

Agile Processes

Dirk Riehle, Univ. Erlangen

AMOS B03

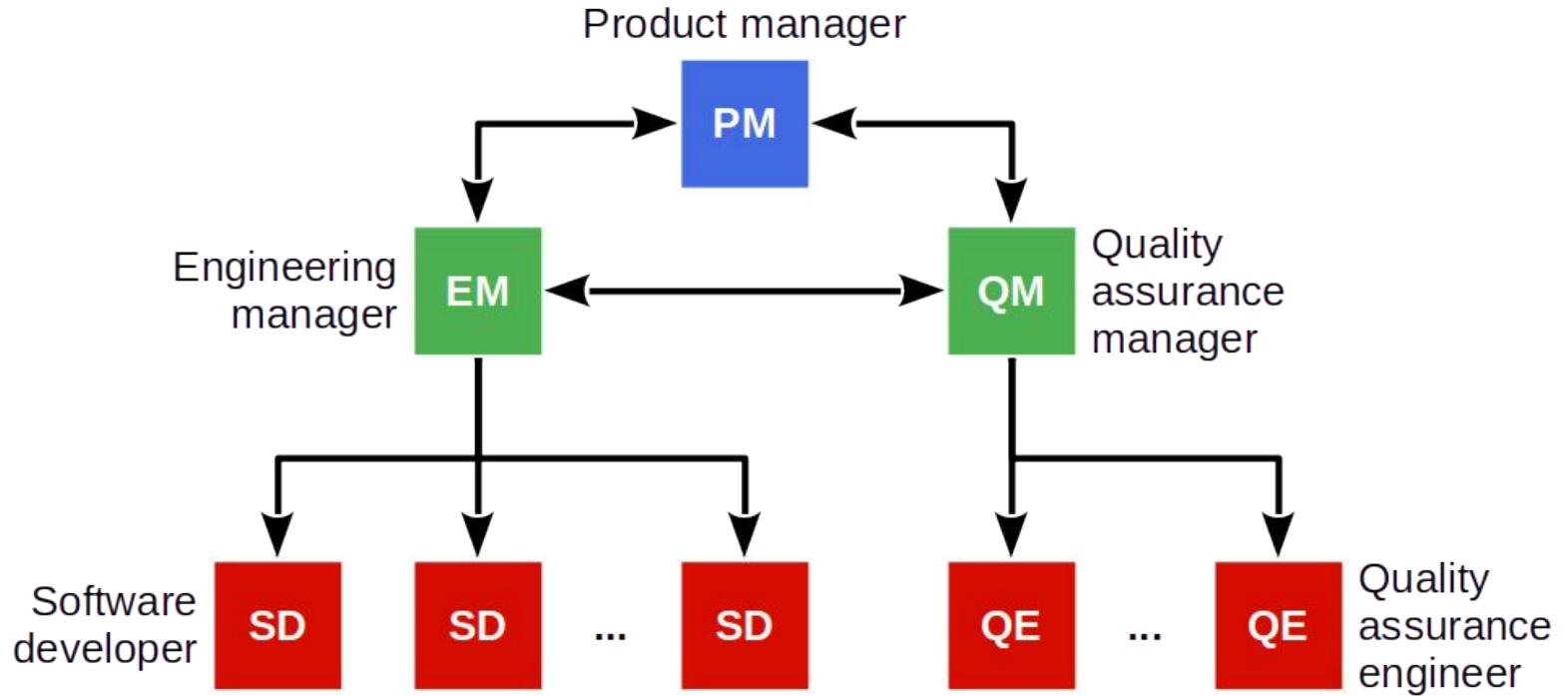
Licensed under [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

Agenda

1. Software development
2. Plan-driven development
3. Agile methods
4. Scrum

1. Software Development

Typical Software Development Organization



Positions and Job Descriptions

Product manager / product management is responsible for

- Managing a company's products along its life-cycle across a portfolio

Engineering manager / engineering management is responsible for

- Managing the development of the products (along / across)

Software developer / software development is responsible for

- Designing and implementing the products (along / across)

Quality assurance is responsible for

- Ensuring that the quality of the products meets the expectations

What / Who / How?

