

# IF1006 – DevOps Software delviery way: SDLC, DevOps and Pipelines

Fish

@fisholito

jfsc@cin.ufpe.br



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



# What did we learn until here?

---

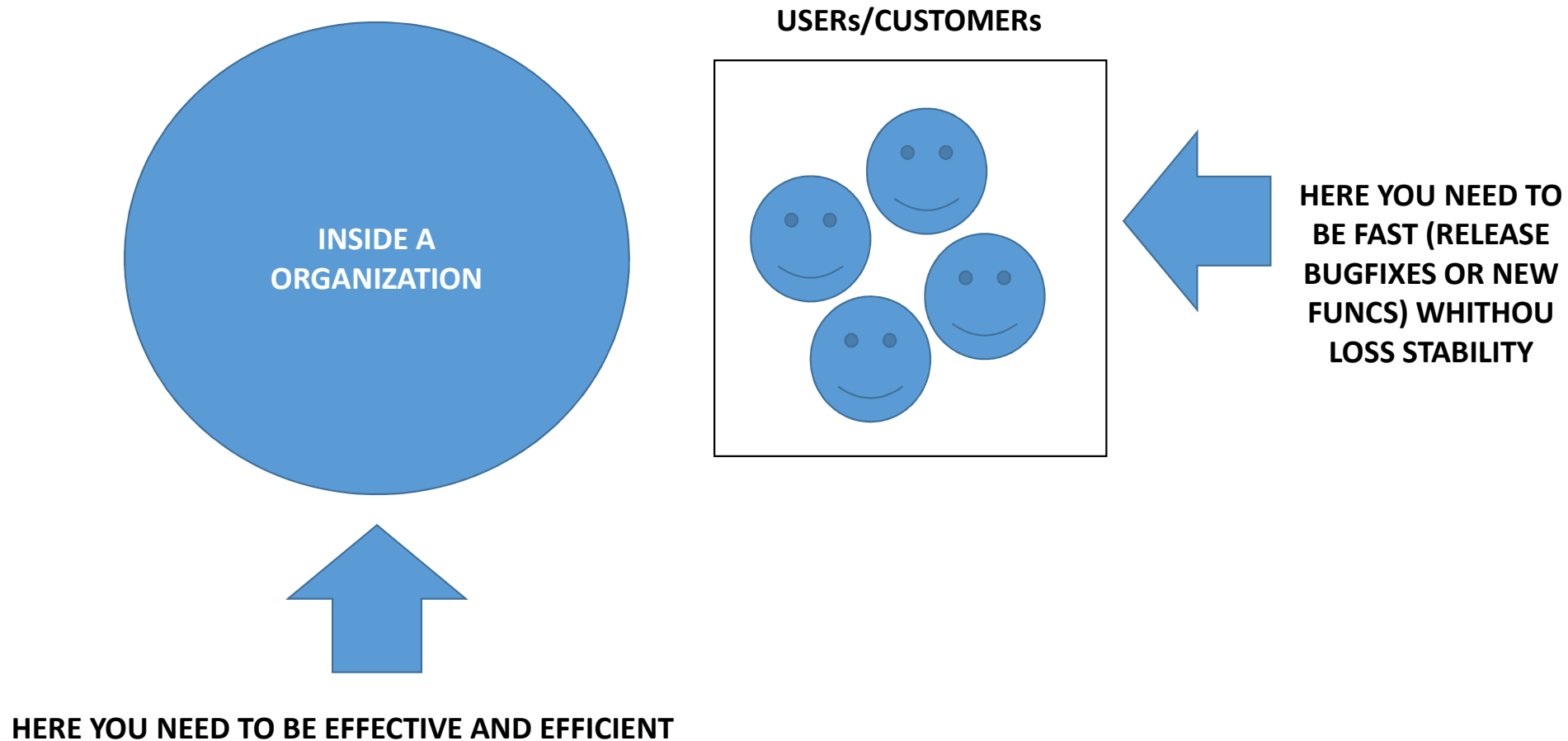
- Chaos in delivery process
- Complexity of software development
- The SDLC should be (re)analysed
- We need to identify our misalignment

