

# Agile Culture



# What is Agility ?

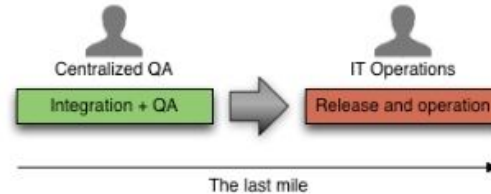
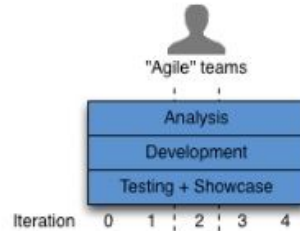
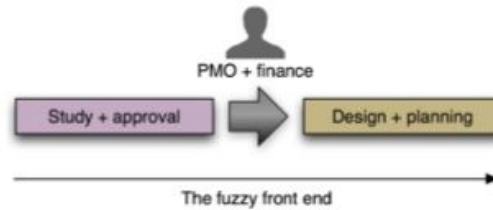
***Agility is a culture, a mindset, a philosophy.***

(It's NOT a process)

# What is Agility ?



# What is Agility ?



water-

scrum-

fall

# What is Agility ?

***Compass over map***



***Adaptive, not predictive***

- Map: you precisely define the route/way with fixed path before you start
- Compass: With compass you do not know exactly the path. You have just a rough goal.
- Compass: On the way you find out which path you take to achieve the goal.

# What is Agility ?

***It's not a time-driven approach,  
it's a value-driven approach.***

Ask yourself always before you do something:

***Does it give value to my customer ?***

Ask yourself at the end of the working day:

***What have I done today that makes my client happier than yesterday ?***

# What is Agility ?

***Doing is better than knowing.***

- Talk less, Control less and Estimate less
- Do more and learn from it!
- Facts are the base for decisions

# Pyramid of Agility





# Foundation - TRUST & COURAGE & RESPECT

- TRUST:

When I trust people, I'm not controlling them. I trust in their expertise and their ability to get the job done.

- COURAGE:

Courage to do mistakes, to phase uncertainties, to NOT know !

- RESPECT:

Each opinion is equal. There is no opinion which overrules others by default. Arguments and facts define which way to go.

# Manifesto for Agile Software Development

## Individuals and interactions over processes and tools

- Let people create the best working environment for them including tools, processes etc.
- Not the individuals who conduct and live the processes are in charge to create and build them. Very often, other people (external experts, management etc.) create processes on paper. (example Time-Manager at Toyota)
- Respect people's expertise to do the best to get the job done.
- Management is supporter, not the commander. Let the people do the job and management helps to setup the best working environment for them.

# Manifesto for Agile Software Development

## Working software over comprehensive documentation (1)

- Working software is the primary measure of progress
- Don't document status by excel sheets, traffic lights / percentages on slides or other documents
- After all, the working software is the reason for the project
- With agile approaches, project teams spend more time on development and less time on documentation/administration, resulting in a more efficient delivery of a working product

# Manifesto for Agile Software Development

## Working software over comprehensive documentation (2)

TIP:

*Stop producing a document and see who complains. If someone misses the paperwork, ask why the document is necessary. In fact, ask, “Why?” five times to get to the root reason the document is missed ( ). When you know the core reason for the report/document, see how you can satisfy that need with a streamlined process.*

# Manifesto for Agile Software Development

## Customer collaboration over negotiation (1)

Do you experienced following scenario:

- Boss/Manager A asks Boss/Manager B: How long do you need to build that software?
- Boss B: Well, my developers said, 9 months, with 10 developers full-time.
- Boss A: Oh, that's too long. We need it faster. Can't you make it faster? Maybe 3 months?
- Boss B: No, 3 months is impossible.
- Boss A: Ok, then maybe 6 months?

*...discussion continues*

# Manifesto for Agile Software Development

## Customer collaboration over negotiation (2)

The outcome of this dialog defines in the end, time and budget for a project. Also what kind of tools, what collaborators or teams and many other things will be defined by this “Basar” approach (by Negotiation).

The problem with this is, that nobody cares about the real solution. It's just important what's written into the contract.

By that the focus to solve the actual problem is lost totally.

You can adapt this scenario to other setups where people negotiate. But they should **collaborate** instead.

# Manifesto for Agile Software Development

## Customer collaboration over negotiation (3)

What would be an agile way?

- Start with small slices which delivers value to the customer
- Customer only pays for these small pieces ("Agile Contracting"). You could just pay for 1 or 2 iterations and see what's the outcome.
- After each iteration, both parties can decide whether to continue or stop the collaboration.

# Manifesto for Agile Software Development

## Customer collaboration over negotiation (4)

- Furthermore the customer can give immediate feedback and adjust the direction of the project in case it's needed.  
So the customer has direct influence how the project goes on.
- By that you work on solving the actual problem, not fulfilling a contract. YOU REALLY COLLABORATE !
- Collaborations like this builds up TRUST and the customer is willing to pay more for longer periods
- You **CONVINCE BY QUALITY, NOT BY CONTRACT !**
- Since the customer just pays (much) less, the risk is lower and the customer can directly see what she/he gets for the money.



# Manifesto for Agile Software Development

## Responding to change over following a plan

- Plans are PREDICTIVE and Agile Working is ADAPTIVE !
- What's the problem with plans (including contracts)?
  - We must predict the future and humans are really bad in predicting the future! We are just guessing. And running a business based on guessing seems not to be the best way. See also
- Since the requirements will change for sure, the plan has to change too. This will happen very often. So why maintaining a plan at all when its always changing?
- Instead, focus on changing the team, skills and software to solve the actual problem.
- Adjusting/Adapting to the volatile needs of customer and market is the core of BEING AGILE.
- When you cannot change easily and fast to new requirements (these can affect team, skills and of course requirements of the software), you cannot fulfill the real needs of the market. A plan or a contract is very often static and fixed which is the opposite of agile and flexible.

# Principles - Principles of Agile Manifesto

# Doing Agility

## What to do?

- Find out where you are
- Take a small step towards your goal
- Adjust your understanding based on what you learned
- Repeat

## How to do it?

- When faced with two or more alternatives that deliver roughly the same value, take the path that makes future change easier

# Processes

## Process Frameworks can support Agility

- eXtreme Programming
- SCRUM
- KANBAN
- Lean
- “Free-Style”
- ...whatever fits for you

See also [Agile Process Frameworks](#)

# Agility - Must-Watch

- [Agile is dead - Allan Holub](#)
- [Agile is dead - Long live Agility by Dave Thomas](#)
- [No Estimates by Allan Holub](#)
- [Explaining Agile - Martin Fowler and Neal Ford](#)