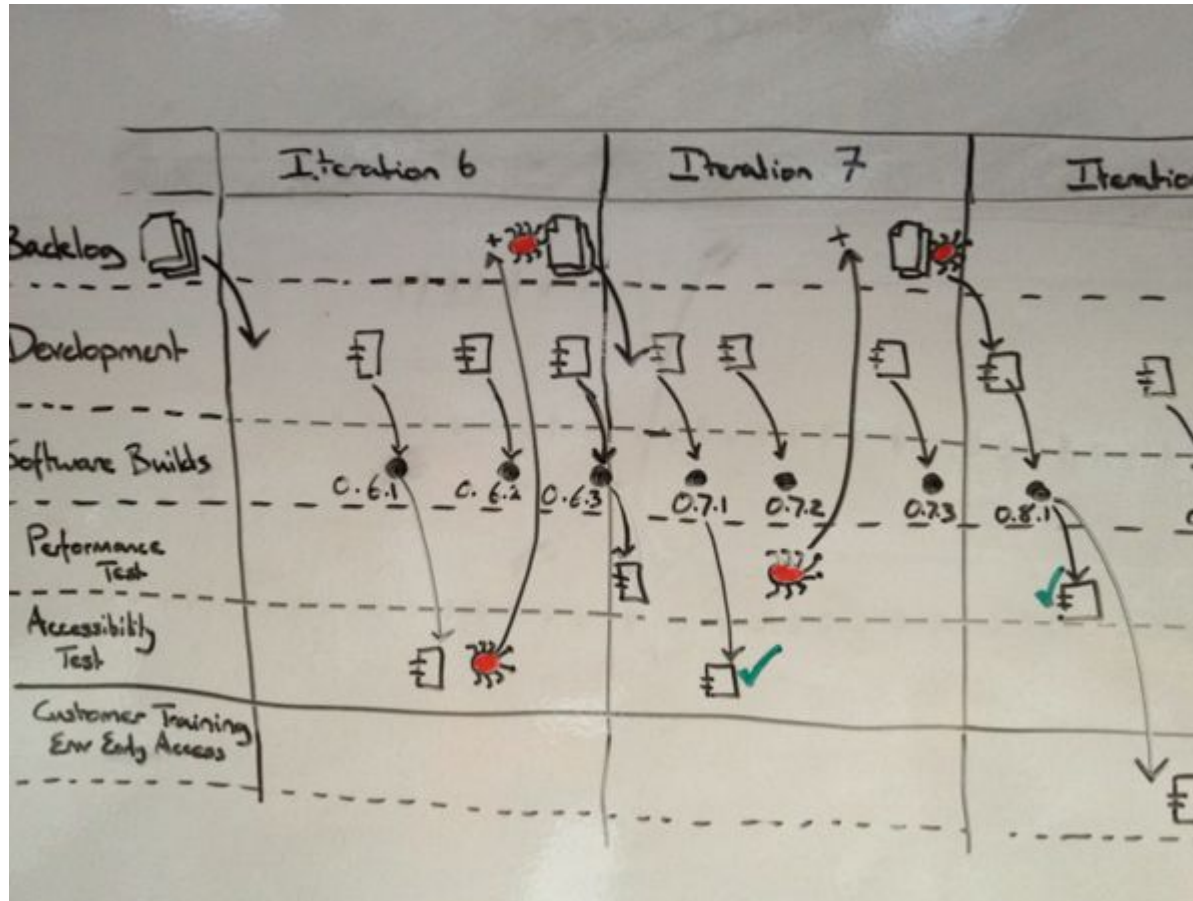


# Big software needs big testing

- Team capacity reserved for development is reduced when preparing and executing integration tests and for stabilizing.
- Parallel to the Team, Customer can execute acceptance tests and report bugs that are planned as Sprint Backlog artifacts and fixed in time planned for stabilization.



# Not all the testing could be done within the team



# Agenda

- Proposition
- Contracting
- How to get things DONE?
- **Development AND Operations**



# Demand, deliver, operate

Customer/Client Business



I want to sell my stuff in an online-shop. As I don't know all the requirements yet, I want to do an agile project for that.

