Software Processes

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

AMOS C01

Agenda

1. Product management
2. Engineering management
3. Software development
4. Quality assurance
5. Software process models
6. Plan-driven development
7. Agile methods

1. Product Management

Product Management

- **Product management**
  - Is the **management** [1] of a company's products
  - Along the **product's life-cycle**
  - Across the **product portfolio** (if any)

[1] The purposeful act, manner, or practice of handling, supervising, or controlling [products, product life-cycles, product portfolios]

What?

What first?

Two Sides of Product Management

1. **Strategic product management**
   - Focuses on assessing and defining the opportunity
   - Responsible for the Marketing Requirements Document

2. **Technical product management**
   - Focuses on defining the product and its features
   - Responsible for the Product Requirements Document

Example Processes and Artifacts

Processes	Artifacts
Opportunity Assessment	Marketing Requirements Document (MRD)
Product Specification	Product Requirements Document (PRD)
Product Roadmapping	Internal Product Roadmap External Product Roadmap
Release Planning	Release Plan

Products vs. Projects

	Product	Project
Strategic	Strategic product manager	Project leader Senior business analyst
Technical	Technical product manager	(Junior) business analyst

Video From “The Pentagon Wars” [1]

The New Bradley Design

(Ten years in the life of a project manager)

[1] See https://youtu.be/r0op8e0LuoU

Video Lessons

- Multiple stakeholders
  - Bargaining leads to suboptimal results
- Meddling stakeholders
  - Intervening in the tank design process
- Unclear market
  - From US military to foreign markets
- Cost explosion
  - With changing requirements, costs explode
- Inconsistent requirements
  - From fast and small to big with firepower
- Changing requirements
  - Lack of focus invalidates prior work
- Feature creep
  - From troop carrier to tank

2. Engineering Management

Engineering Management

Engineering management is the **management**[1] of a company's **product development process** [along the **product life-cycle**] [across a **product portfolio**]. [DR]

[1] The purposeful act, manner, or practice of handling, supervising, or controlling [developers and engineering processes]

Engineering Management (Simplified)

Who?

By when?

Example Processes and Artifacts

Processes	Artifacts
Release Planning	Release Plan
Resource Allocation	Project Plan Task Board
Outsourcing	Budget Project Plan
Project Retrospective	Note Book

3. Software Development

Software Development

Software development **(programming)** is the process of turning product requirements into working software. [DR]

Software Development (Simplified)

How?

How fast?

Processes and Artifacts

Processes	Artifacts
Effort Estimation	Release Plan
Programming	Source Code

4. Quality Assurance

Quality Assurance

Quality assurance is the process of assuring that the software being developed **has a defined quality.** [DR]

Quality Assurance (Simplified)

Releasable?

Good enough?

Quality

The **quality** of a software system is the degree to which it **conforms** to its **stakeholders' expectations**. [DR]

Stakeholders and Expectations

- Product management
  - Functional requirements
  - Non-functional requirements
- Engineering management
  - Product quality
  - Maintainability and evolution
- Support and operations
  - Operations costs, usability
  - Non-functional requirements

Processes and Artifacts

Processes	Artifacts
Automated Test Design and Implementation	Test Plan Test Code
Manual Test Design and Implementation	Test Plan Test Script
Release Sign-off	Email
Release Packaging	Software Release

Quiz on Organizational Issues

1. Where in the organizational chart to put product management?
   a) Sales and marketing
   b) Engineering
   c) Stand-alone

2. Where in the organizational chart to put quality assurance?
   a) Engineering
   b) Product management
   c) Stand-alone

5. Software Process Models

Software Process Model [DR]

