Agile Software Development in Corporate Environments

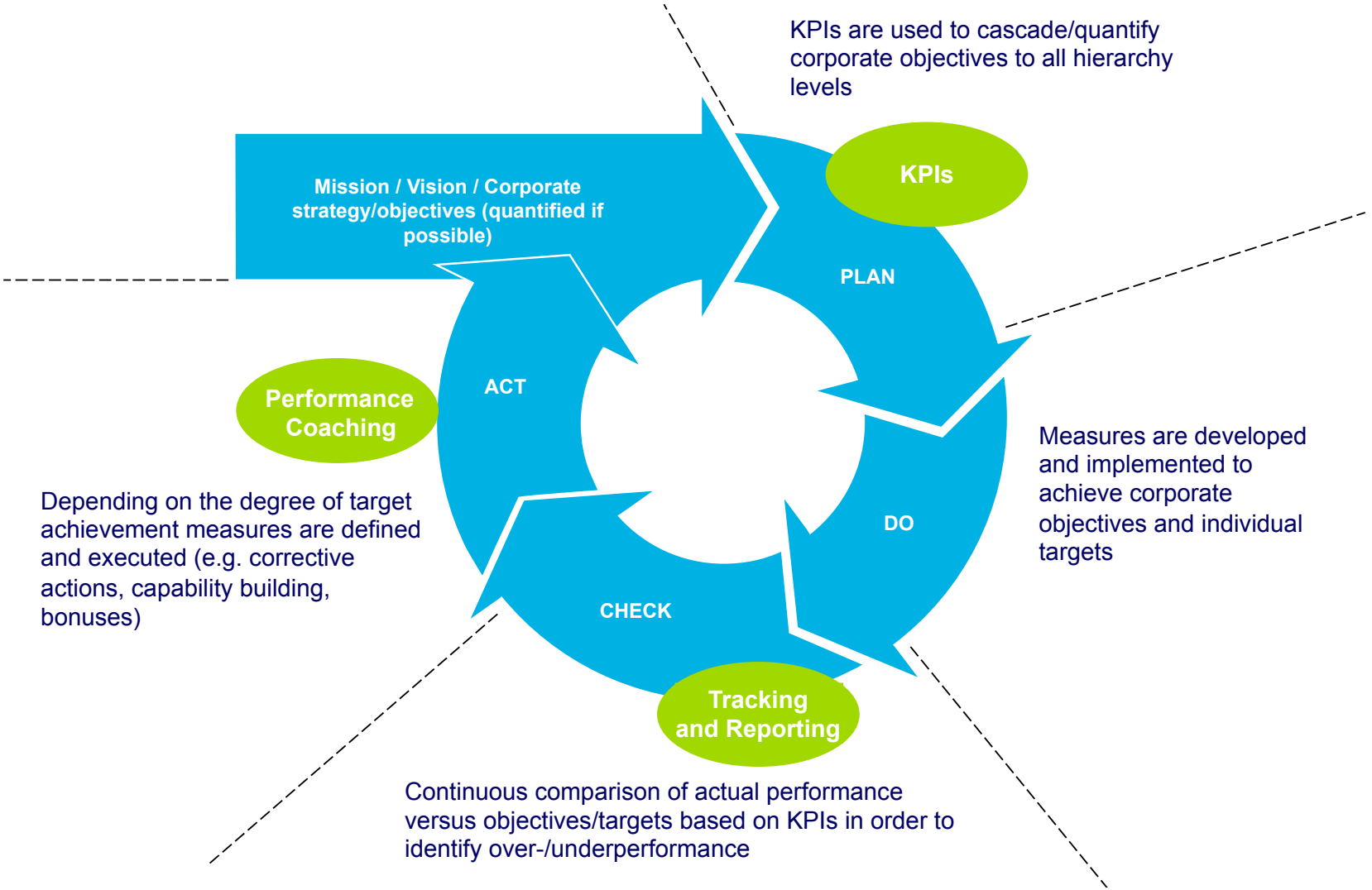
Dr. Alexander Schatten
biac (Vienna Insurance Group)
alexander.schatten@biac.at
alexander@schatten.info
http://www.schatten.info

Overview

- 20 years (70 years?) agile practices
- Motivation
- The *Agile Manifesto*
- eXtreme Programming, SCRUM, (Software) Kanban
- Corporate challenges
- Requirements and transparency

20 YEARS AGILE PRACTICES

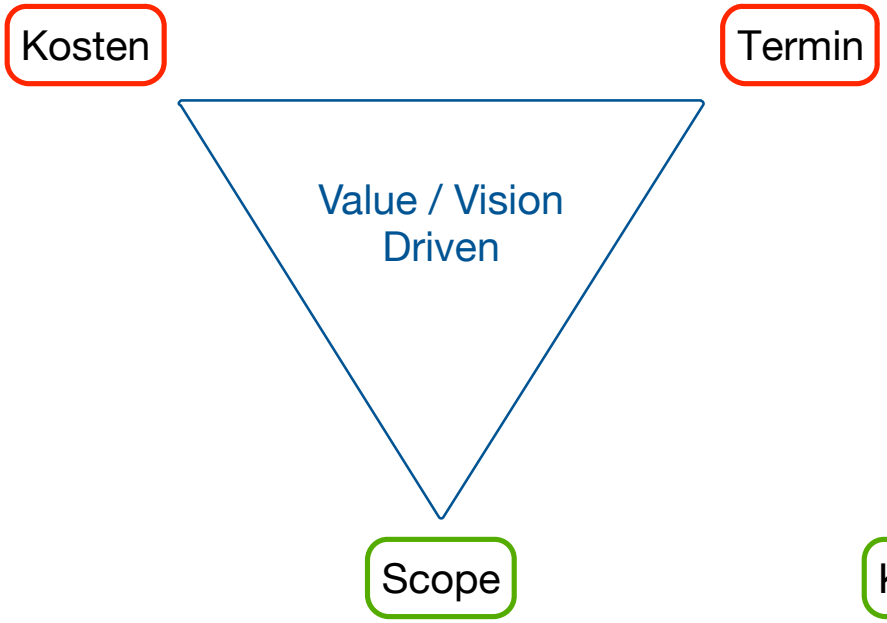
Plan / Do / Study / Act (1940)



Motivation

- Organisational challenges
- Technical challenges
- The illusion of predictability and forecast reliability

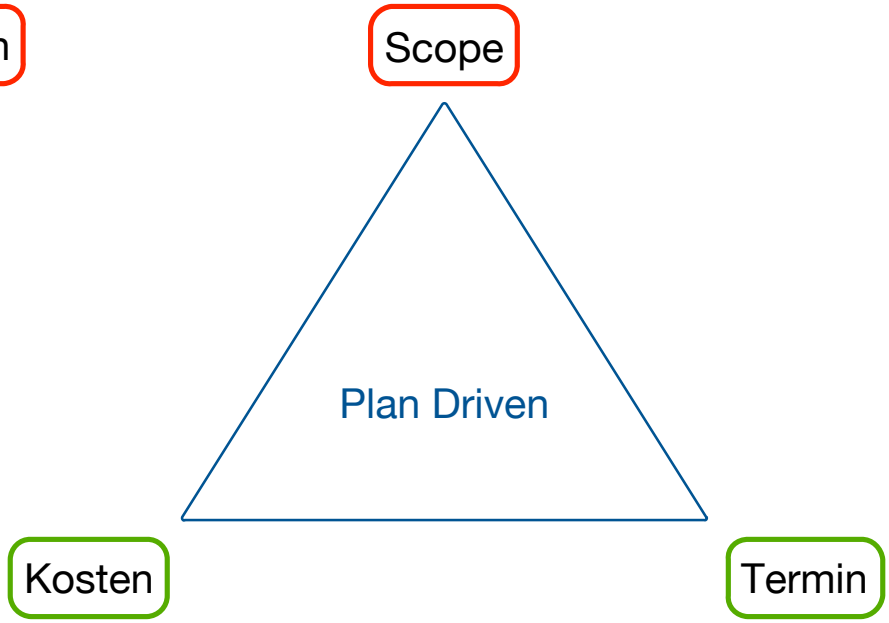
Fixpunkte



Schätzung / Variable

Agile Methodik

Fixpunkt

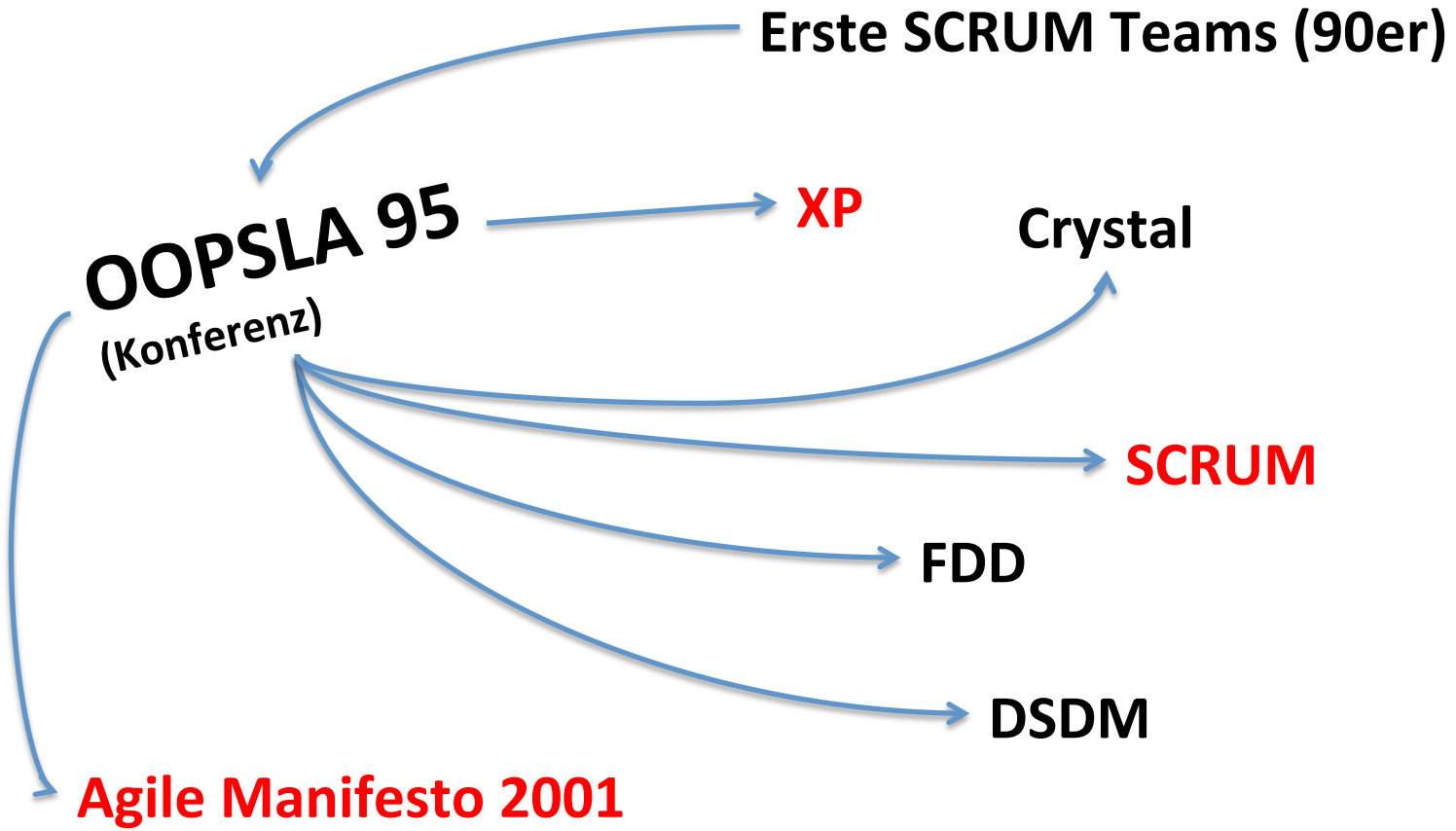


Schätzungen / Variable

“Wasserfall”

Process	Waterfall Development	Iterative and Incremental	Agile Development
Measure of Success	Conformance to plan	→	Response to change, working code
Management Culture	Command and control	→	Leadership/ collaborative
Requirements and Design	Big and up front	→	Continuous/emergent/ just-in-time
Coding and Implementation	Code all features in parallel/test later	→	Code and unit test, deliver serially
Test and Quality Assurance	Big, planned/ test late	→	Continuous/concurrent/ test early
Planning and Scheduling	PERT/detailed/fix scope, estimate time and resource	→	Two-level plan/fix date, estimate scope

The First 20 Years...