Customer/Client Operations

We deliver increments every three weeks to get fast feedback on the results

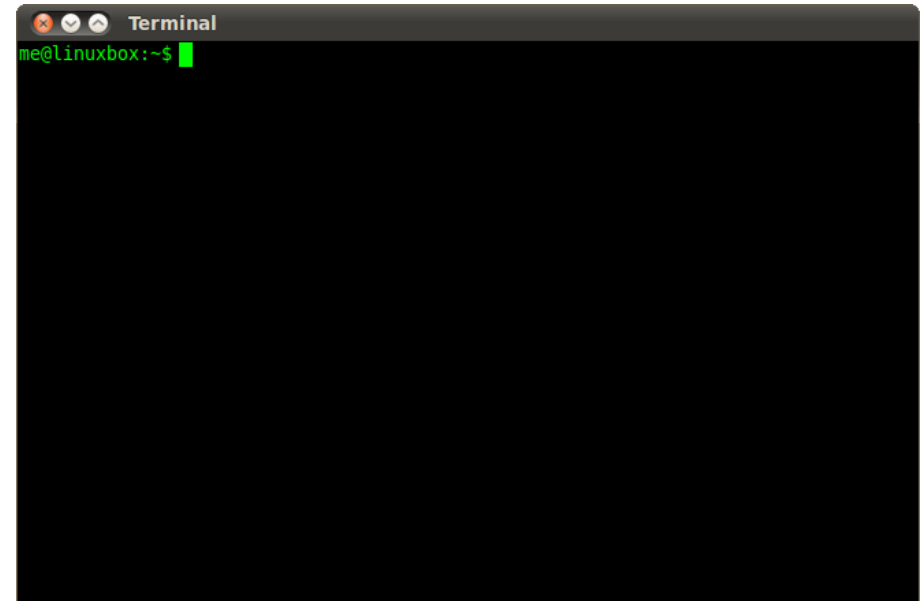
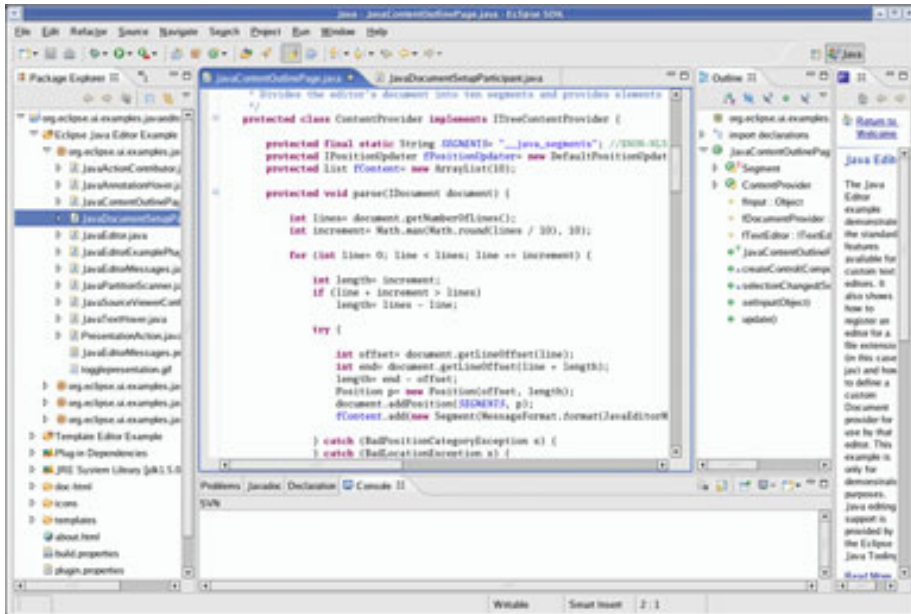


You



Stop! Release cycle time is six months.