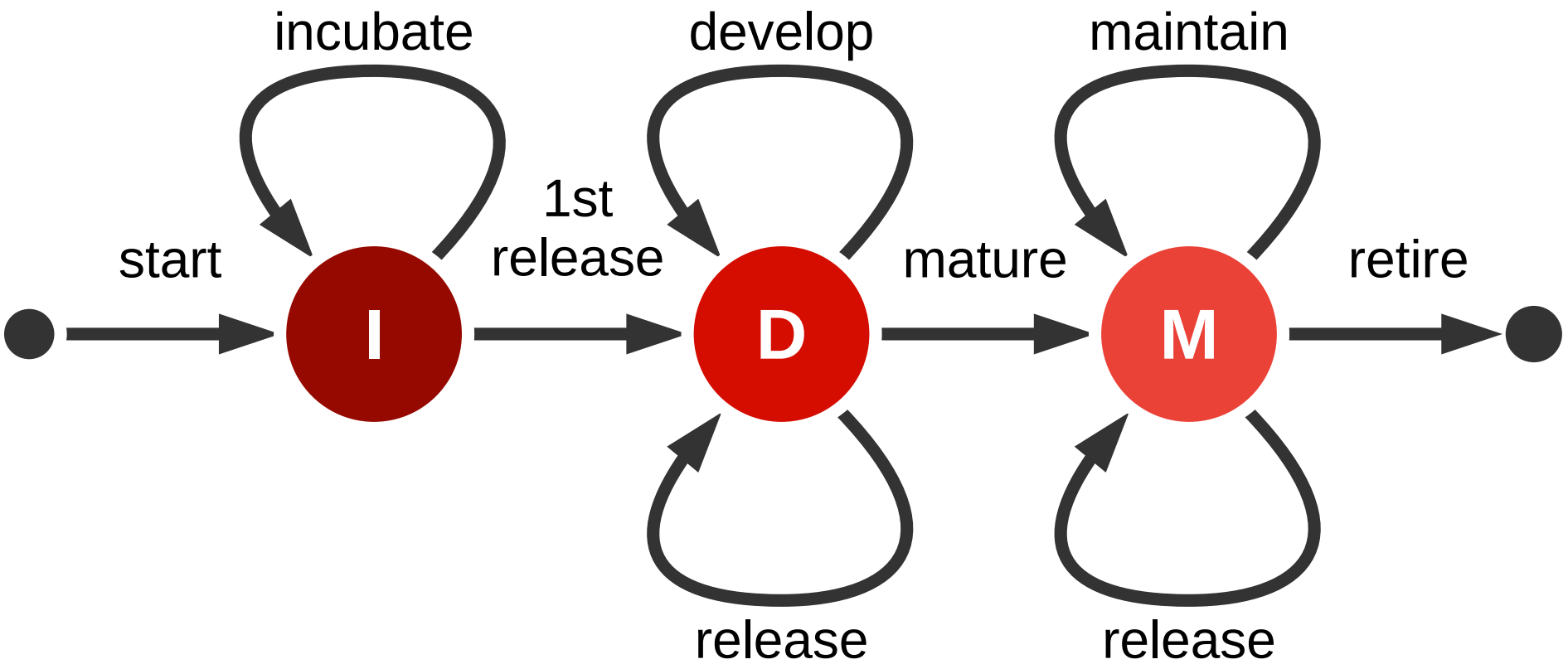
- A **software process**
  - Is a **process** performed with goal of creating and evolving software
- A **software process model**
  - Is a **model** of a **software process**
- Software process **model elements**
  - Comprises **roles, practices, and artifacts** that describe the **valid software process instances**