Software Kanban (~2007)

Agile Manifesto

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

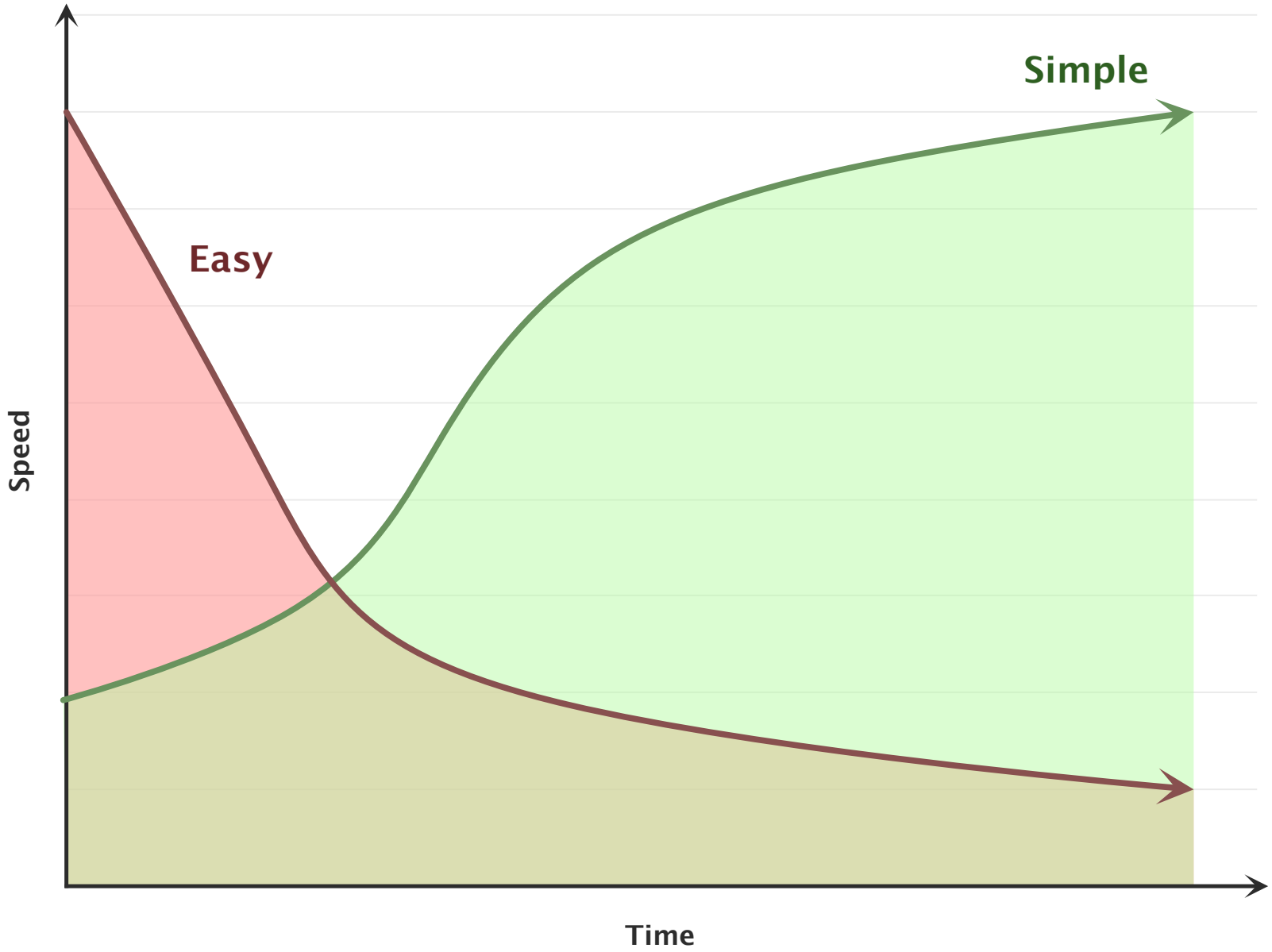
Agile Practices Today

- Prozess-oriented
  - SCRUM
  - Software Kanban
- Methodical building blocks
  - eXtreme Programming
- Geeks only?
  - Meanwhile ~15 years of experience of agile methods in corporate environments
  - Usage in small and large international companies, but also in Austria (e.g. banks)
  - Usage in small projects (individual teams, 3–10 devs)
  - Usage in large projects (multiple teams, hundreds of devs)

EXTREME PROGRAMMING

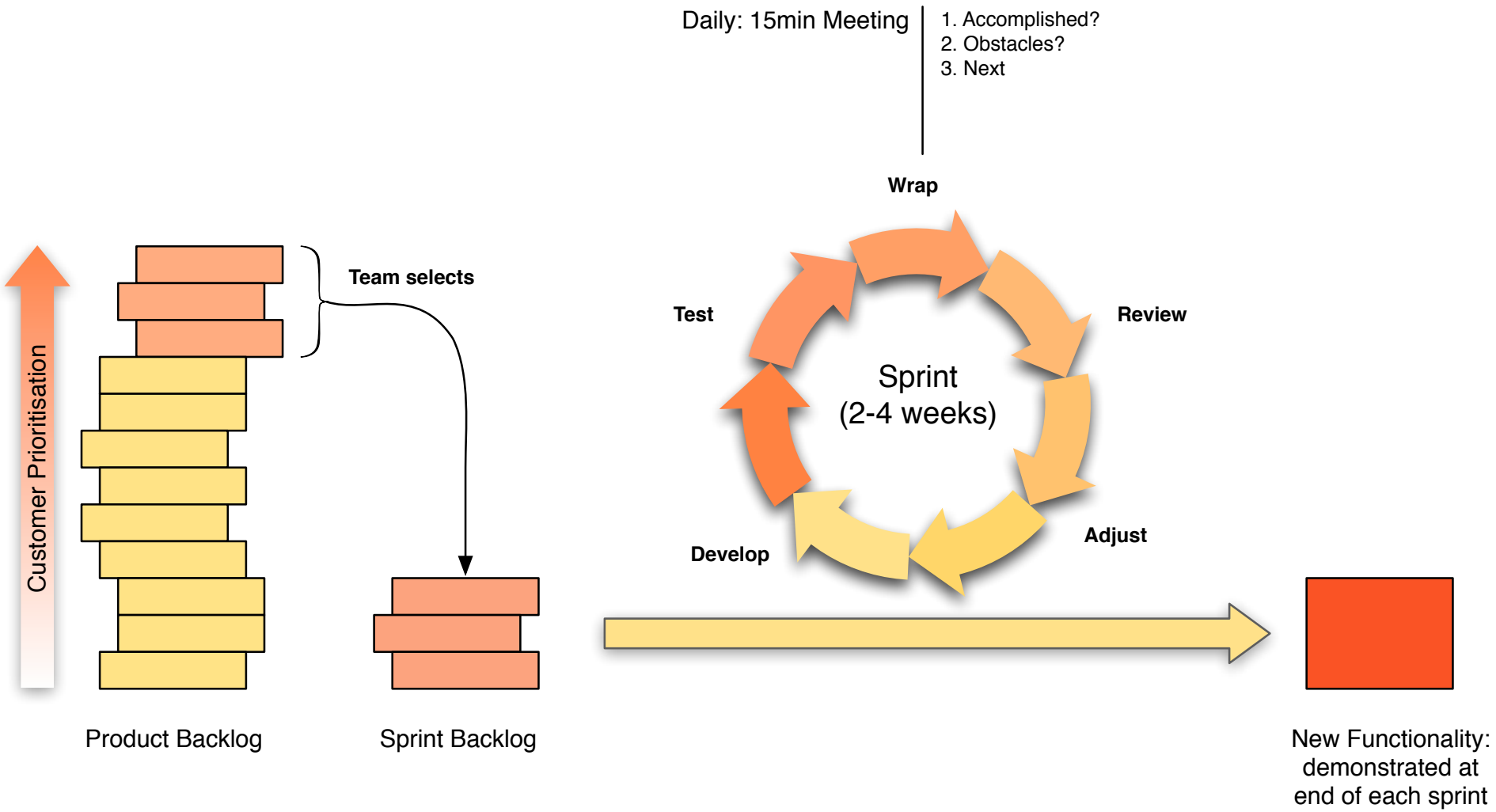
XP-Praktiken

- **Communication / Collaboration / Architecture**
  - Planning Game (WS)
  - Metaphor (WS)
  - Simple Design (WS)
- **Process**
  - Small Releases
  - Pair Programming
  - Collective Code Ownership (WS)
  - 40-hrs Week
  - On-Site Customer (WS)
- **Technical**
  - Coding Standards
  - Testing
  - Continuous Integration
  - Refactoring