Product managers are responsible for

- What needs doing

Engineering managers are responsible for

- Who gets to do it and when

Software developers are responsible for

- How it gets done and how long it will take

Products vs. Projects

Products have a life-cycle; may live forever

- Products are developed for a market (many customers)

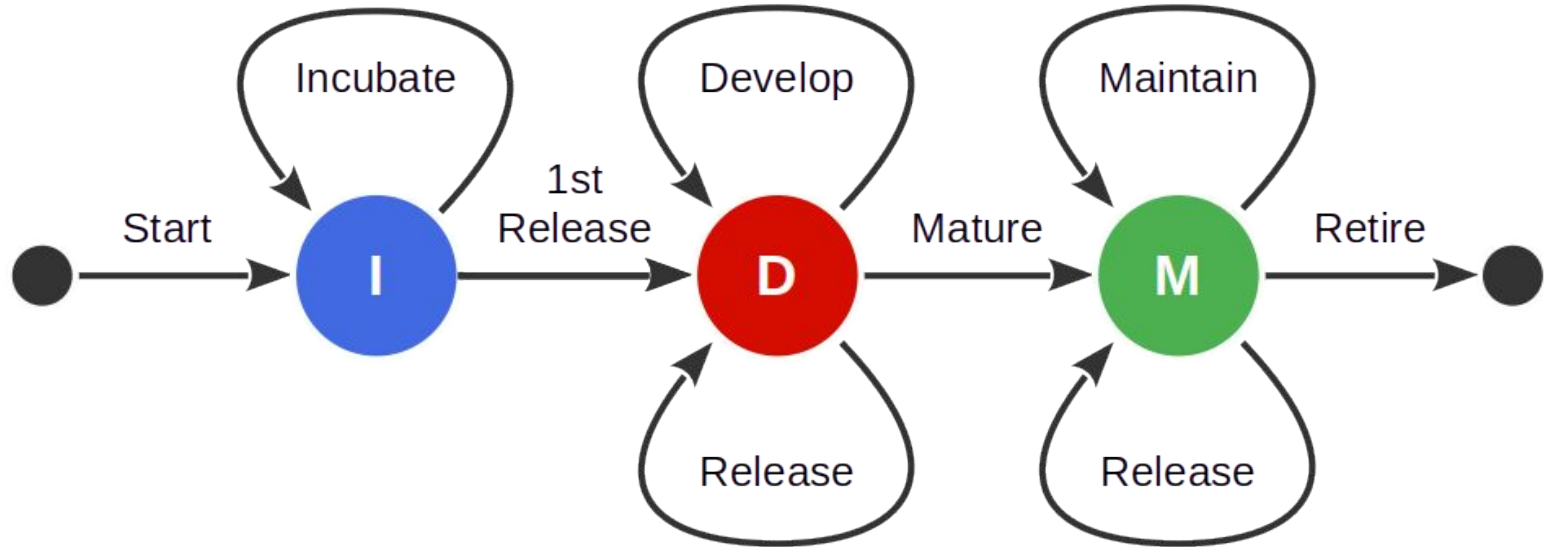
Projects have a defined start and end date

- Projects are developed for one client (one customer)

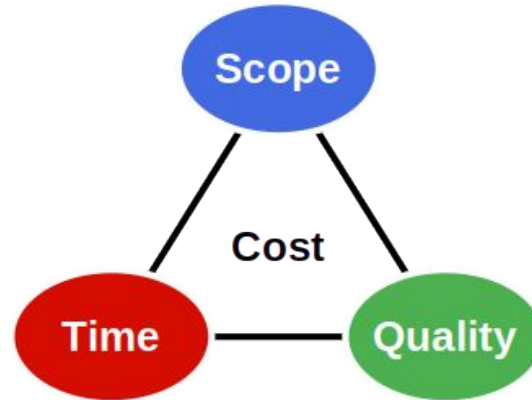
Project terms, roles, and positions often differs from product terms

- Product manager → business analyst, requirements engineering
- Engineering manager → project manager

Basic Software Product Life-Cycle



The Magic Triangle (“Pick Two”)



2. Plan-Driven Development

Basic Plan-Driven Development



Video From “The Pentagon Wars” [1]