Project Management Frameworks

Key Activities in Software Engineering

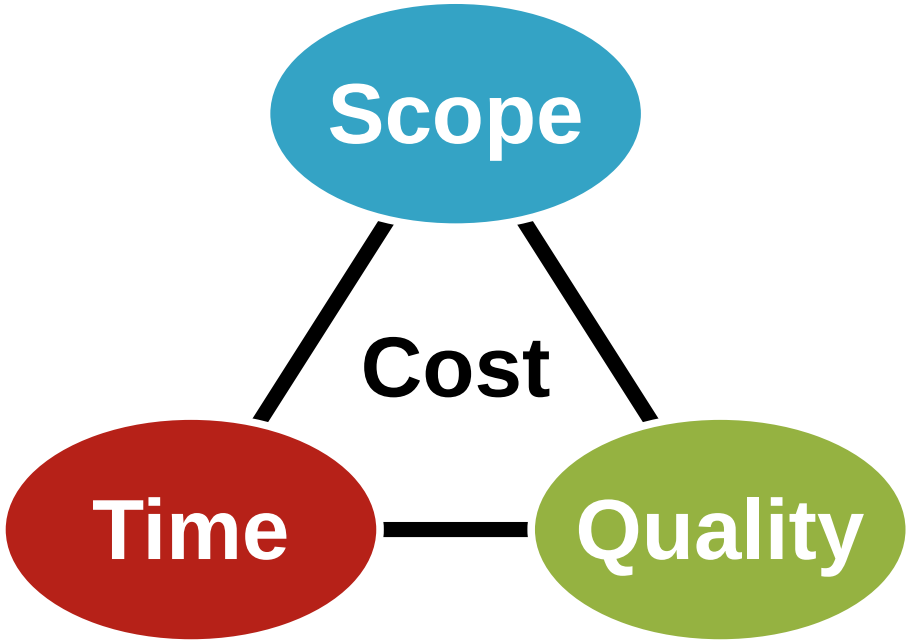
1. Planning
2. Execution
3. Review
4. Release

Example of a Software Product Life-Cycle



I: Incubation
D: Development
M: Maintenance

(One Version of) The Magic Triangle



Cost is usually assumed fixed (defined team).

Because “adding manpower to a late project makes it later.” [B75]

6. Plan-driven Development

Plan-Driven Development

- Linear, phase-oriented, software process models
  - Intend to minimize risk through up-front planning
  - Expect only one iteration, start to finish, not many
  - Equate phases with activities
- Examples: Waterfall, V-Modell, RUP

The Waterfall Model [R04]