Simple
Easy
Speed
Time

SCRUM

SCRUM Prozess

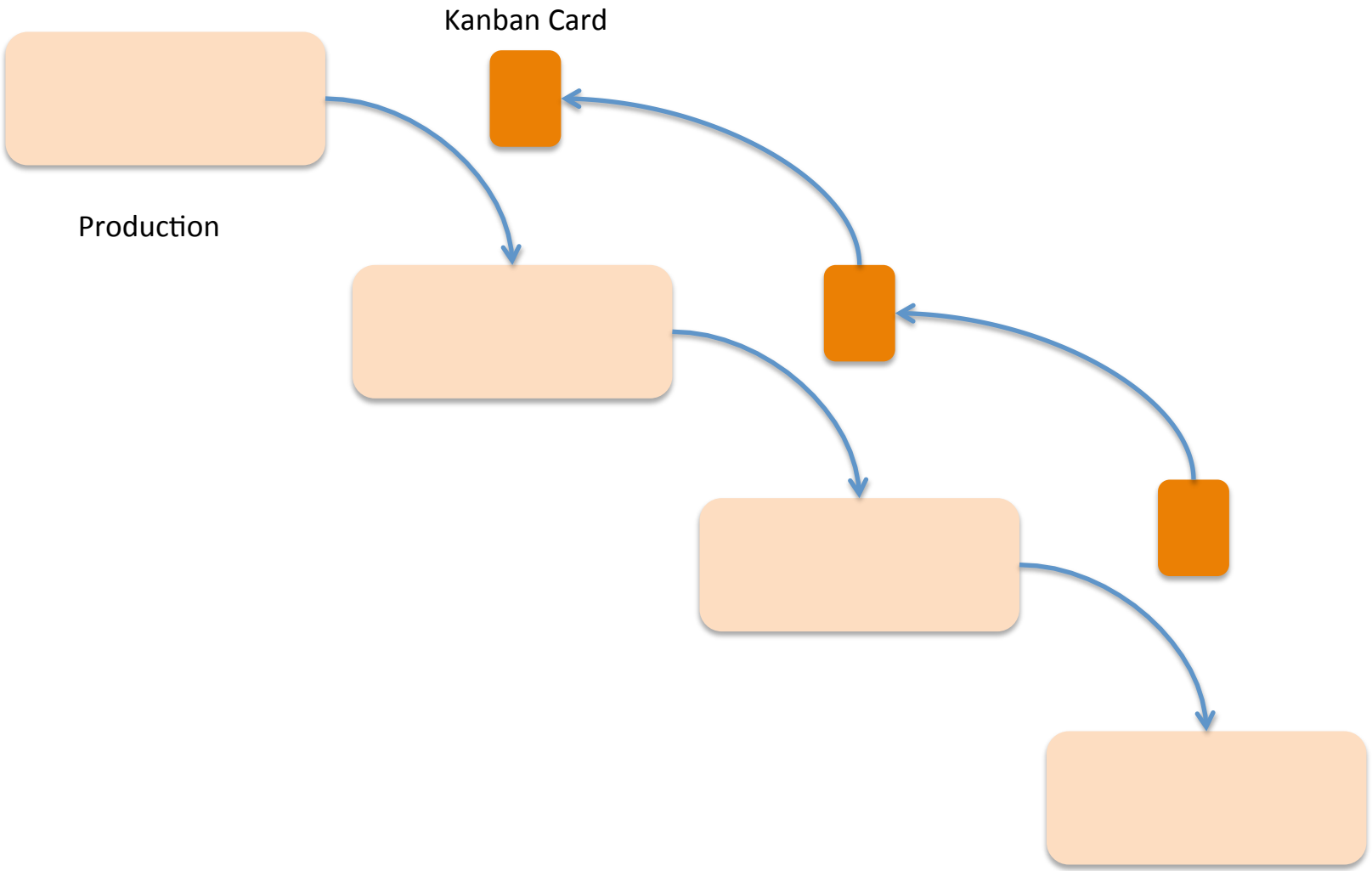


SCRUM Prozess

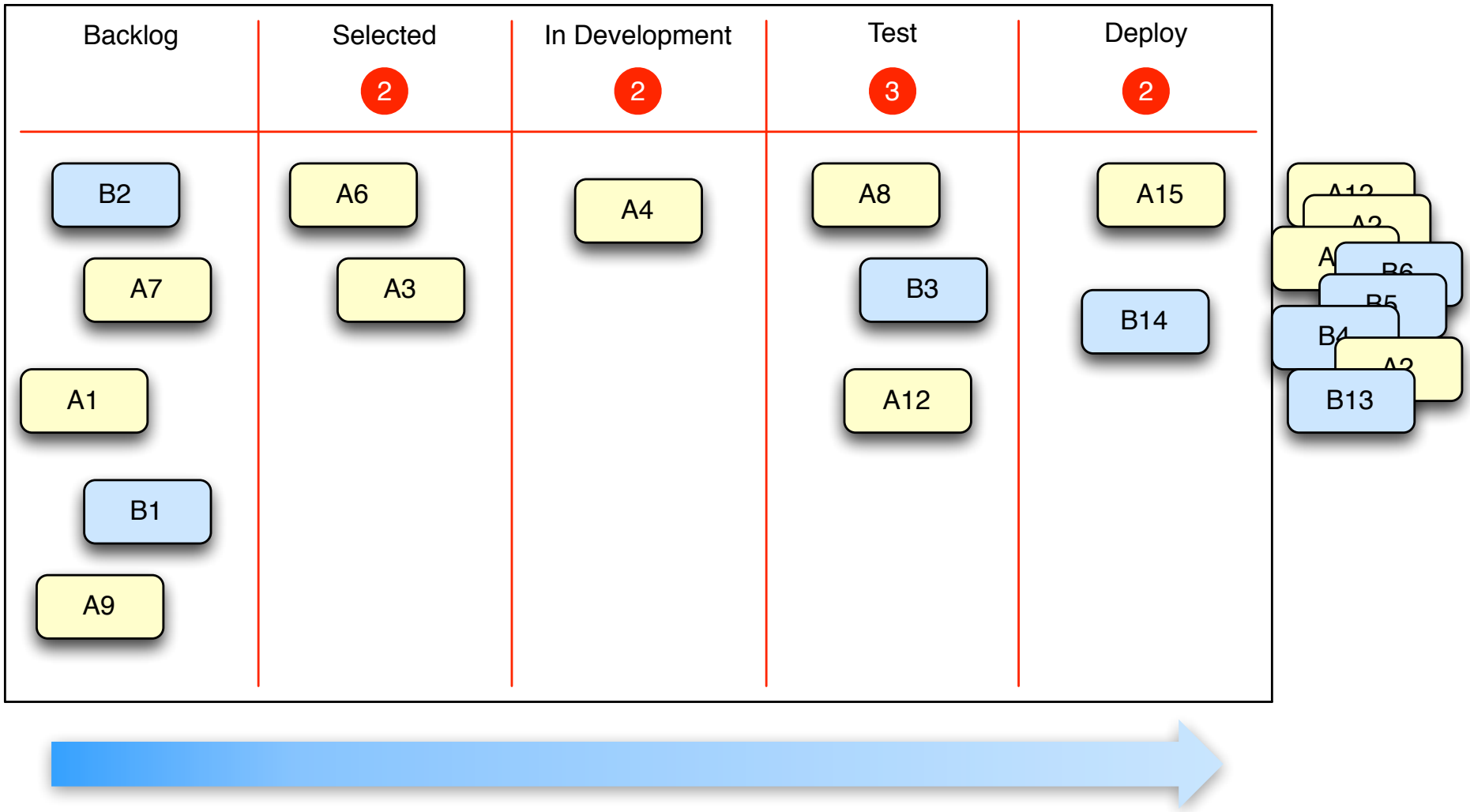
- Work in **Iterations**
- **Team size** < 10 Personen
- Customer is tightly integrated in the process
- Realistic **estimations** hand in hand with **controlling**
  - User Stories
  - Backlogs
  - “Planning Poker”
- **Team-estimations** and **Performance** (checked at each iteration)
- Undisturbed and focussed work for teams during iterations

SOFTWARE KANBAN

“Stop Starting, Start Finishing…”



Workflow Visualisation



Software Kanban

- Focus on “flow”, avoiding bottlenecks
- Visualisation of current process flow and activities
- Real-time metrics (KPIs), e.g.
  - Average Lead Time
  - Cumulative Flow Diagrams: Cycle Time

AGILE PRACTICES IN CORPORATE ENVIRONMENTS

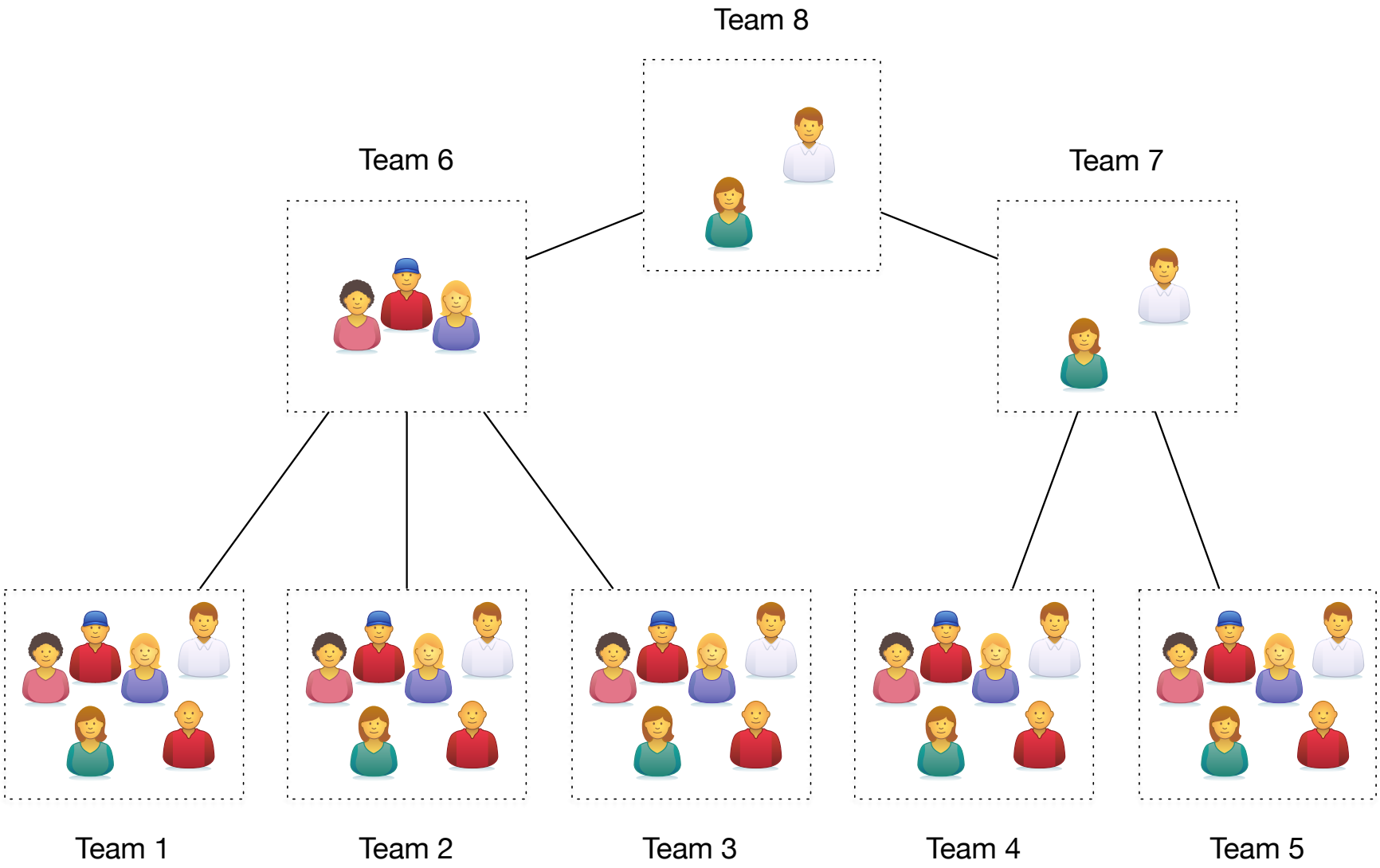
Agile Practices? Startups Only?

- Salesforce
- Google
- Yahoo
- Intel
- Siemens
- McKinsey
- Philips
- JP Morgan
- Bank Austria, BAWAG PSK, Raiffeisen
- Fabasoft, Frequentis, Tricentis, Anecon, Accenture, Kapsch
- Usw.

Challenges in Corporate Environments

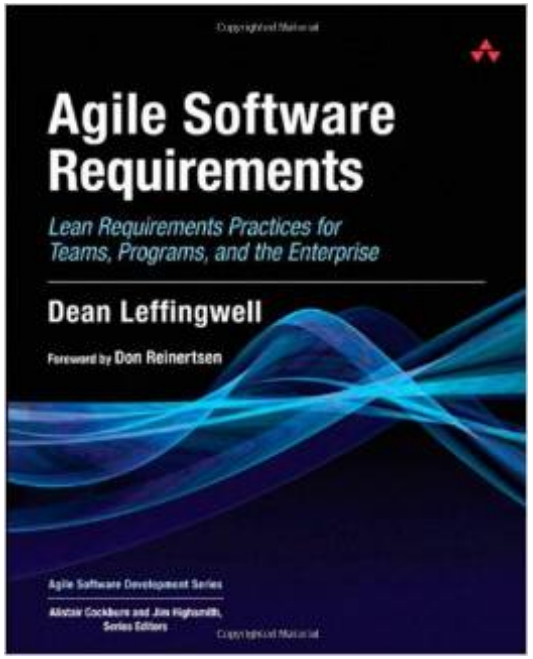
Multiple Teams, many developers, large projects