Video Lessons

Stakeholders problems

- Multiple stakeholders with conflicting interests
- Meddling stakeholders intervening into the process

Requirement problems

- Inconsistent requirements (poor quality assurance)
- Changing requirements (wandering focus, long project)
- Feature creep (from troop carrier to tank)

Product problems

- Cost explosion due to lack of focus, rework
- Unclear market and wandering purpose

The Waterfall Model [1]

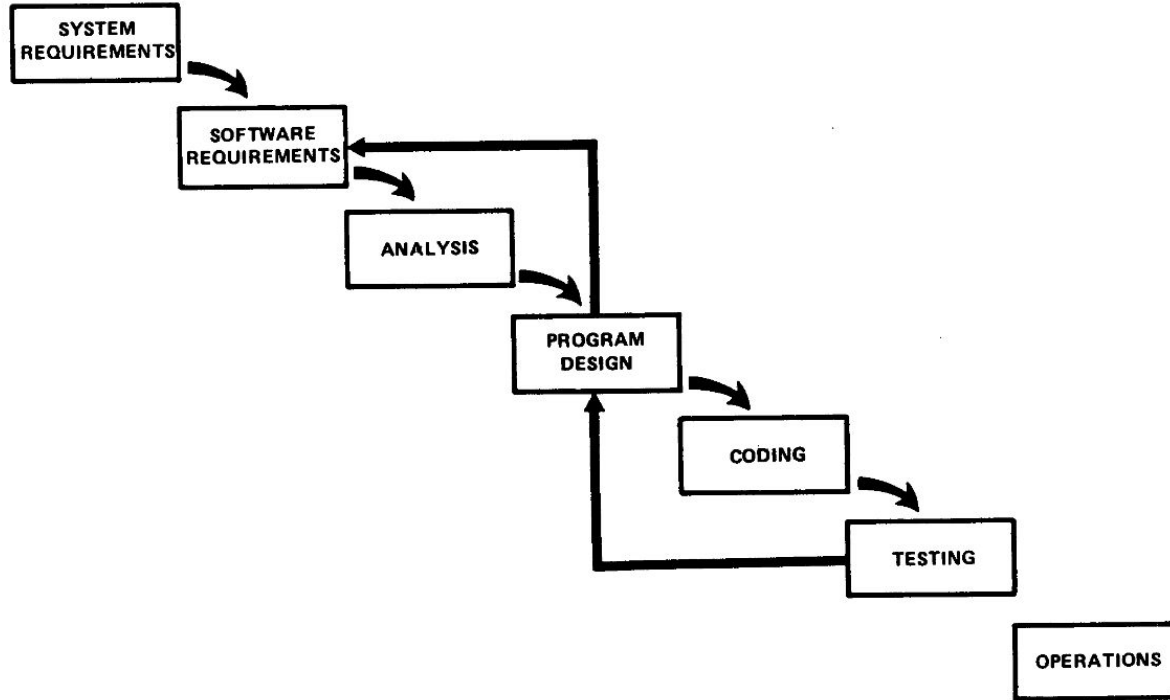


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

[1] Royce, W. W. (1970). Managing the development of large software systems. Proceedings of IEEE WESCON. Los Angeles, 328-388.

Main Lesson From Plan-Driven Development

Phases \neq Activities

3. Agile Methods

Agile Methods

Agile methods are a category of software development methodologies

- Defined in opposition to plan-driven development
- Driven by consultants as a significant business opportunity

The key idea of agile methods is to have a fast feedback loop

- Steer, don't plan and blindly execute
- Codified as the agile manifesto

Examples agile methodologies

- Scrum, XP, the Crystal Methods, Feature Driven Development

Principles of the Agile Manifesto [1]

