

# Last time

- Software process model determines the order of the stages involved in software development and evolution
- We need processes to scale up with complexity
- There are several types of process models: waterfall, incremental, spiral...
- There is no silver bullet, the context impacts process
- Process modeling helps communicate and share knowledge on how development teams perform their tasks



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team



## Plan and document models failed often

- Often missing cost, schedule, and quality target
- P&D requires extensive documentation and planning



Lecture 6 - Introduction to  
Agile Development

**TDresearchteam**  
Technical Debt Research Team



How can we build software effectively without  
relying on extensive documentation?

# Introduction to Agile Development

Dr. Rodrigo Spínola



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team



## Agenda

6

- Agile Development Model
- Agile Then and Now
- When opt between traditional and agile?



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team

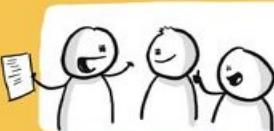




menti.com + 7944 6077

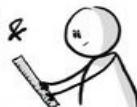
## What is agile software development?

### 4 KEY VALUES OF AGILE SOFTWARE DEVELOPMENT



INDIVIDUALS &  
INTERACTIONS

OVER



PROCESSES &  
TOOLS



WORKING  
SOFTWARE

OVER



COMPREHENSIVE  
DOCUMENTATION



CUSTOMER  
COLLABORATION

OVER



CONTRACT  
NEGOTIATION



RESPONDING TO  
CHANGE

OVER



FOLLOWING  
A PLAN



## 12 OFFICIAL AGILE PRINCIPLES



10

## Agile Development Model

- Embraces changes as a fact of life: continuous improvement vs. strict phases
- Developers continuously refine working until customers happy, with customer feedback on each iteration (every ~1 to 4 weeks)
- Agile emphasizes Test-Driven Development (TDD) to reduce mistakes, written down user stories to validate customer requirements

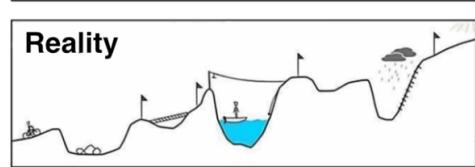
Why these are important?

Do you remember it?

Your plan



Reality



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team

 **VCU**  
Computer Science  
College of Engineering

11

# Agile Development Processes

**What are the common misunderstandings about agile methodology represented in this figure?**

We still have planning and documentation!

Agile is not about doing it by yourself or doing it any way you want.



© Scott Adams, Inc./Dist. by UFS, Inc.



Lecture 6 - Introduction to Agile Development

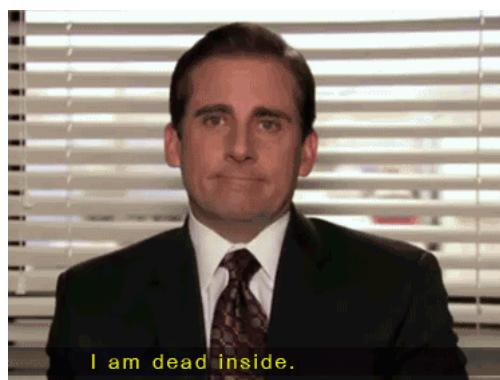
**TDresearchteam**  
Technical Debt Research Team



12

But....

- Do NOT fool yourself
- You still needs a process, a lightweight one, but still a process
- Regardless a plan-and-document driven or agile, you will need a process to have a north



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team



13

# Extreme Programming

- If short iterations are good, make them as short as possible (weeks vs. years)
- If simplicity is good, always do the simplest thing that could possibly work
- If testing is good, test all the time. Write test code before you write the code to test
- If code reviews are good, review code continuously, by programming in pairs, taking turns looking over each other's shoulders



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team

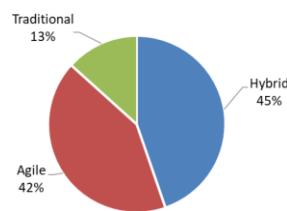


14

# Agile Then and Now

- Controversial in 2001
  - "... yet another attempt to undermine the discipline of software engineering... nothing more than an attempt to legitimize hacker behavior." - Steven Ratkin, "Manifesto Elicits Cynicism", IEEE Computer, 2001.
- Accepted in 2013
  - 2012 study of 66 projects found majority using Agile, even for distributed teams
- And in 2022...
  - Study with about 653 software organizations from 6 countries

Robert Ramač, Vladimir Mandić, Nebojša Taušan, Nicollis Rios, Sávio Freire, Boris Pérez, Camilo Castellanos, Dario Correal, Alexia Pacheco, Gustavo Lopez, Clemente Izurieta, Carolyn Seaman, Rodrigo Spinola, Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry, Journal of Systems and Software, Volume 184, 2022, 111114, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2021.111114>.



Lecture 6 - Introduction to Agile Development

**TDresearchteam**  
Technical Debt Research Team





Software Development  
Methodology

# What is Agile?



When should we choose  
between traditional and agile approaches?

17

**Yes -> Plan and Document**

**No -> Agile**

- Is a detailed specification required?
- Are customers unavailable?
- Is the system to be built very large?
- Is the system to be built very complex (e.g., real time)?
- Are you using poor software tools?
- Is the system to be built subject to regulation?
- Is the team part of a document-oriented culture?
- Does the team have very poor programming skills?

Summarizing,  
sometimes we need  
more control to  
reduce risks



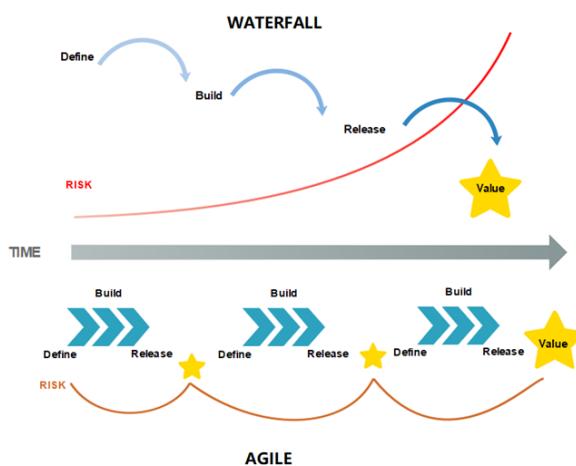
Lecture 6 - Introduction to  
Agile Development

**TDresearchteam**  
Technical Debt Research Team



18

## Plan and Document vs. Agile



Lecture 6 - Introduction to  
Agile Development

**TDresearchteam**  
Technical Debt Research Team



# Summary

- Agile embraces changes as a fact of life
- Very short iterations
- We still have a process, a lightweight one, but a process
- The focus is on the value (working software) to customers



Lecture 6 - Introduction to  
Agile Development

**TDresearchteam**  
Technical Debt Research Team



Class is  
over,  
questions?

# Introduction to Agile Development

Dr. Rodrigo Spínola



Lecture 6 - Introduction to  
Agile Development

**TDresearchteam**  
Technical Debt Research Team