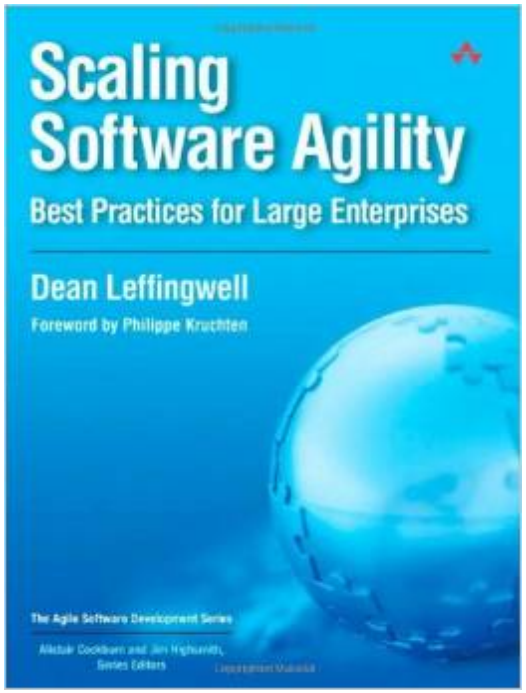
Scrum of Scrums

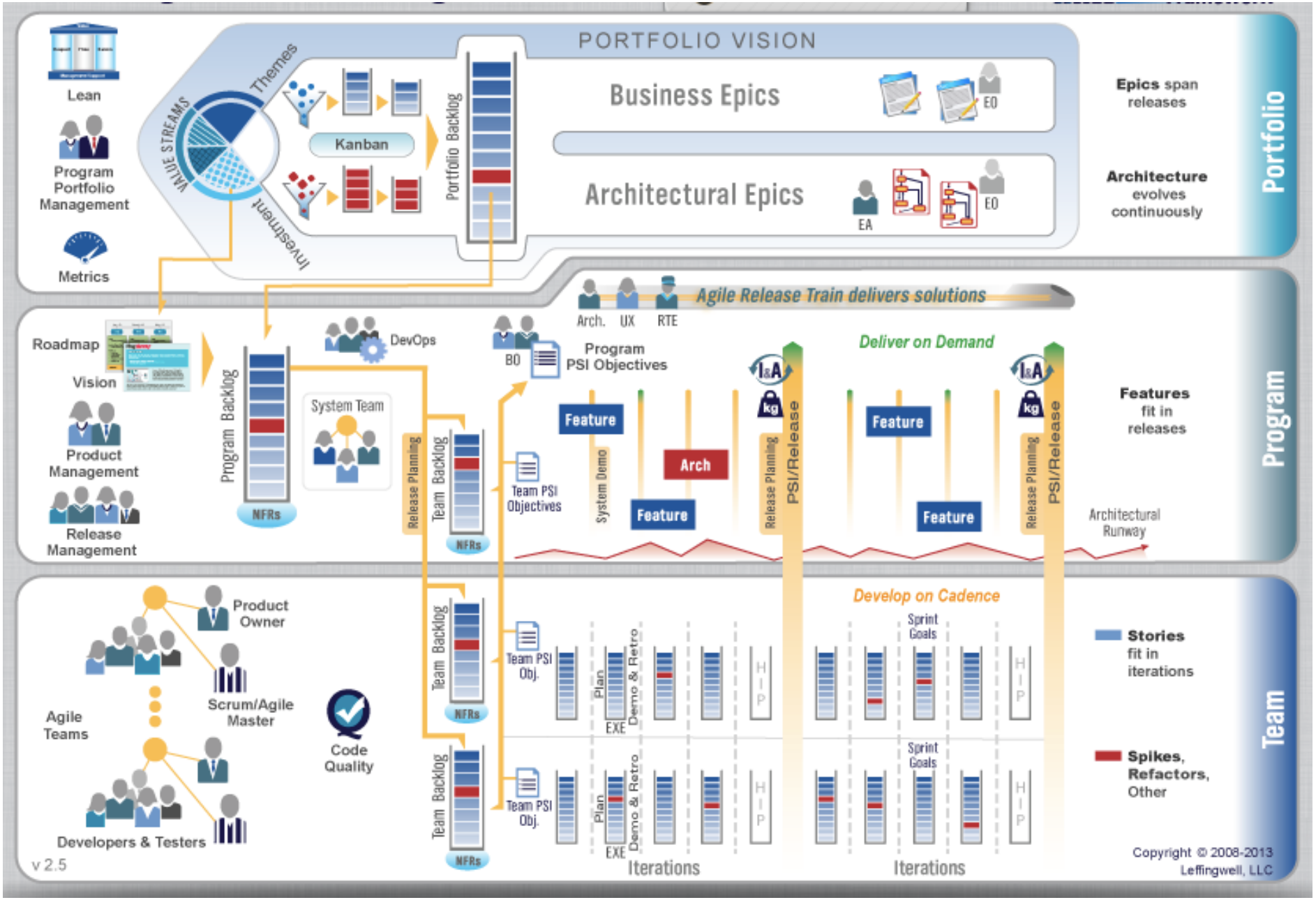


SAFE Framework

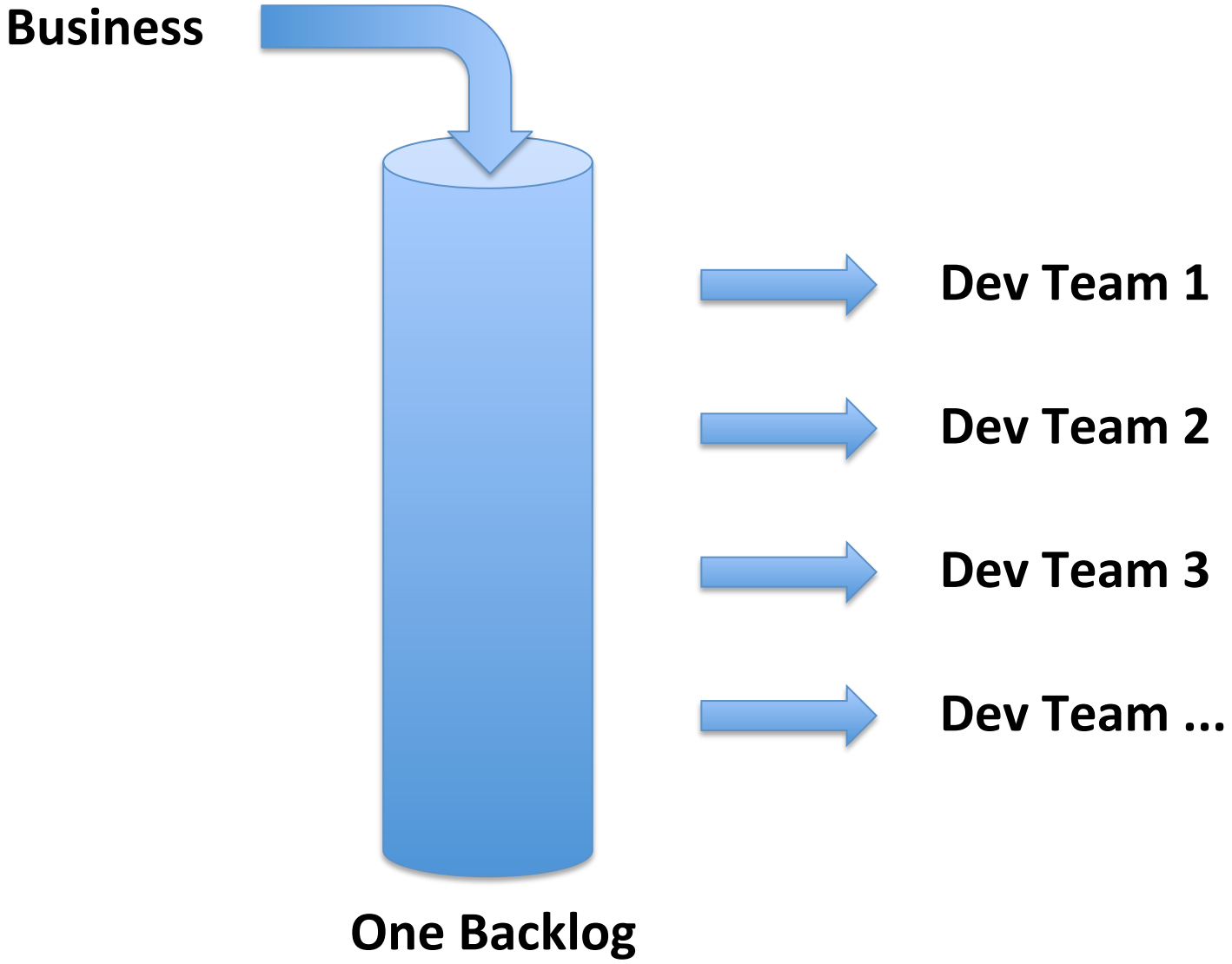
Dean Leffingwell: Scaled Agile Framework
http://scaledagileframework.com



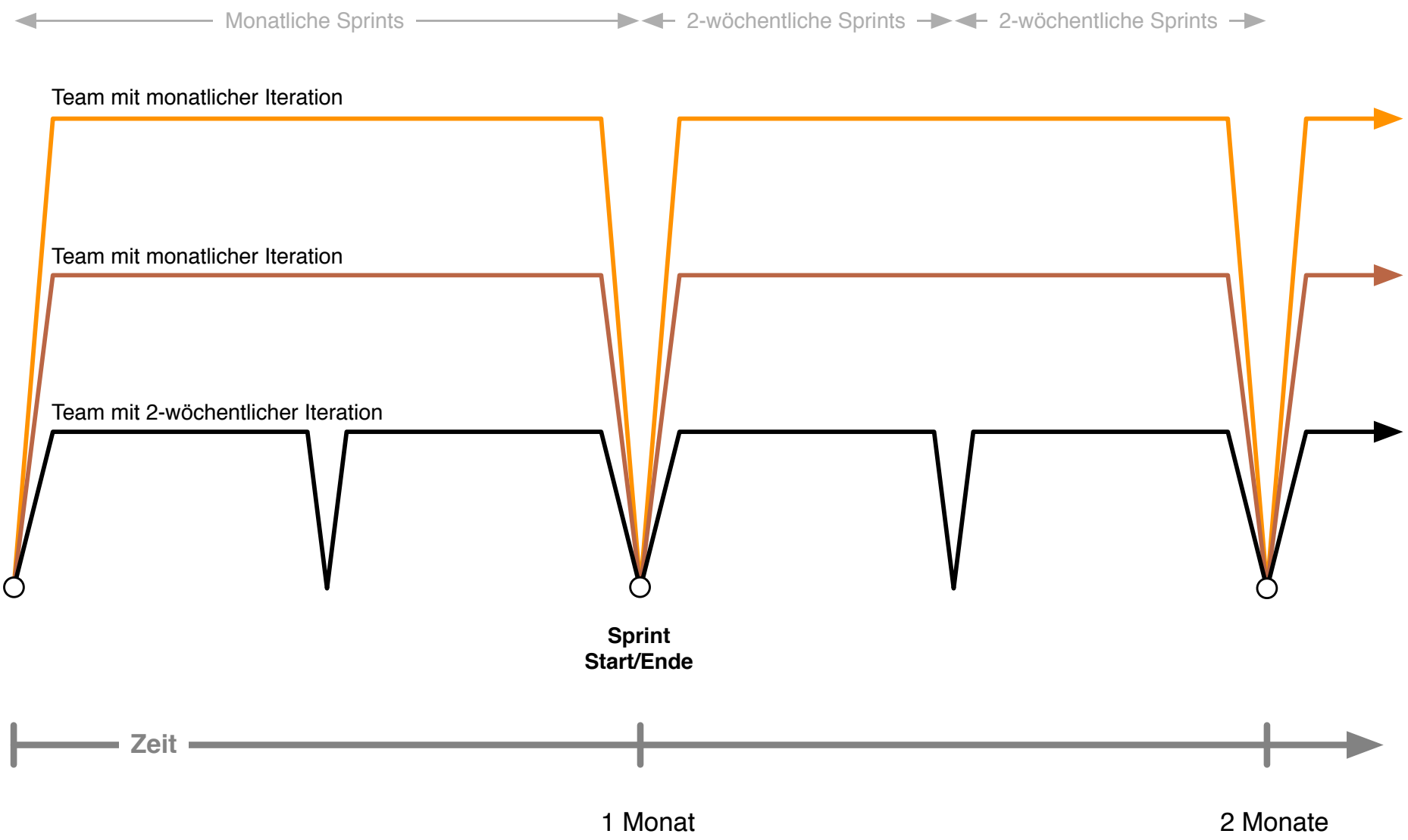



PORTFOLIO VISION
Lean
Program Portfolio Management
Metrics
VALUE STREAMS
Themes
Investment
Kanban
Portfolio Backlog
Business Epics
EO
Architectural Epics
EA
EO
Epics span releases
Architecture evolves continuously
Portfolio
Agile Release Train delivers solutions
Arch.
UX
RTE
Roadmap
Vision
Product Management
Release Management
Program Backlog
NFRs
DevOps
System Team
BO
Program PSI Objectives
Deliver on Demand
Feature
System Demo
Arch
Feature
Feature
Feature
I&A
kg
Release Planning
PSI/Release
Architectural Runway
Features fit in releases
Program
Team Backlog
Team PSI Objectives
Product Owner
Scrum/Agile Master
Agile Teams
Developers & Testers
Code Quality
Team PSI Obj.
Plan
EXE
Demo & Retro
H I P
Develop on Cadence
Sprint Goals
Iterations
Stories fit in iterations
Spikes, Refactors, Other
Team
v 2.5
Copyright © 2008-2013 Leffingwell, LLC