Individuals and interactions

- Over processes and tools

Working software

- Over comprehensive documentation

Customer collaboration

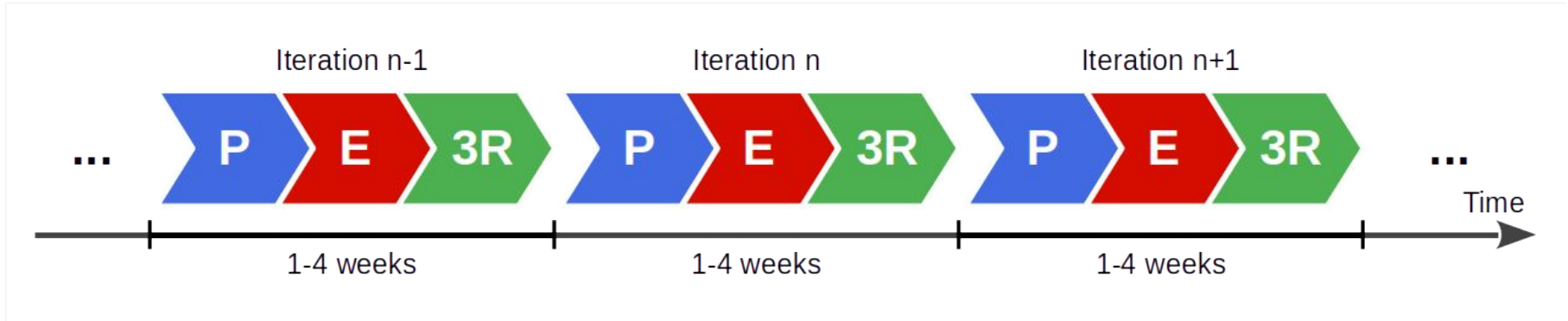
- Over contract negotiation

Responding to change

- Over following a plan

Agile Development Process

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



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

Benefits of Fast Feedback Loops

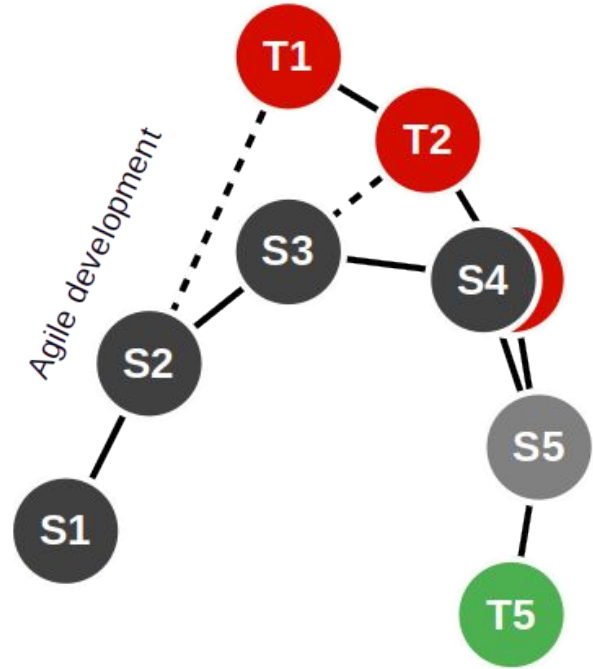
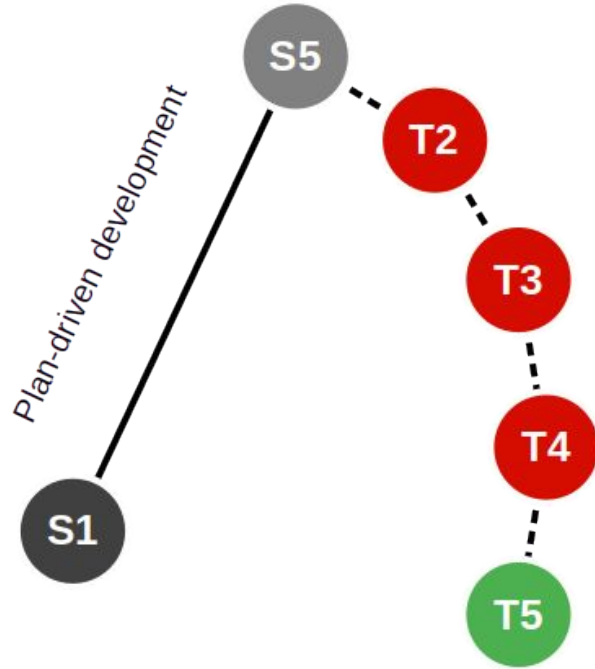
Short iterations