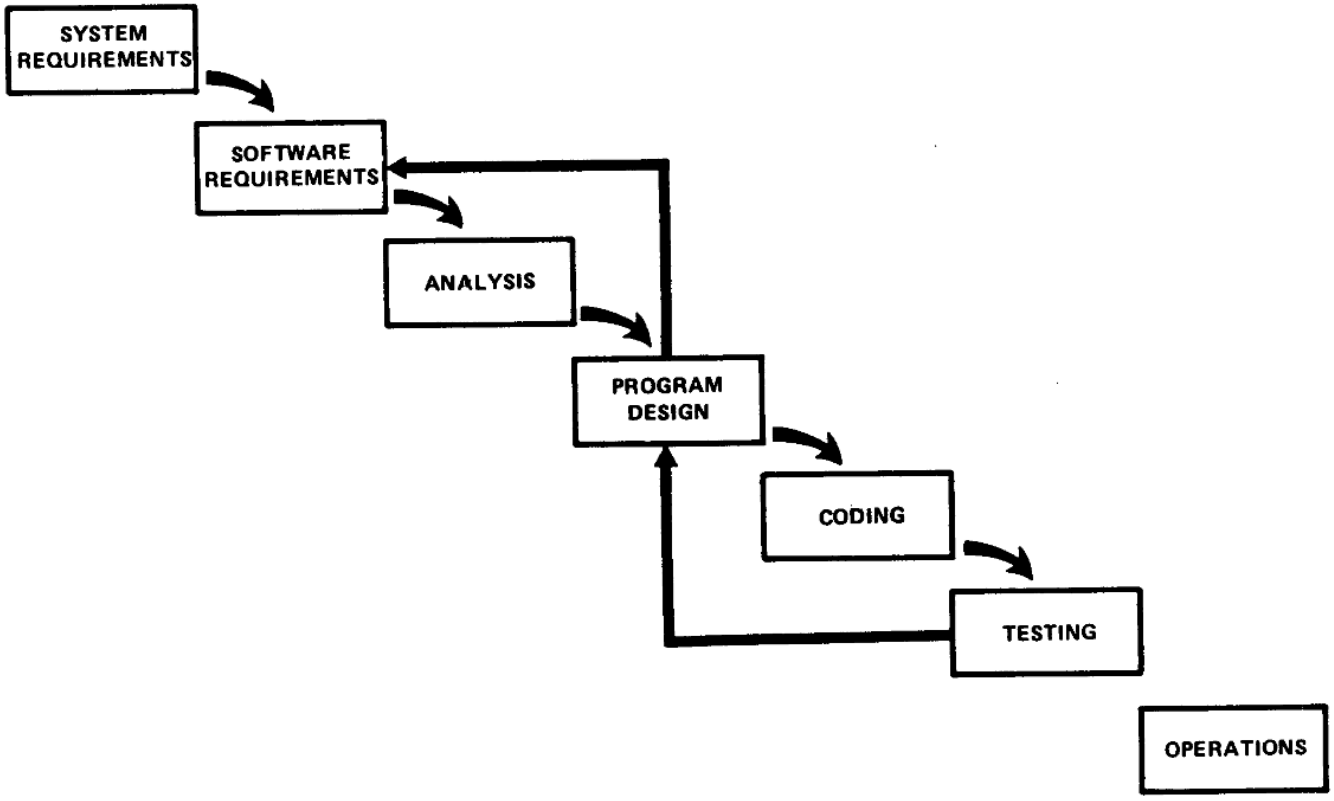


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

Aphorisms on Predicting and Planning

- **“Prediction is very difficult, especially about the future.”**
  - Attributed to Niels Bohr, date unknown
- **“Kein Plan überlebt die erste Feindberührung.”**
  - Helmuth (Karl Bernhard) von Moltke, date unknown
- **“Plans are worthless, but planning is everything.”**
  - Dwight D. Eisenhower, Nov 14, 1957

Lesson from Plan-Driven Development

Phases ≠ Activities

(Activity = performing a practice)