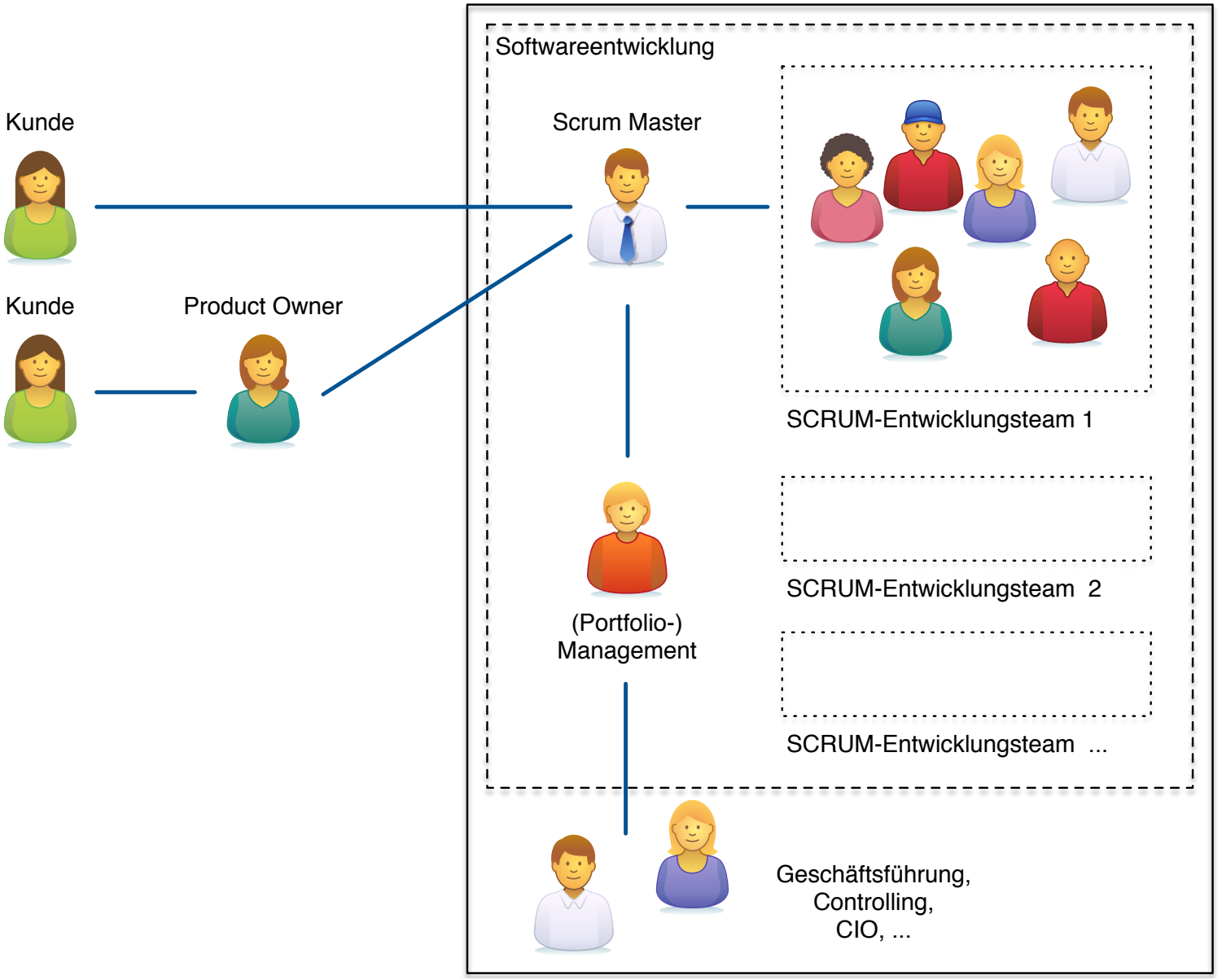
“Factory” Approach



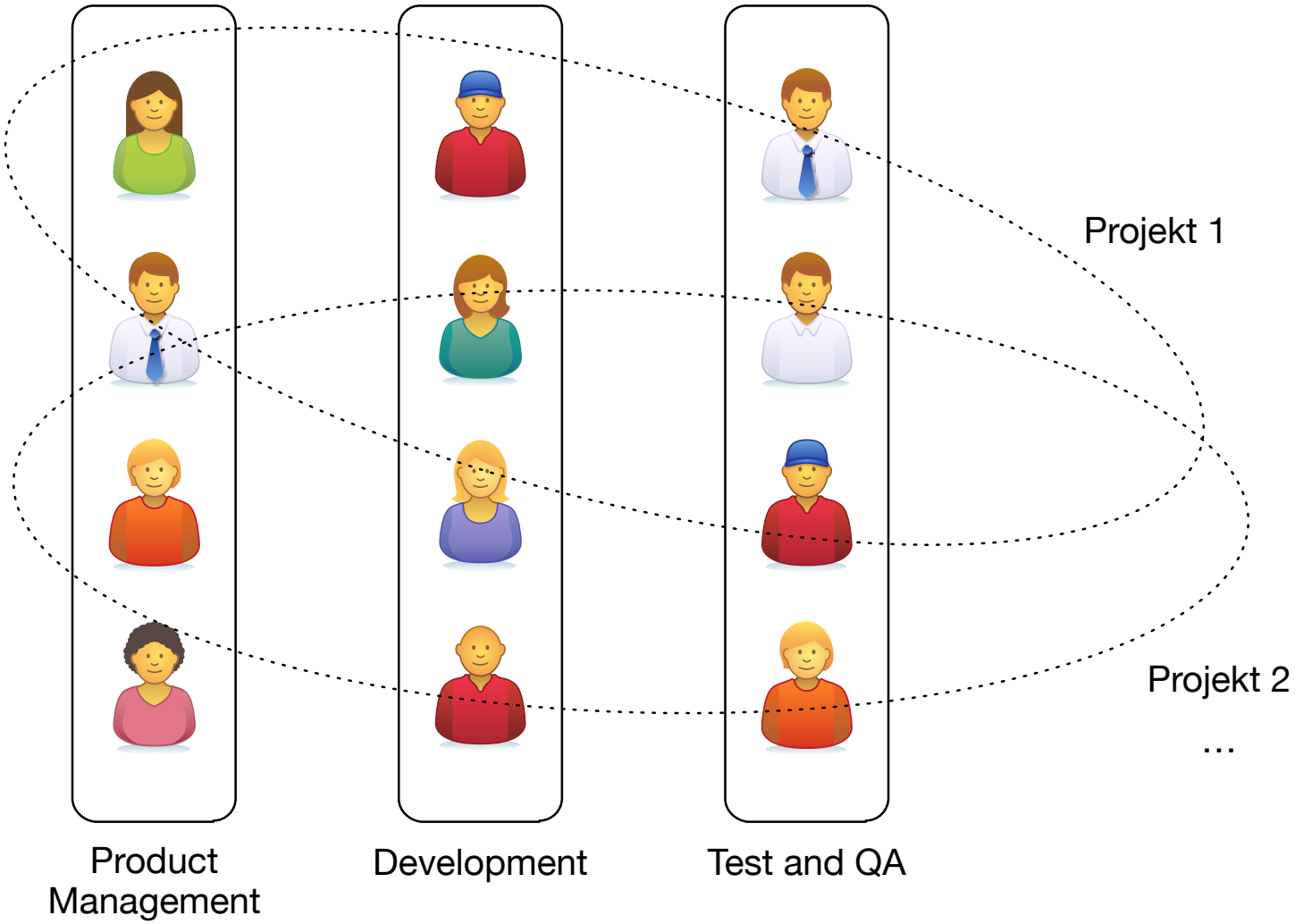
Portfolio Approach



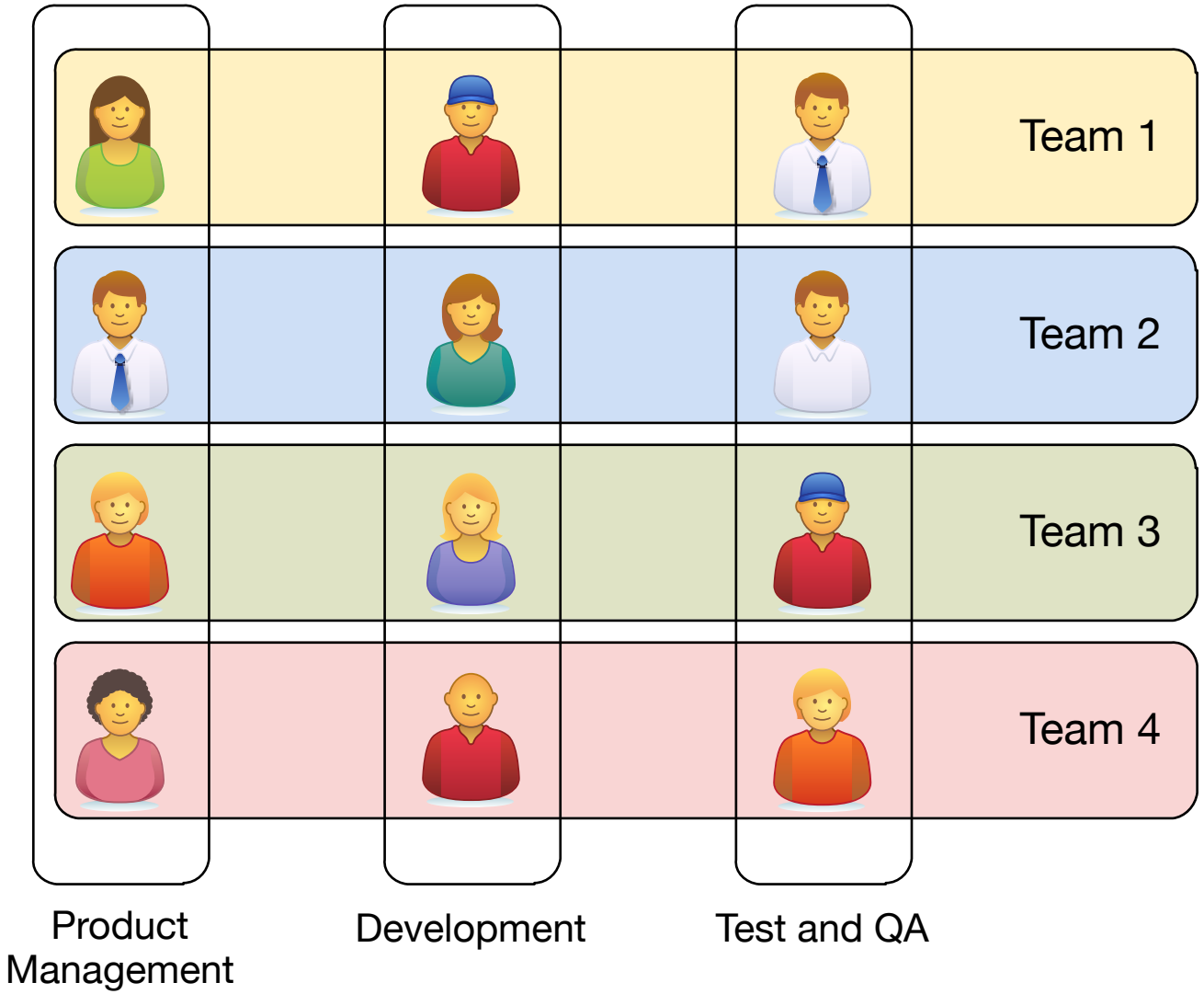
Roles in Agile Processes



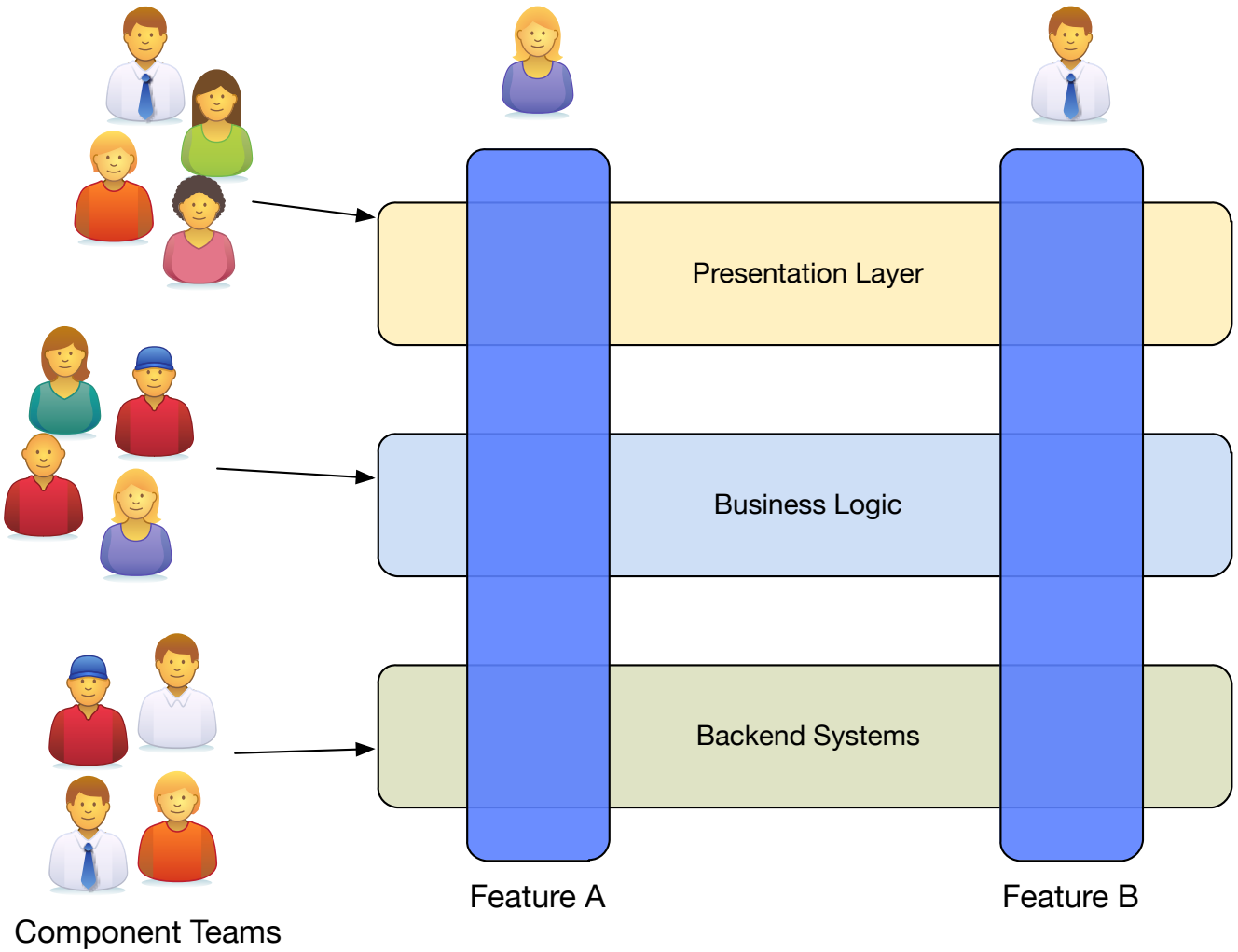
Functional Silos (traditional)



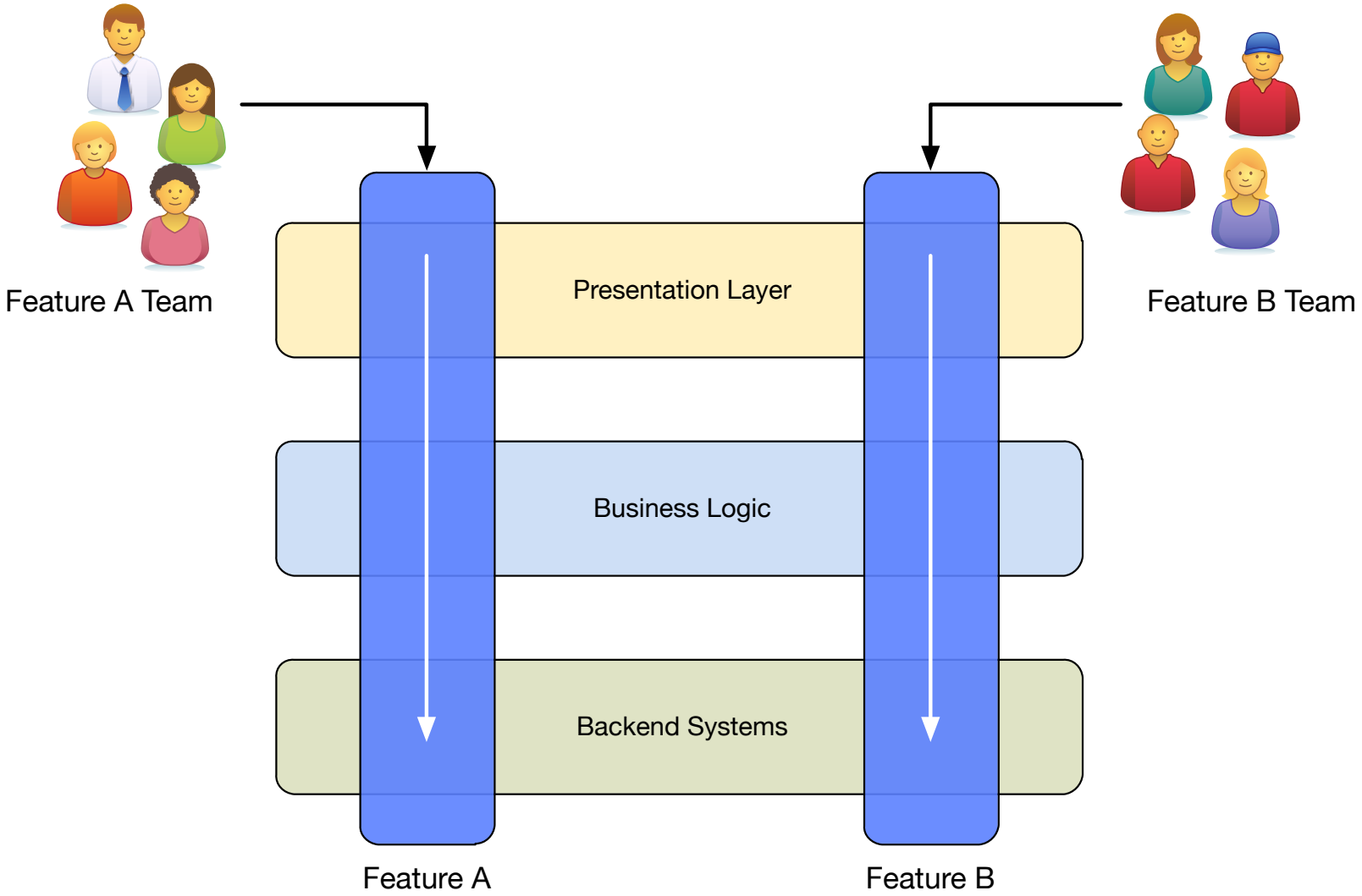
Agile Teams


Team 1
Team 2
Team 3
Team 4
Product Management