- 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

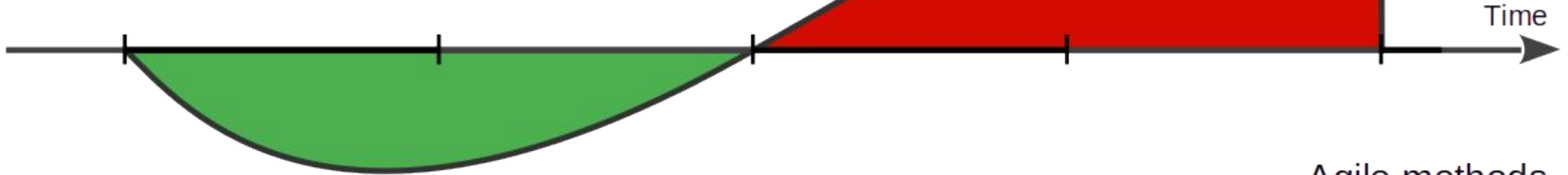
- User feedback helps team steer product to meeting needs right
- Feedback loop ensures that problems surface early
- Feedback helps recognize and realize new innovative features

Plan-Driven vs. Agile Processes

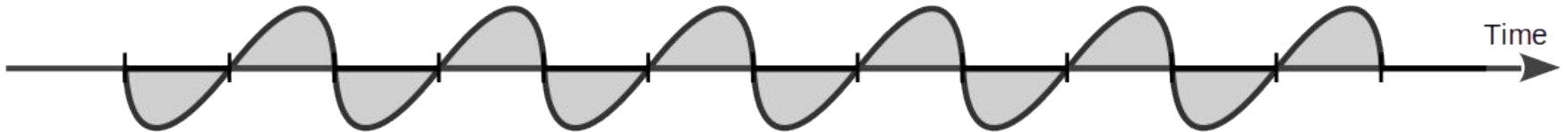


Plan-Driven vs. Agile Work Rhythms

Plan-driven methods
Long iterations



Agile methods
Short iterations



Do Agile Methods Lead to Cowboy Coding?

Agile methods are high discipline

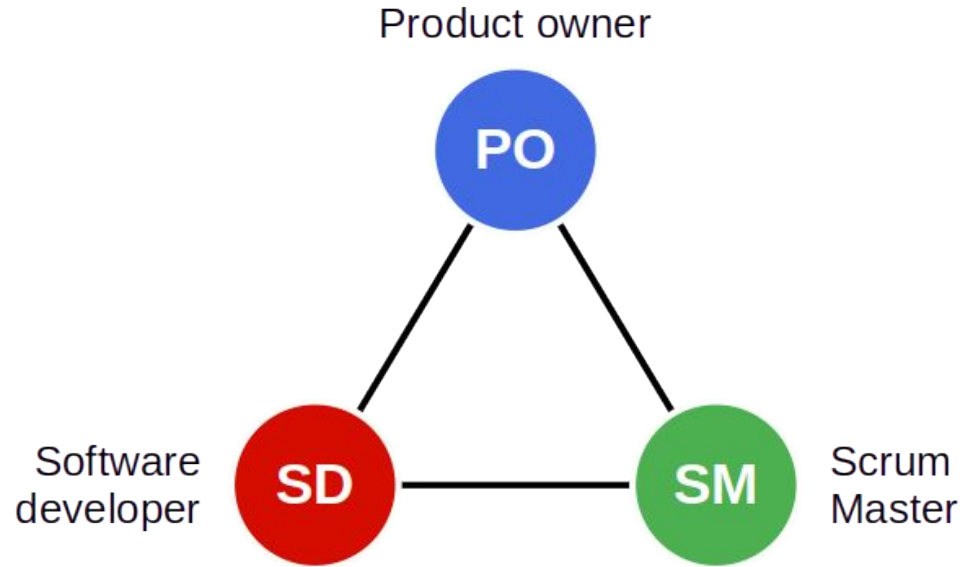
4. Scrum

Scrum [1]

Scrum is an agile method invented around 1993, 1995

- Has a minimal (agile) process model
- Is applicable to any domain, not just software development