7. Agile Methods

Video on Predictability of Processes

[1] See https://youtu.be/D7rbiLNf-JI

Video Lesson

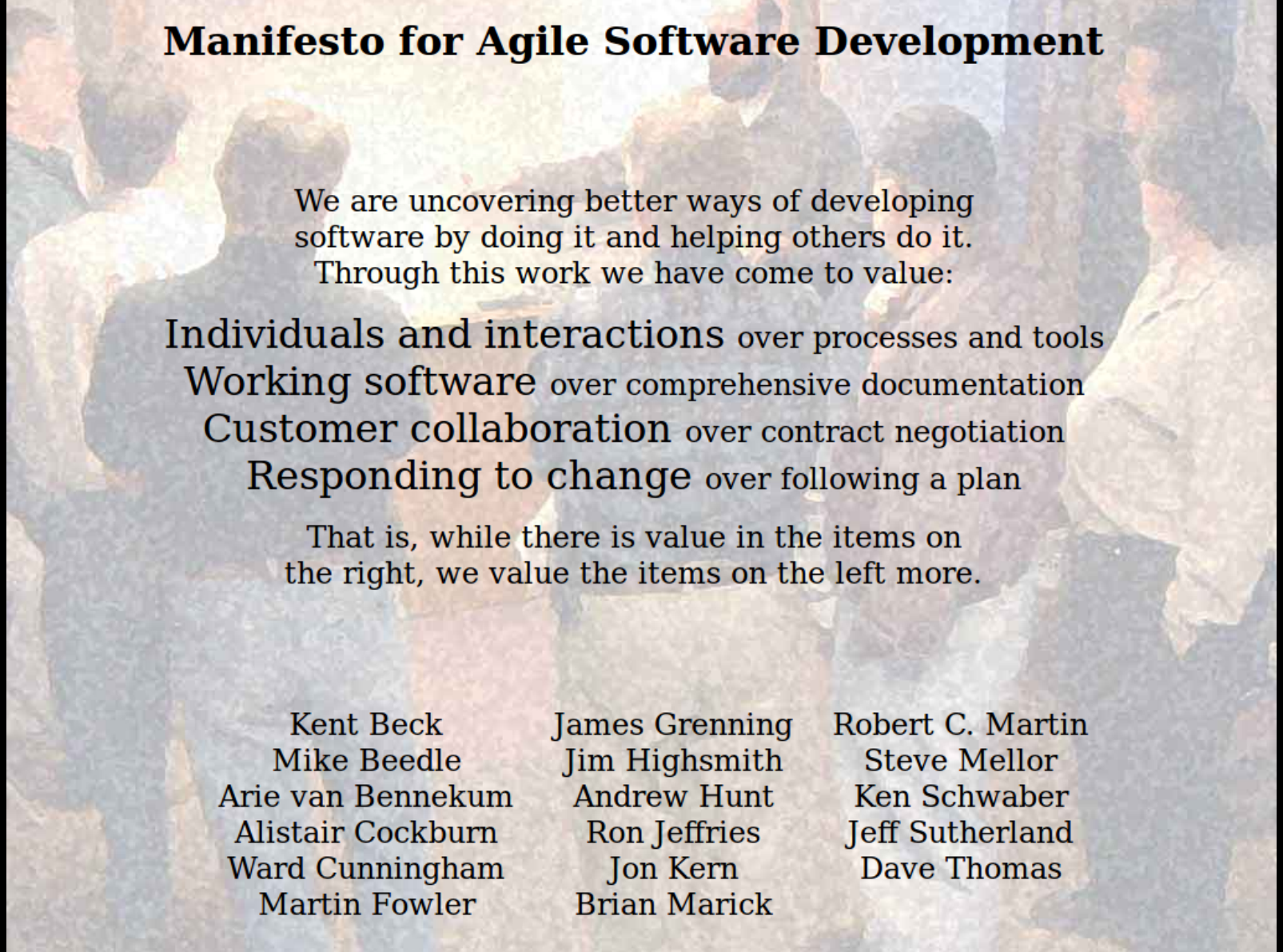
- It is impossible to predict such a flight
  - Little bumps on the way have big consequences
  - Little wind gusts will get you way off track
- Executing a plan without steering is dangerous
  - There is no way to ensure you will achieve the desired outcome
  - Belief in flawless execution is, well, flawed and risky
- Also see “the making of Megawoosh” excerpt
  - See https://youtu.be/_n065KE00J0

Agile Methods

- Invented during the late 1990ties
  - In response to failure of plan-driven methods
  - Driven by consultants as a significant business opportunity
- Repeated iteration over short linear process models
  - Defined equal-length iterations with deliverables
  - Consistent involvement of users for feedback

Examples of Agile Methods

- Scrum
- XP (eXtreme Programming)
- Adaptive Software Development
- The (set of) Crystal Methods
- Feature-Driven Development
- Pragmatic Programming



Manifesto for Agile Software Development

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.