# Problems in communication everywhere



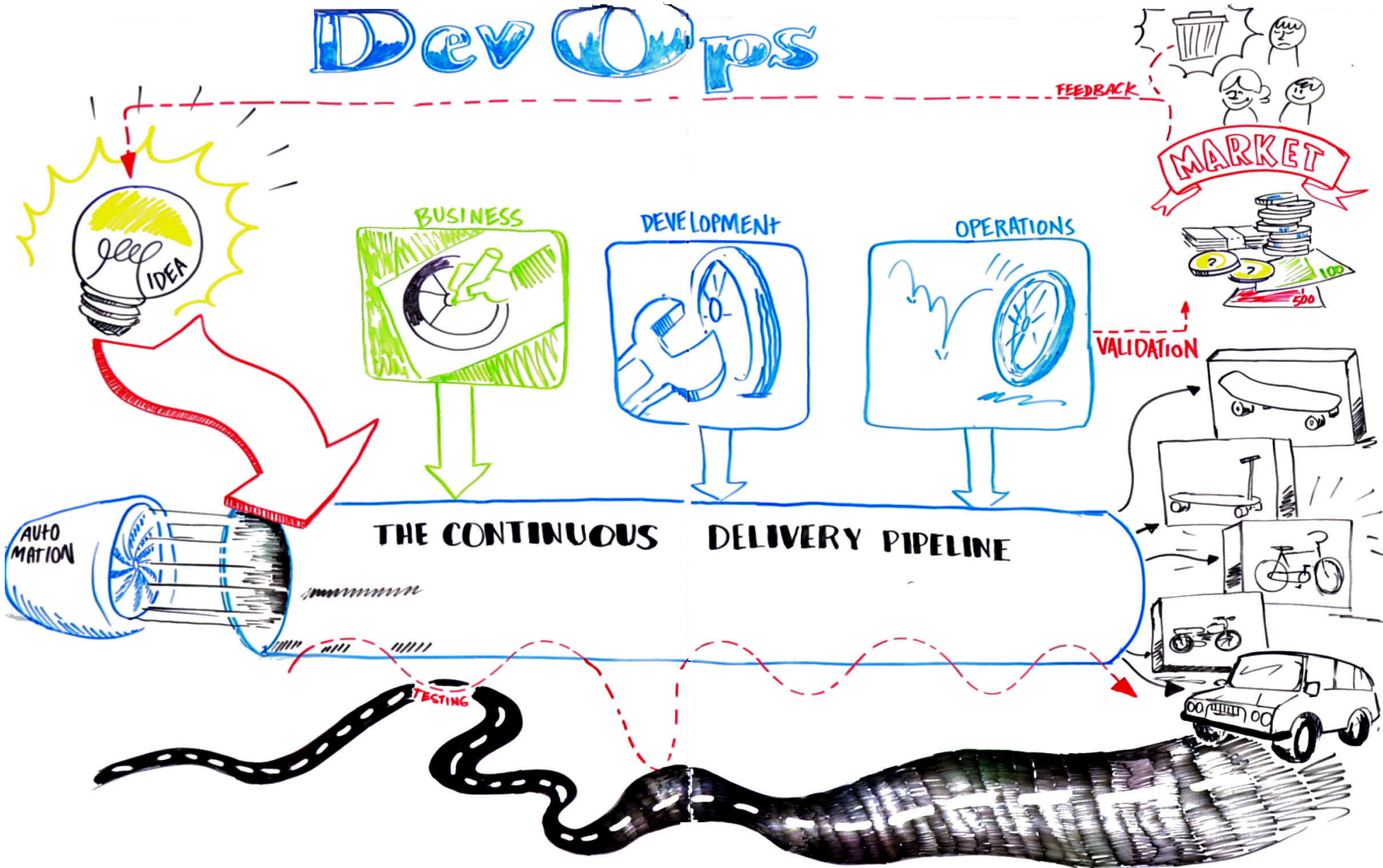
Heap      Classpath      Garbage Collection  
Garbage Collection      Libraries