Scrum Roles



Committed vs. Involved Roles

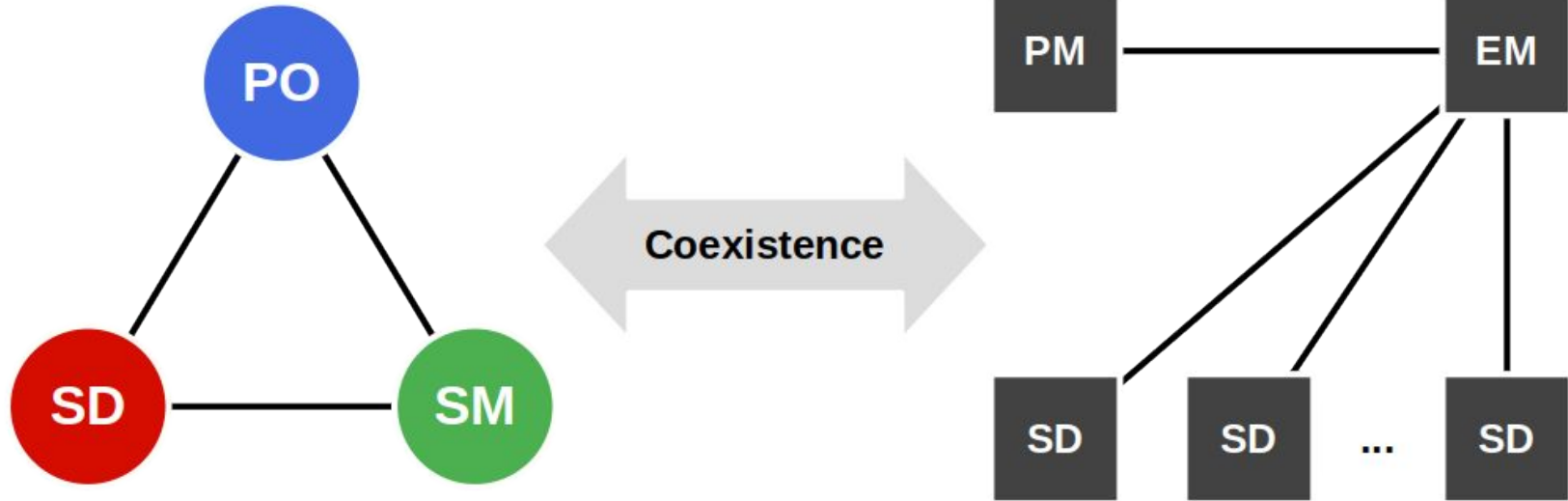
Committed parties

- Product owner
- Software developer(s)
- Scrum Master

Involved parties

- Customer
- Sponsor / funder
- ...

Mapping Roles to Posts



Roles / Posts Correspondence

Product management
Engineering management



Product owner

Software development
Engineering management
Quality assurance



Software developer

Engineering management



Scrum Master

Terms (The Scrum Terminology Mess)

Scrum	Product development	Project implementation
Product owner	Product manager	Business analyst
Product goal	Product vision [1]	Project mission [1]
Product backlog	Product requirements documentation	Requirements specification

[1] This resolution is specific to AMOS, though the terms are generally known and used