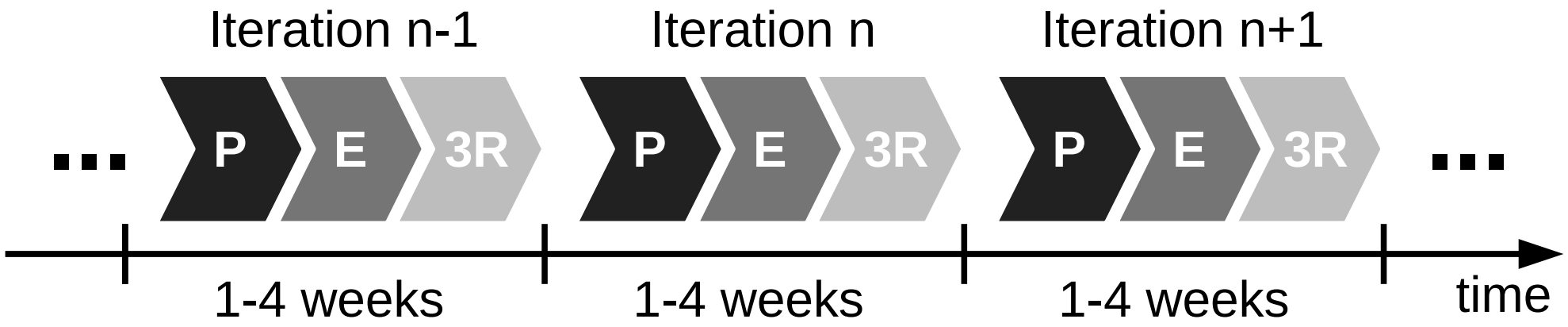
Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Principles of the Agile Manifesto [A01]

1. **Individuals and Interactions**
   (over Processes and Tools)
2. **Working Software**
   (over Comprehensive Documentation)
3. **Customer Collaboration**
   (over Contract Negotiation)
4. **Responding to Change**
   (over Following a Plan)

Agile Development Process

- Succession of **equal-length** **iterations** (“time-boxes”)
- Intervention points are during planning and review
- User feedback only available during review