**Dev**

Load      LVM      Pipes  
CICS      sed / awk

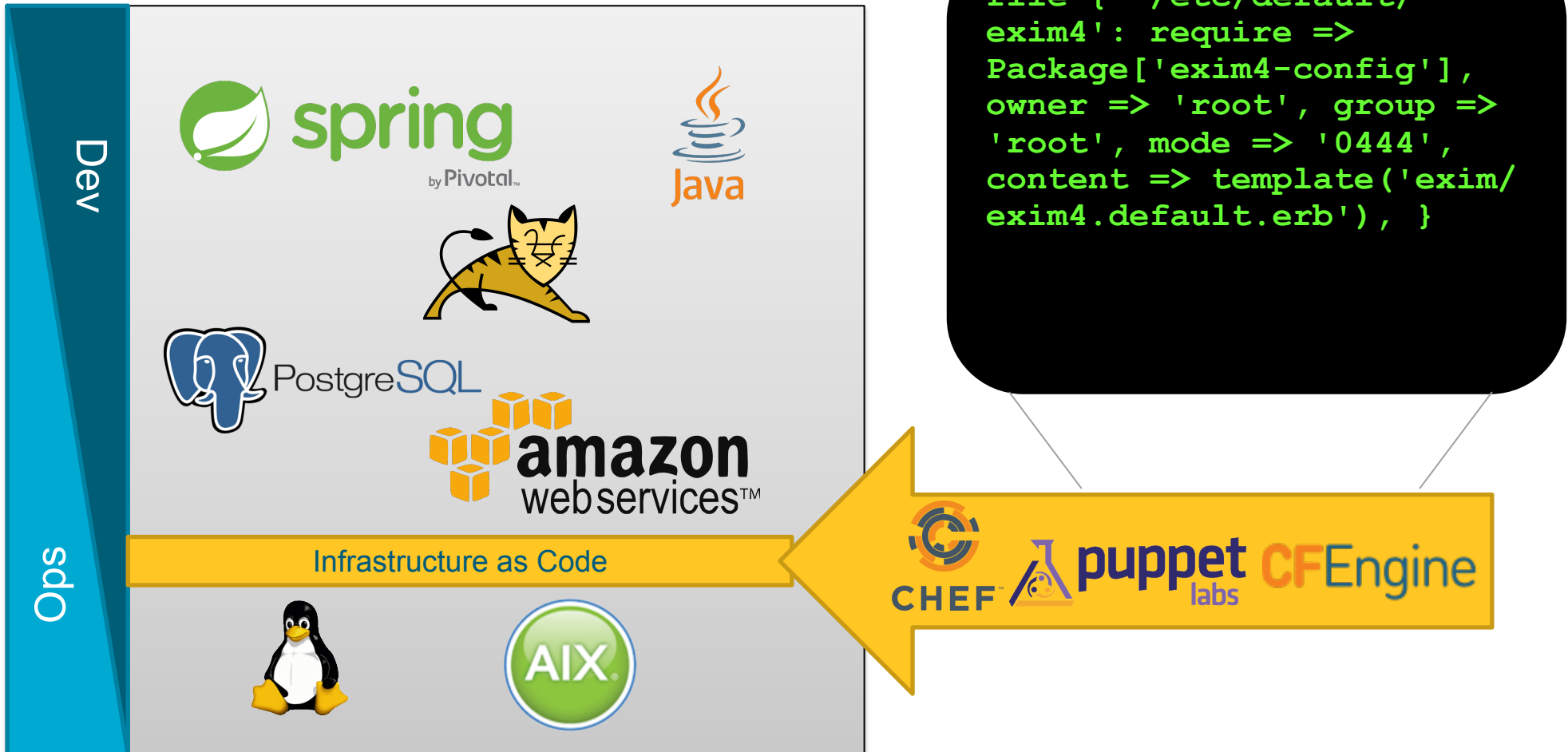
**Ops**

# DevOps



# Infrastructure as Code as common language

Application Stack



## People matter, results count.

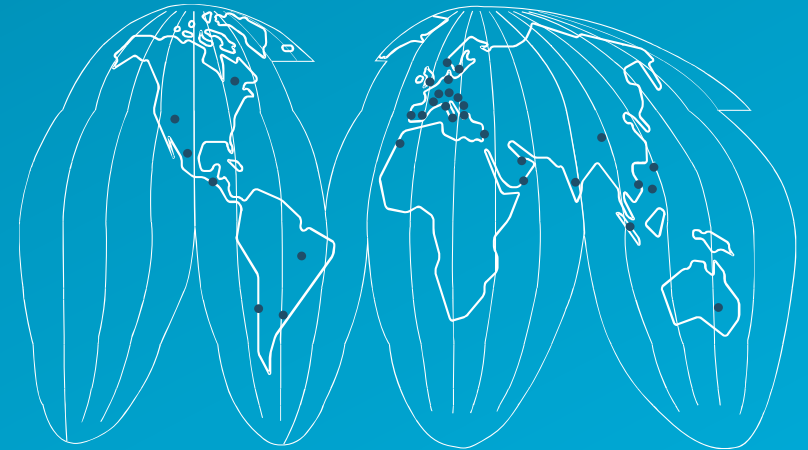


### About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

*Rightshore® is a trademark belonging to Capgemini*



[www.capgemini.com](http://www.capgemini.com)