Development
Test and QA

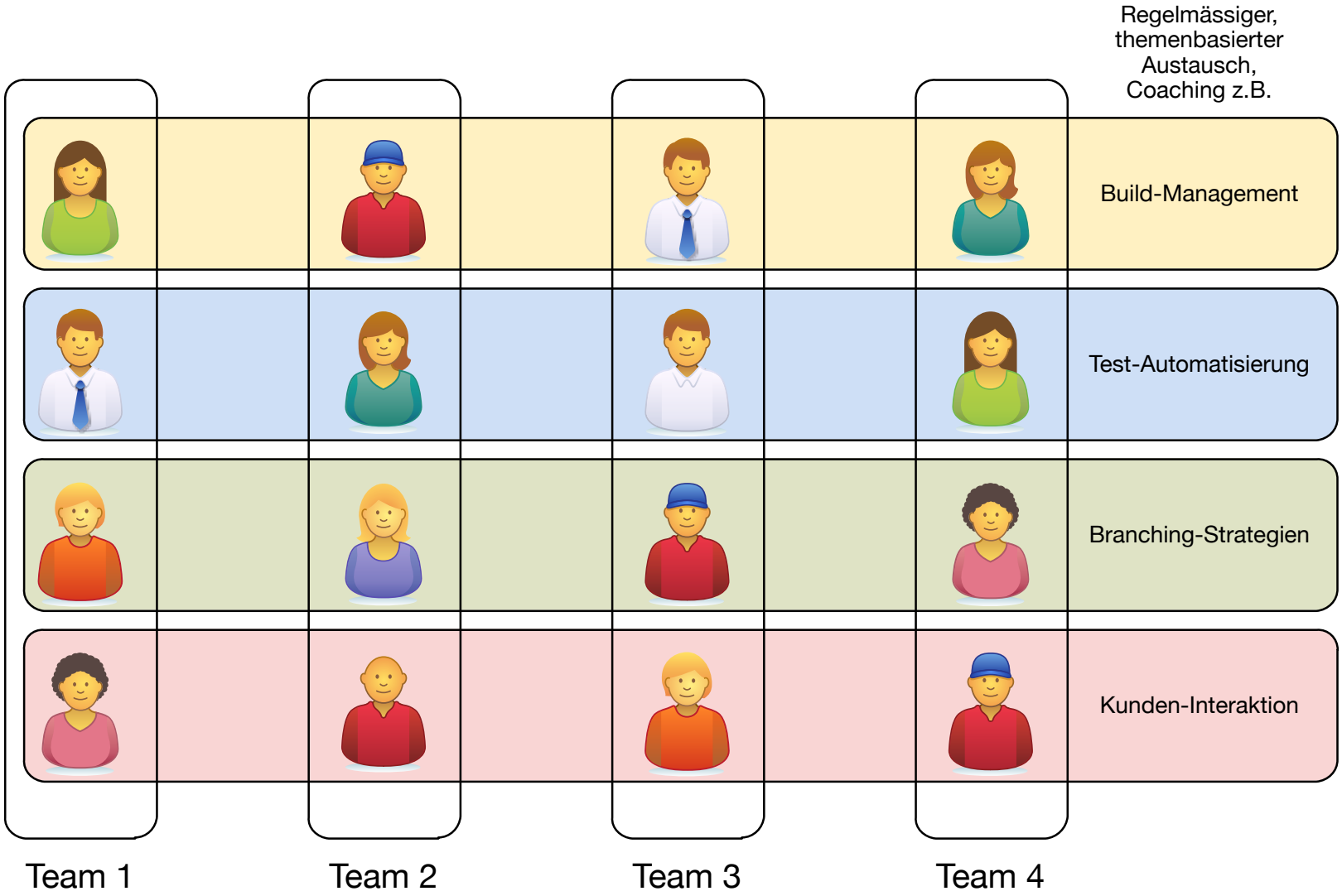
Organisation Following Components (traditional)



Agile Team Organisation Following Features



Cross-Team Expert Meetings



Customer Responsibility

- Product Owner
- Roles and process responsibilities have to be clarified
- Lack of clear roles is often the main factor for failure in agile projects!