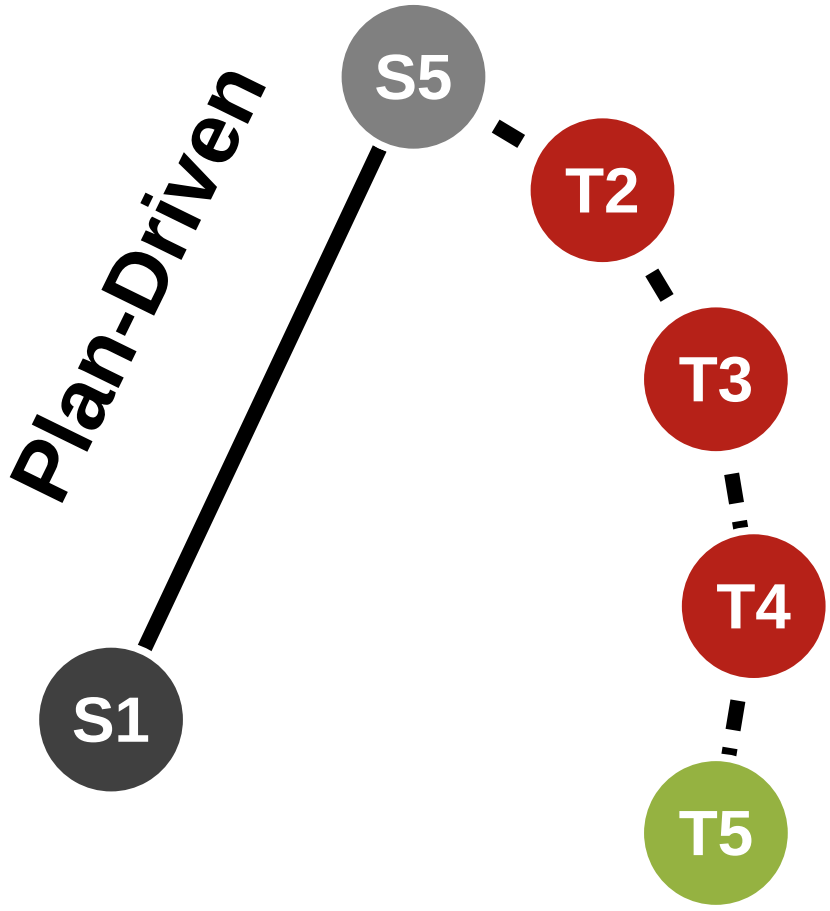


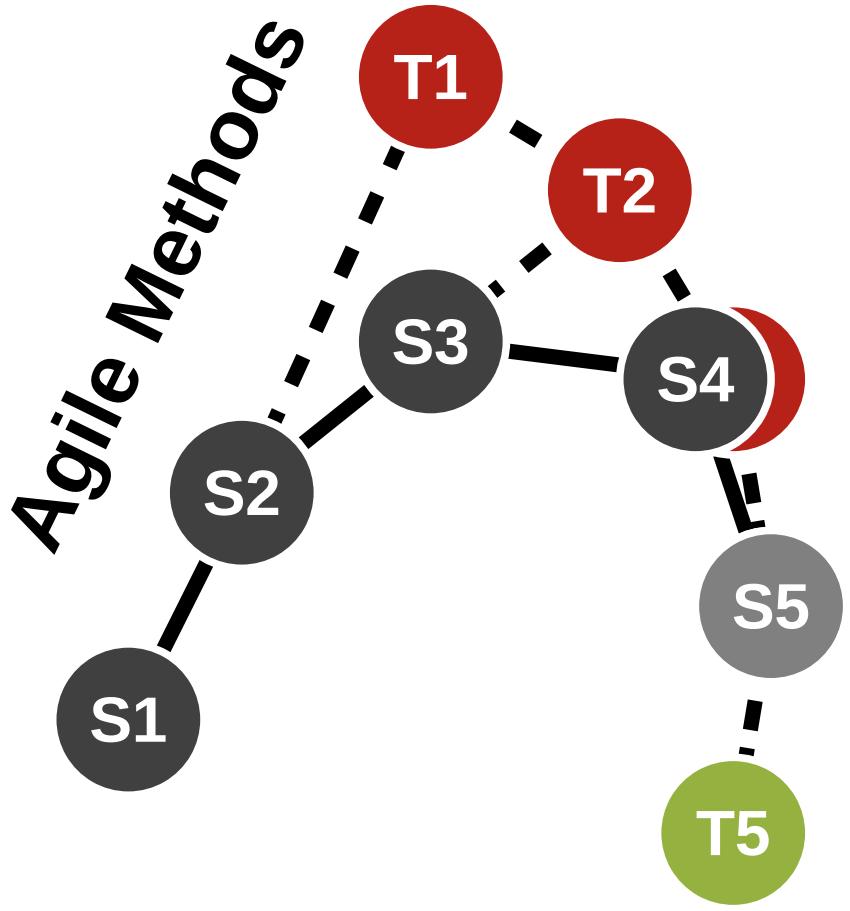
P: Planning
E: Execution
3R: Review, release, and retrospective