# What is DevOps, again?

---

A set of practices to help organizations to deliver software fast without loss quality – [Culture]

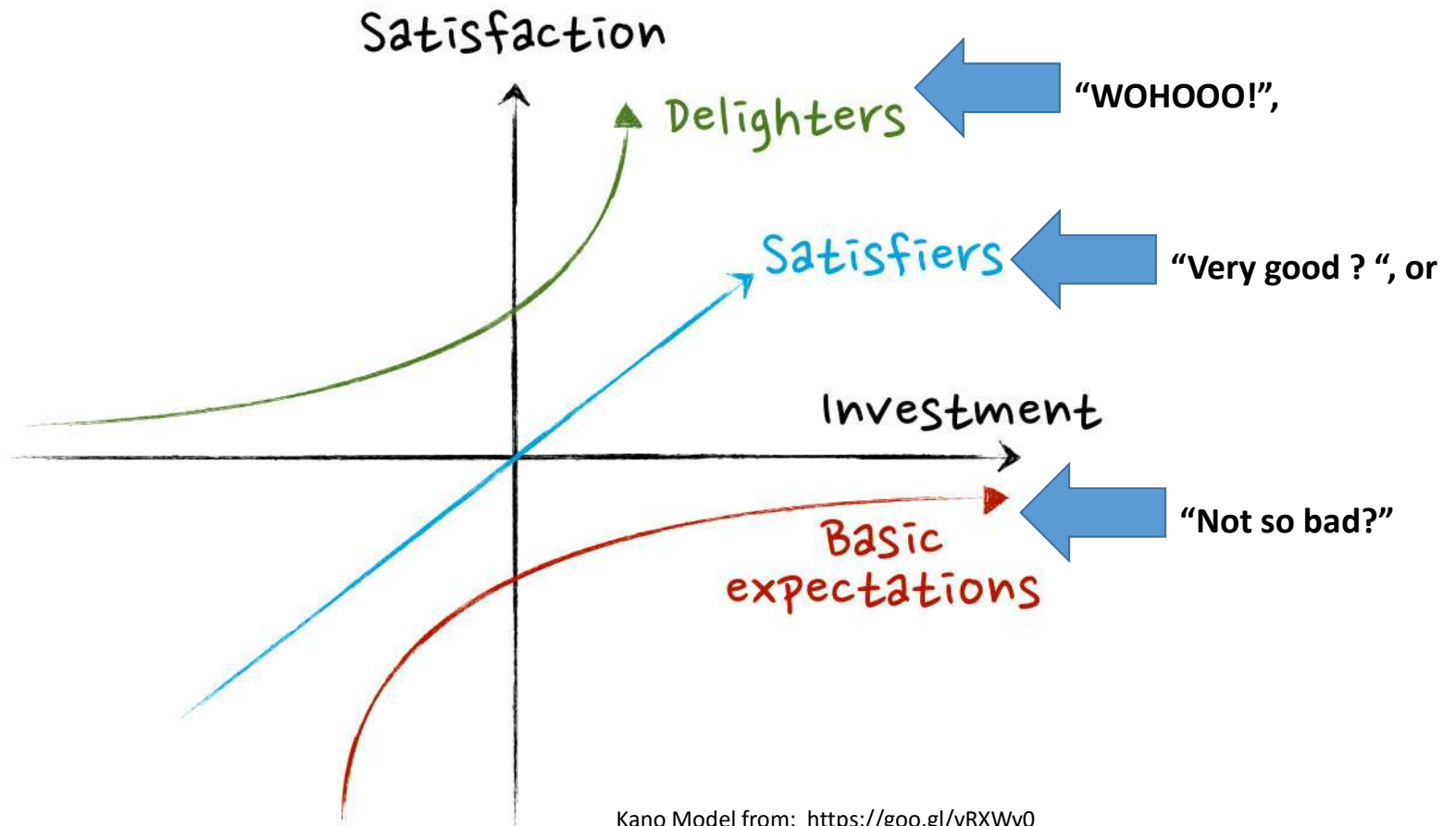
# DEVOPS PRACTICES EXPECTATIONS: ORGANIZATION AND USER



# Starting a DevOps implementation: knowing yourself

What are your customer saying about your product? And what do you want they say about that?