Further Roles

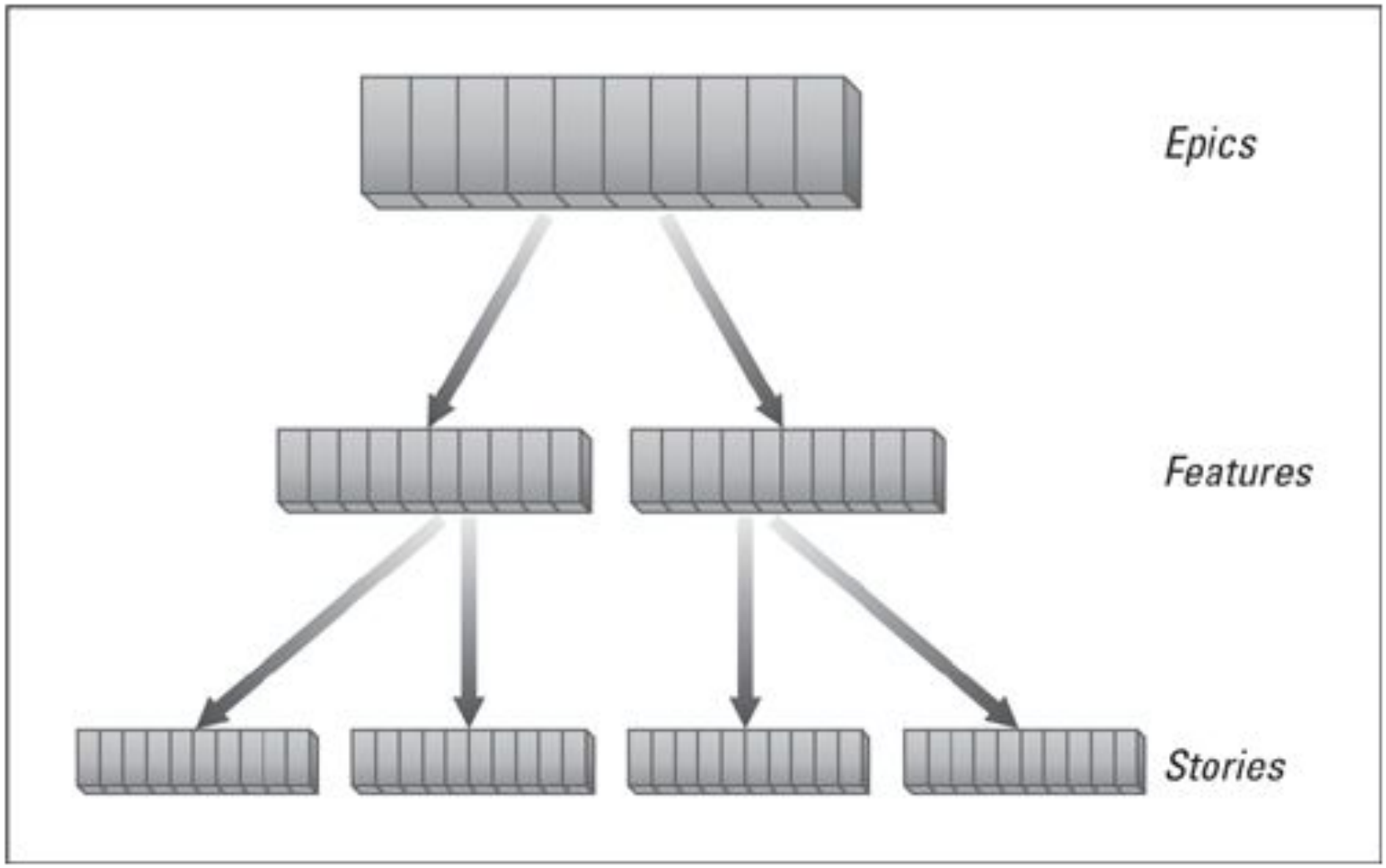
- SCRUM-Master
- System Team
- Product Manager
- Operations

Contracts and Accounting

- Fixed price project
- Cooperation model
  ("Time and Material")

REQUIREMENTS, QS, TRANSPARENCY

Requirement Engineering


Epics
Features
Stories

Type of Information	Description	Responsibility	Time Frame and Sizing	Expression Format	Testable
Investment theme	*Big*, audacious, game changing, initiatives. Differentiating, and providing competitive advantage.	Business executives, Portfolio management.	Span strategic planning horizon, 12 to 18+ months. Not sized, controlled by percentage investment.	Any: text, prototype, PPT, video, conversation.	No
Epic	Bold, impactful, marketable differentiators.	Portfolio management. Business analysts, product and solution management, system architects.	6 to 12 months. Sized in points.	Most any, including prototype, mock-up, short phrase, or vision statement.	No
Feature	Short, descriptive, value delivery and benefit-oriented statement. Customer and marketing understandable.	Product manager and product owner.	Fits in an internal release (PSI), divide into incremental subfeatures as necessary. Sized in points.	Key phrase or user story voice form. May be elaborated with system use cases.	Yes
Story	Small, atomic. Fit for team and detailed user understanding.	Product owner and team.	Fits in a single iteration. Sized in story points.	User story canonical form.	Yes

User Stories

As WHO I want WHAT so that WHY

Acceptance Criteria

Small, “one card”

INVEST

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

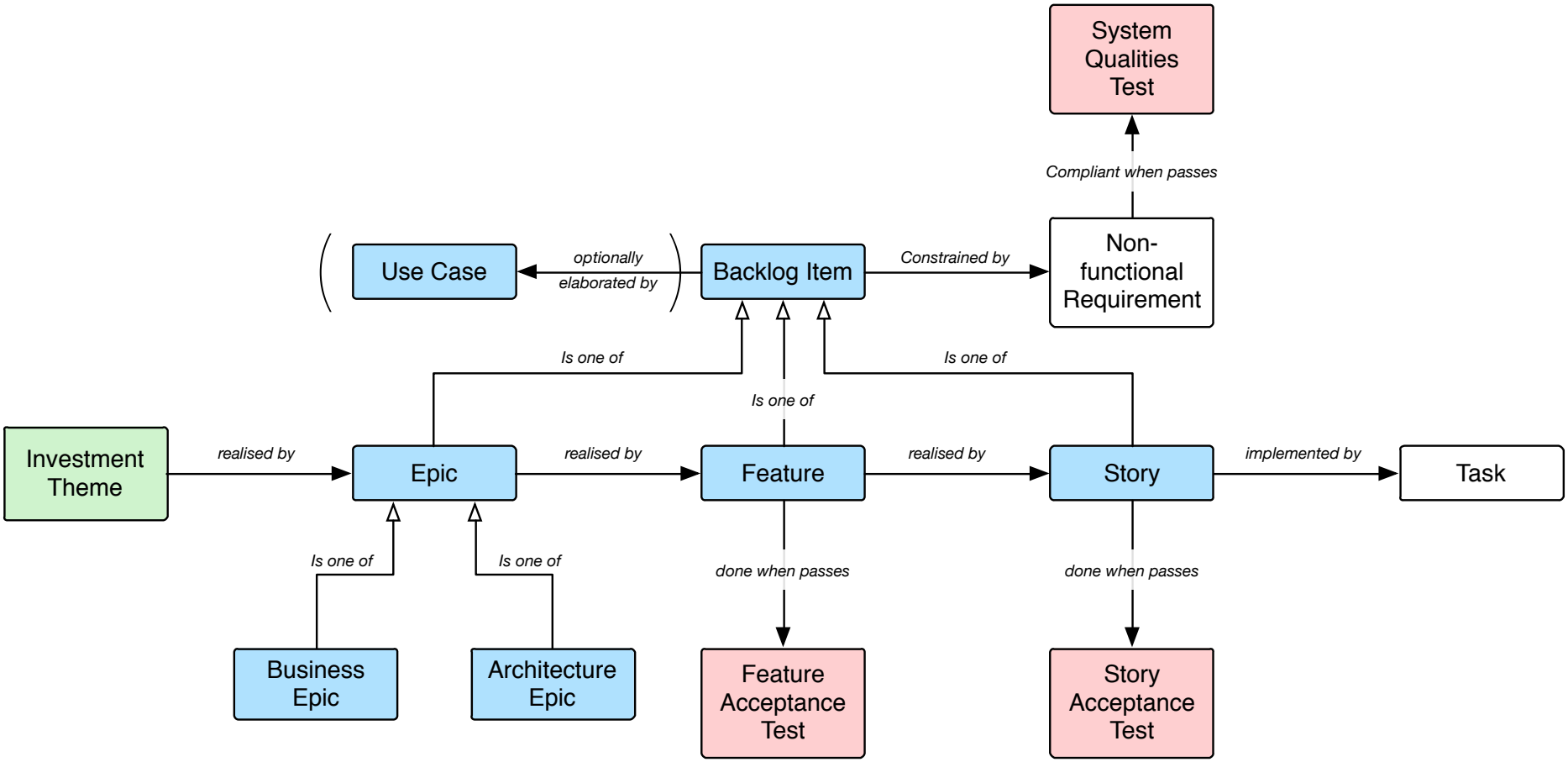
Estimation


?
0
½
1
2
3
5
8
13
20
40
100
∞
Books by Mike Cohn
Our Agile Courses
Succeeding with Agile®

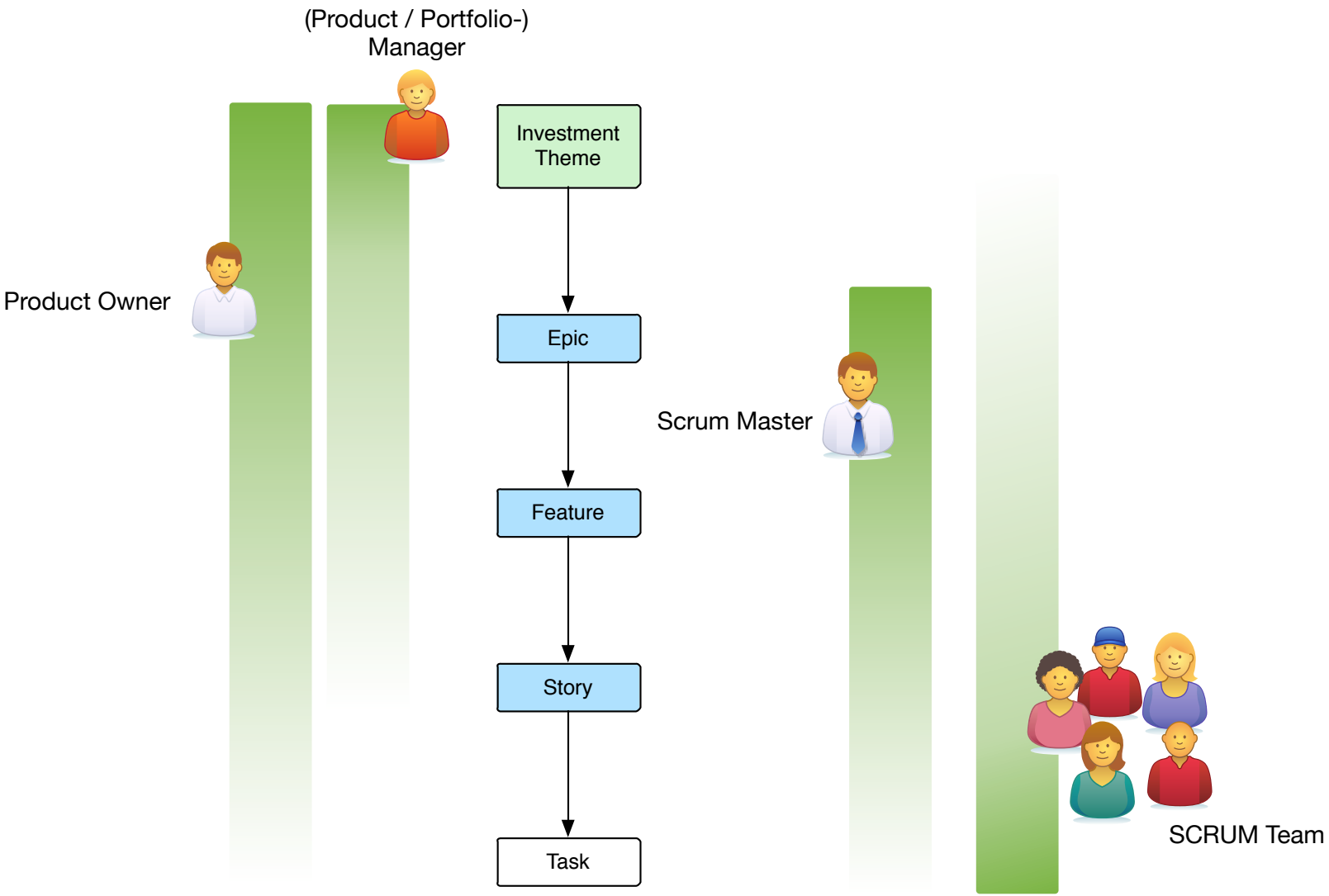
SAFE Framework

- Vision and Roadmap
- Release Management
- Deployment
- Resource Management
- Cross-team tasks