Scrum Scope

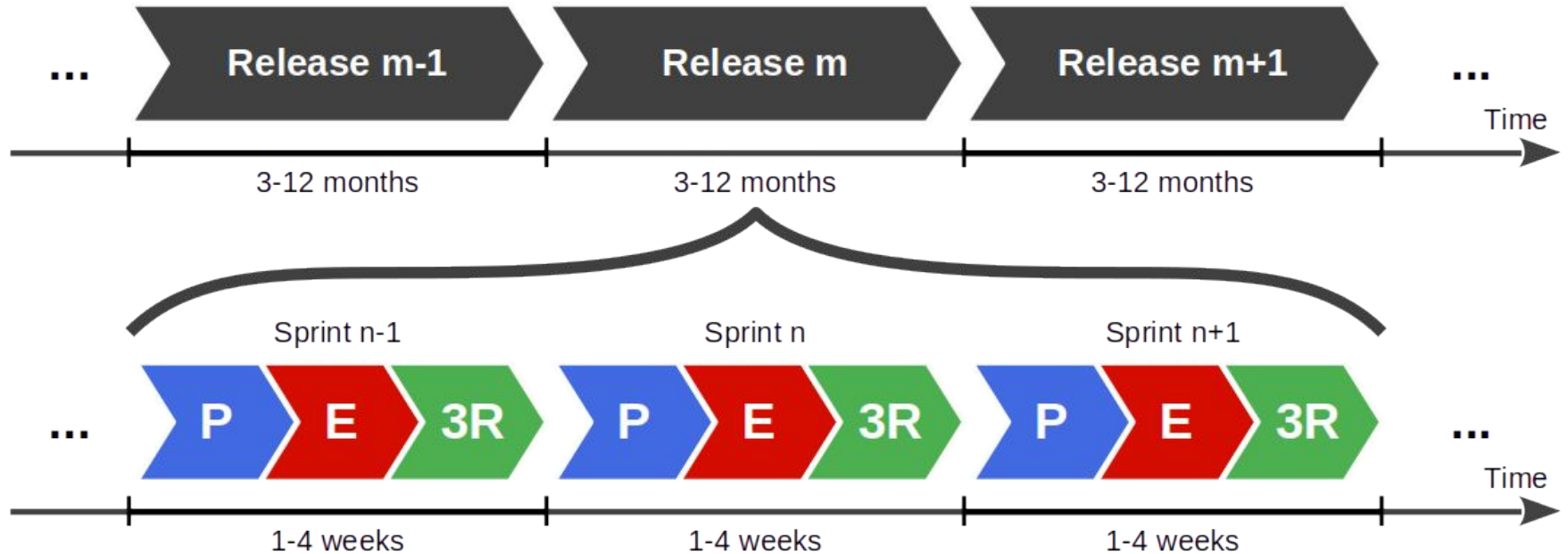
Scrum proper covers

1. Day
2. Sprint (weeks)
3. Project/product releases (months)

Further evolutions e.g. SAFe cover

4. Product life-cycle (years)
5. Portfolio

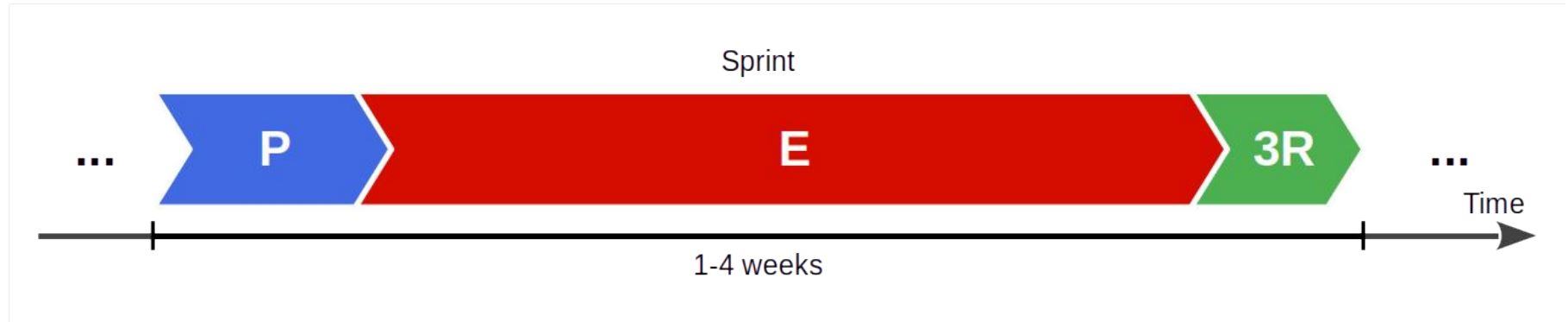
Scrum Process



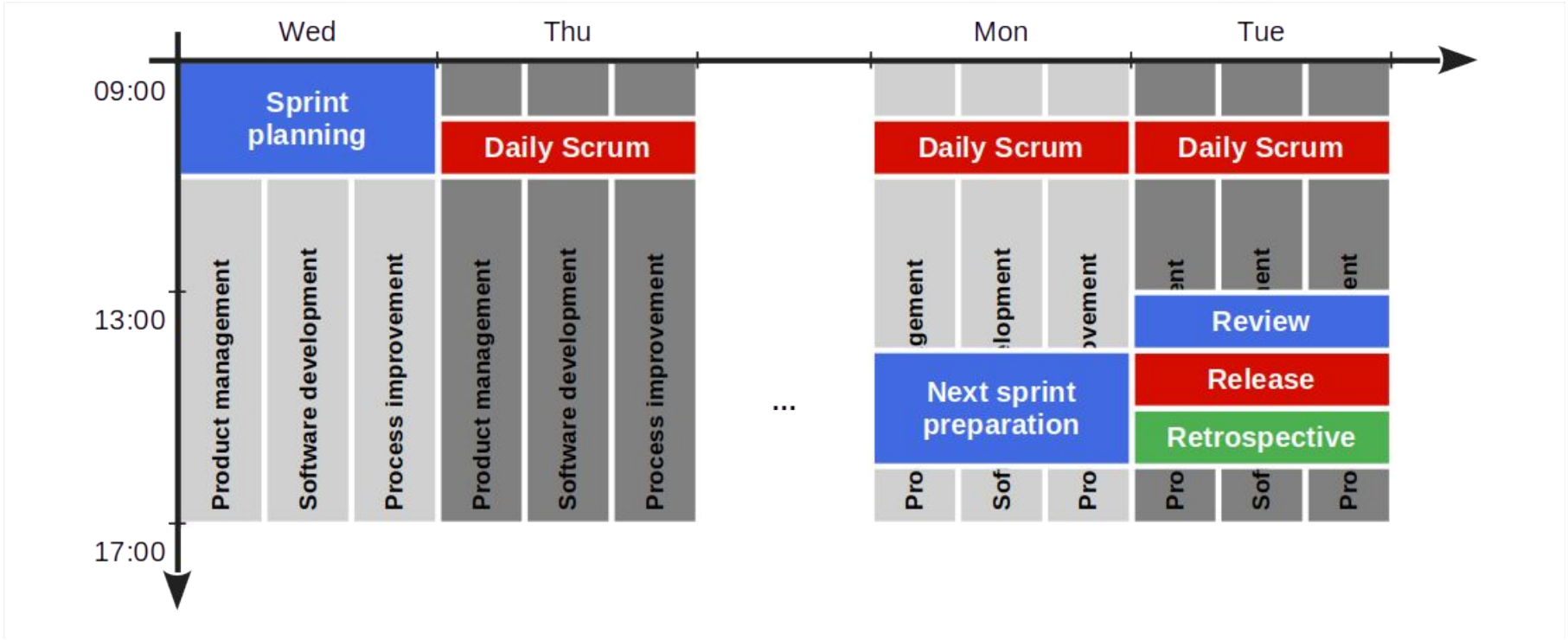
Scrum Sprint

A sprint is Scrum's iteration; it is an equal-length time-box

It is a highly structured process with defined feedback points



Sprint Structure



Sprint Meetings

1. Next sprint preparation

- a. Product owner and senior developer groom the product backlog

2. Sprint review

- a. Team reviews this sprint's results, signs off on them

3. Sprint release

- a. Team decides on sprint release

4. Sprint retrospective

- a. Team reviews process, commits to improvements

5. Sprint planning

- a. Team discusses upcoming work, commits to it

6. Daily Scrum

- a. Team members update each other on work progress

The AMOS
team meeting

Sprint Workstreams

Product management; the product owner

- Grooms the product backlog
- Answers questions to developers

Software development; software developers

- Break down backlog items into tasks, self-organize
- Design and implement sprint backlog items

Process improvement (Scrum Master)

- Observes and facilitates team dynamics
- Removes or reduces process impediments

Summary

1. Software development
2. Plan-driven development
3. Agile methods
4. Scrum

Thank you! Any 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](https://creativecommons.org/licenses/by/4.0/) license

Copyright

- © Copyright 2023 Dirk Riehle, some rights reserved