How resilient are your business?



## Seeing the whole;

- Look at SDLC and its stakeholders as whole and ask: How could I help my organization to deliver software fast without loss quality?

*First, we need **people believe** that they can deliver or receive (customer side) the most valuable software ever. Second, we need to **support\*** a development environment (place to work or develop) through tools and processes which they can deliver/receive valuable software.*

\* Be careful, DevOps environments can be expensive

# The DevOps Approach for this course

---



## Spending effort on an agile mindset

---

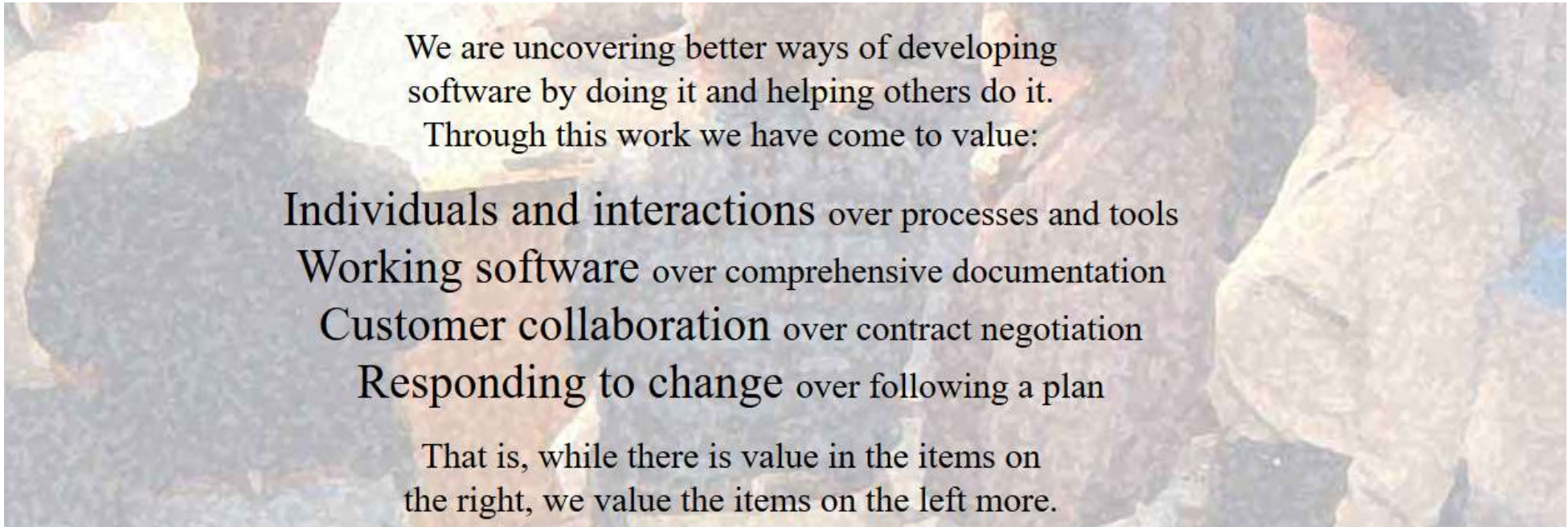
- “Agile is the ability to create and respond to change in order to succeed in an uncertain and turbulent environment.”