Short Iterations and User Feedback

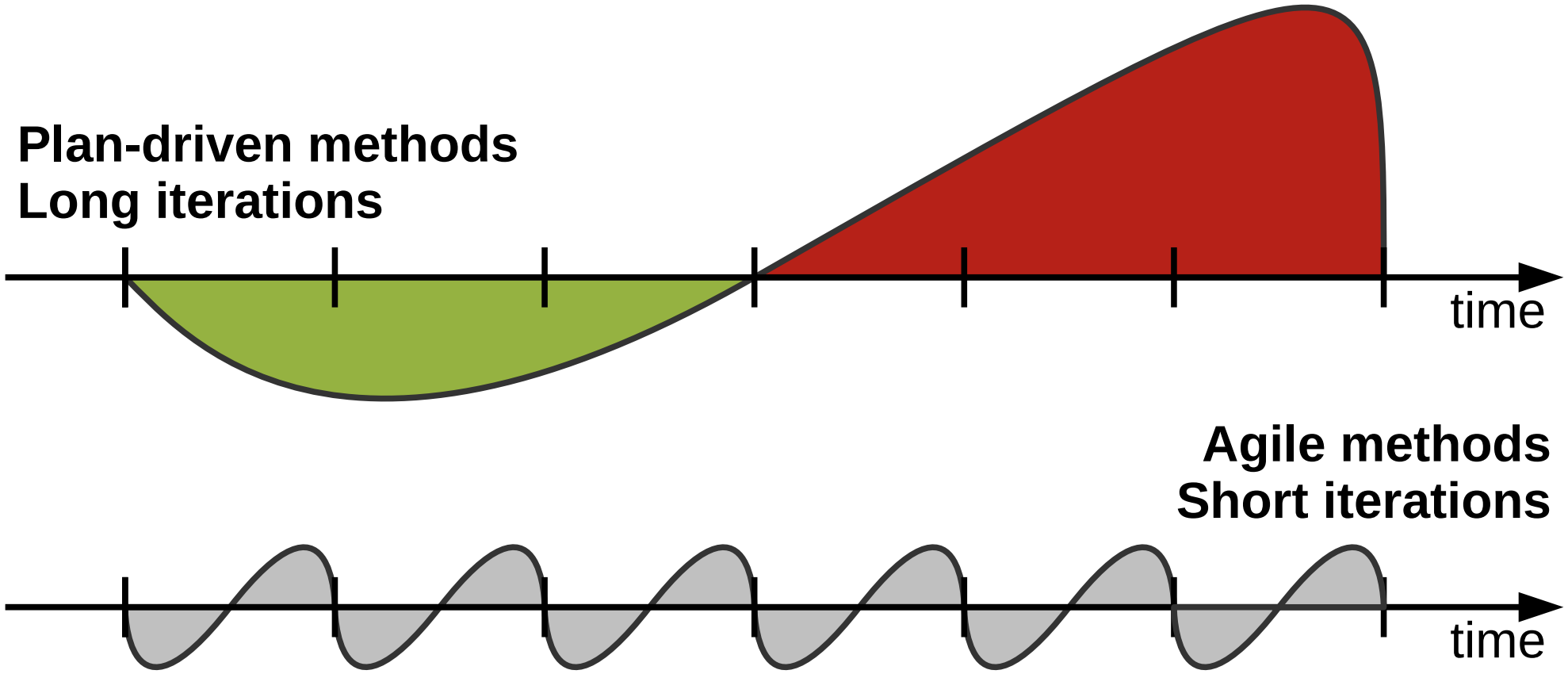
- Short iterations
  - Short iterations lead to focus on high-value features first
  - Established well-worn rhythm is sustainable, avoids burnout
  - Partial functionality is better than none
- User feedback
  - User feedback helps team steer product to meeting needs right
  - Feedback loop ensures that problems surface early
  - Feedback helps recognize and realize new innovative features

Plan-Driven vs. Agile Processes





Plan-Driven vs. Agile Work Rhythms



Agile Methods and Discipline

Agile methods
are high-discipline

(more so than plan-driven methods)

Quiz on Types of Projects

1. Which process model fits Fixed-Price-Projects?
   a) Plan-driven process
   b) Agile methods process
   c) Open source process

2. Which process model fits Time-and-Materials-Projects?
   a) Plan-driven process
   b) Agile methods process
   c) Open source process

3. Which process model fits Inter-Firm-Collaboration-Projects?
   a) Plan-driven process
   b) Agile methods process
   c) Open source process

Summary

1. Product management
2. Engineering management
3. Software development
4. Quality assurance
5. Software process models
6. Plan-driven development
7. Agile methods

Thank you! Questions?

dirk.riehle@fau.de – https://oss.cs.fau.de

dirk@riehle.org – https://dirkriehle.com – @dirkriehle

Legal Notices

- License
  - Licensed under the CC BY 4.0 International license
- Copyright
  - © 2010-2022 Dirk Riehle, some rights reserved