AGILE ALLIANCE



# Agile values

---



We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

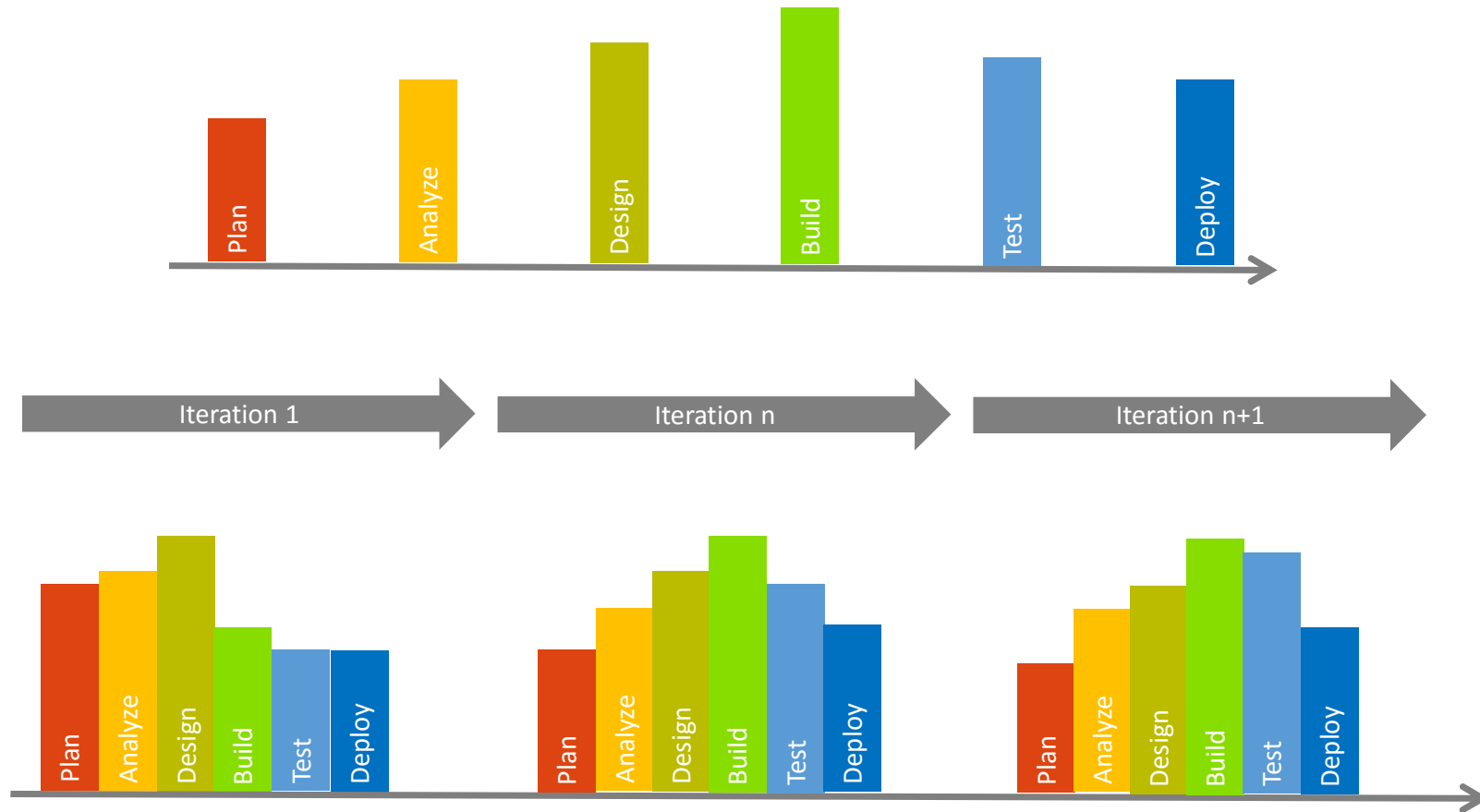
**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

# Agile principles

- 1** *Satisfy Customer with valuable software*
- 2** *Change are welcome even later in Project.*
- 3** *Deliver working software frequently (weeks)*
- 4** *Bus+Dev working daily throughout the project*
- 5** *Trust and support the team*
- 6** *Face-to-face conversation*
- 7** *Working software is the primary measure of progress*
- 8** *Sustainable development*
- 9** *Continuous attention to technical excellence and good design*
- 10** *Simplicity – maximizing the amount of work not done*
- 11** *Self-organizing teams product more quality*
- 12** *Reflects on how to become more effective, and tunes and adjusts*

# The main difference



# Five keys to a successful Google team



Agile is an umbrella – methods are implementations

---

- Lean software development
- Scrum
- FDD
- DSDM
- Crystal

# The Scrum framework