Full Enterprise Requirement Model (SAFE Framework)

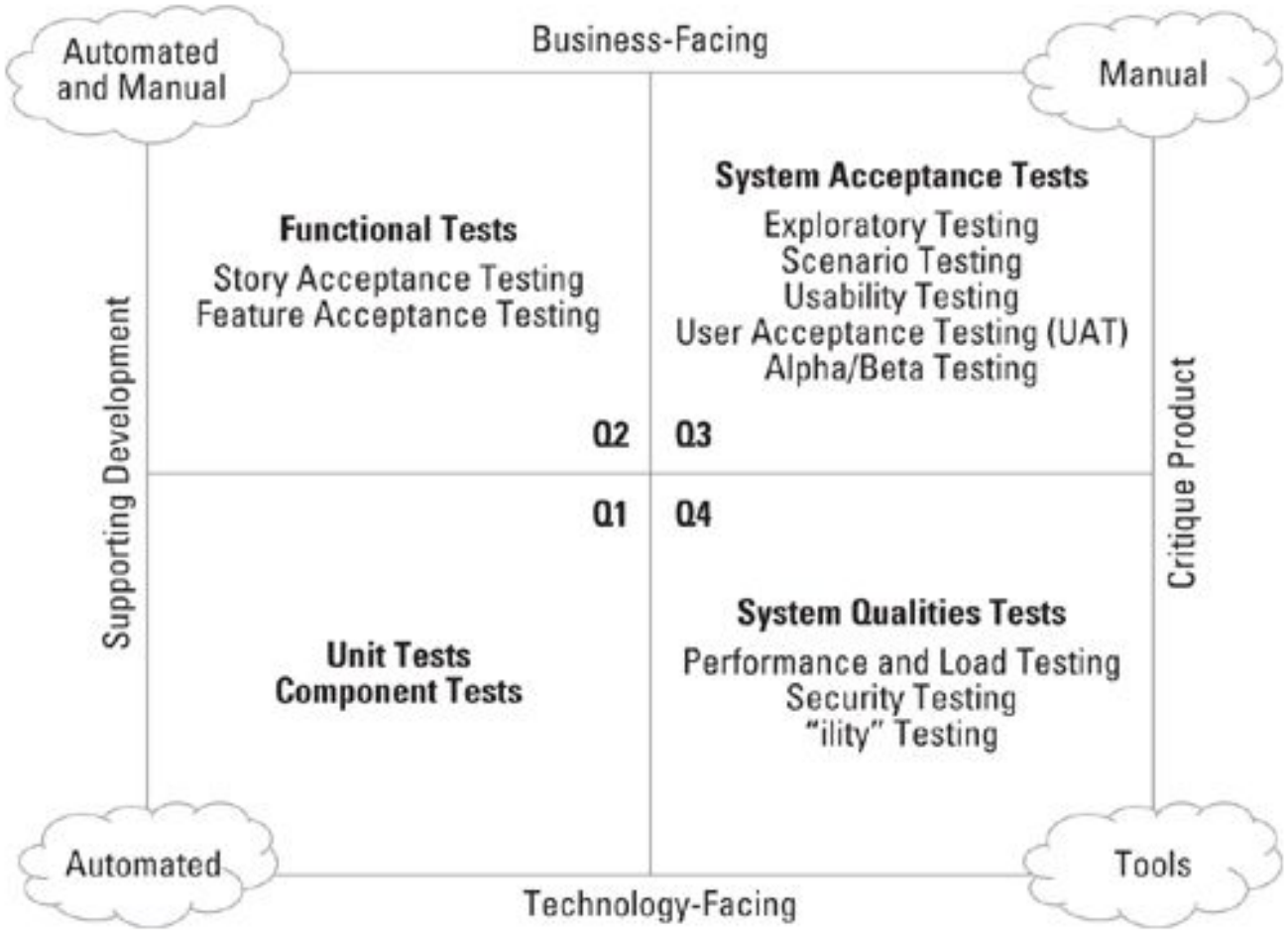


Roles and “Stories”



Testing

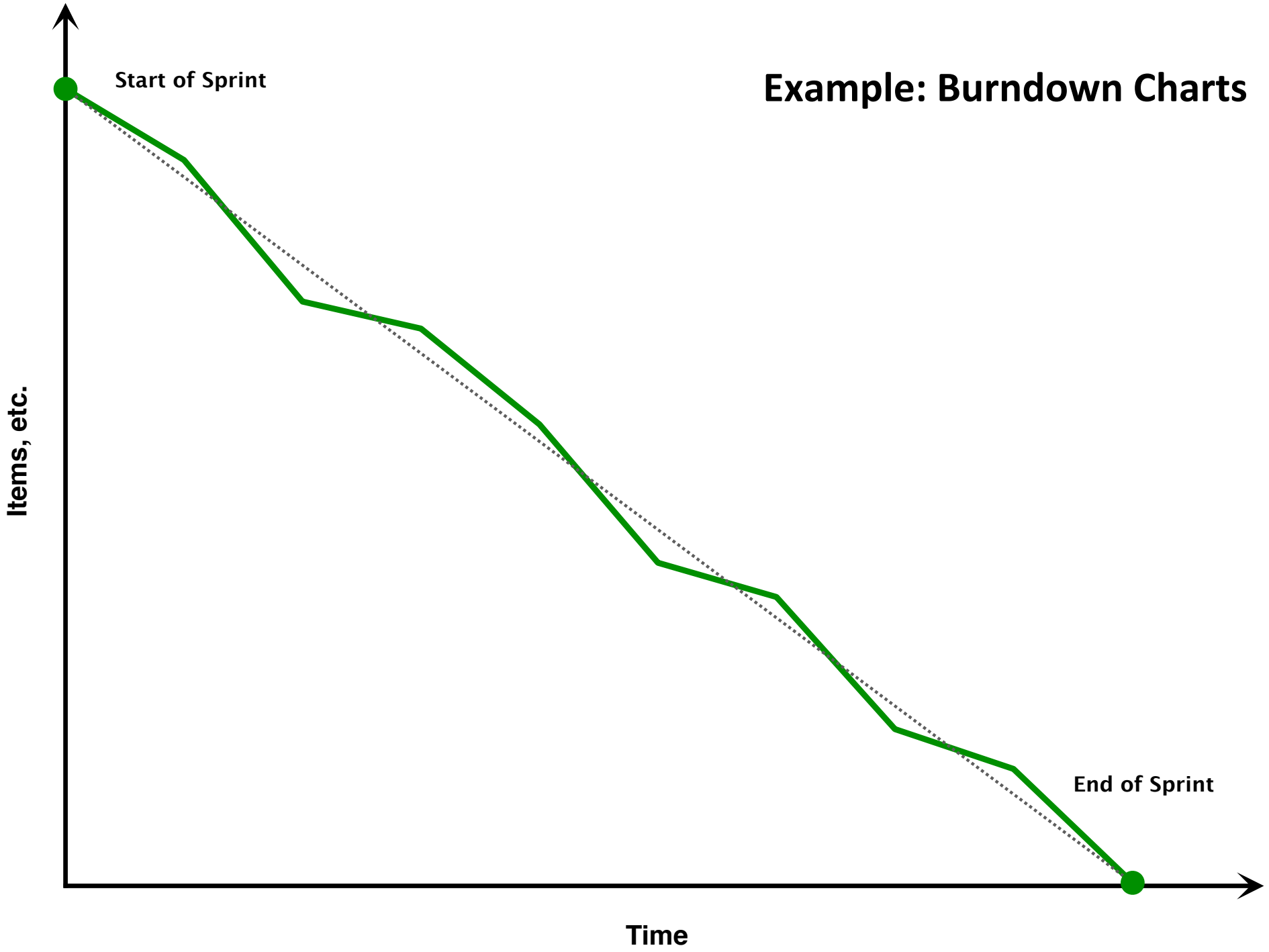
Testing is Integral Part of each Sprint

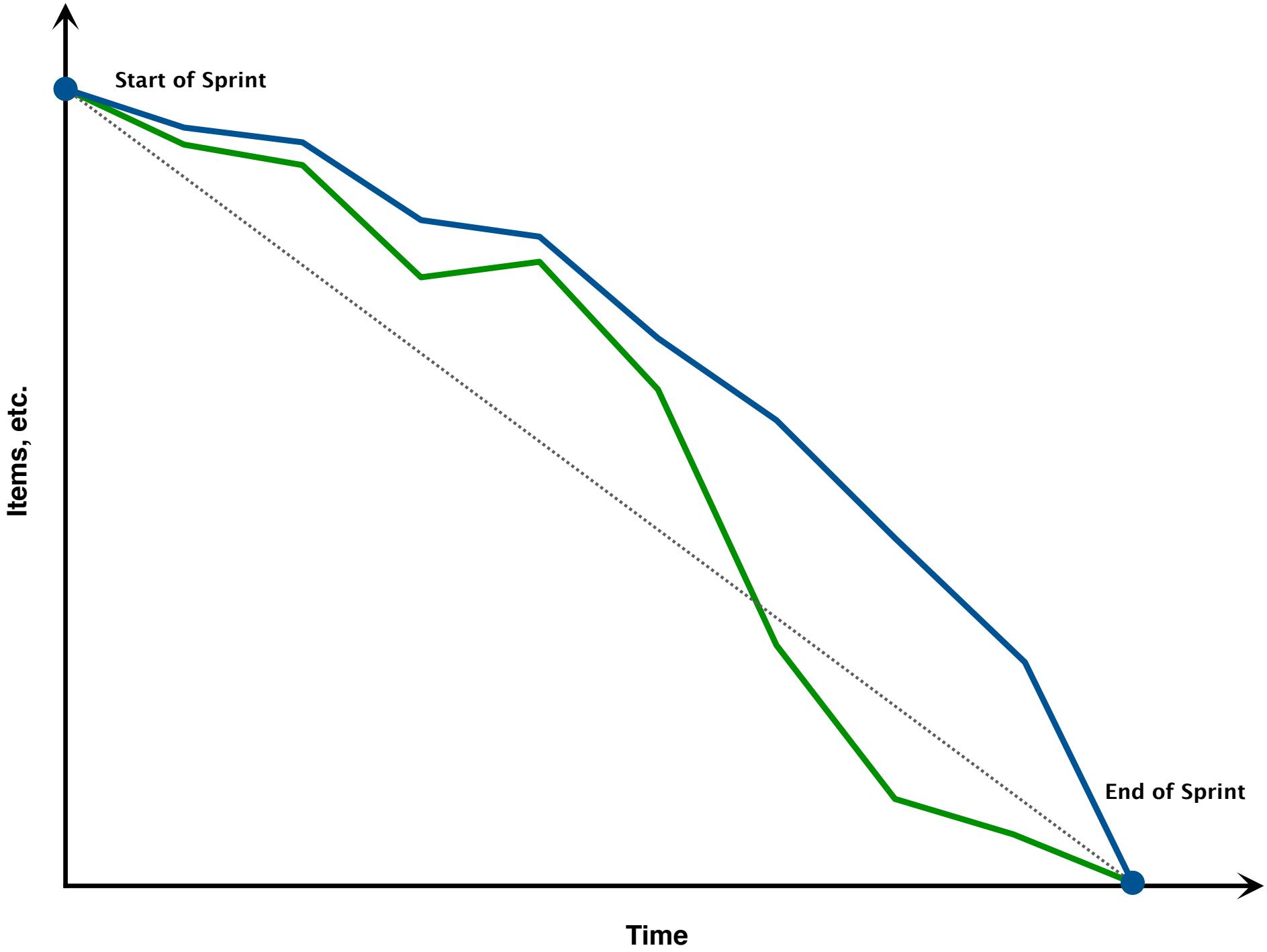


Transparecy

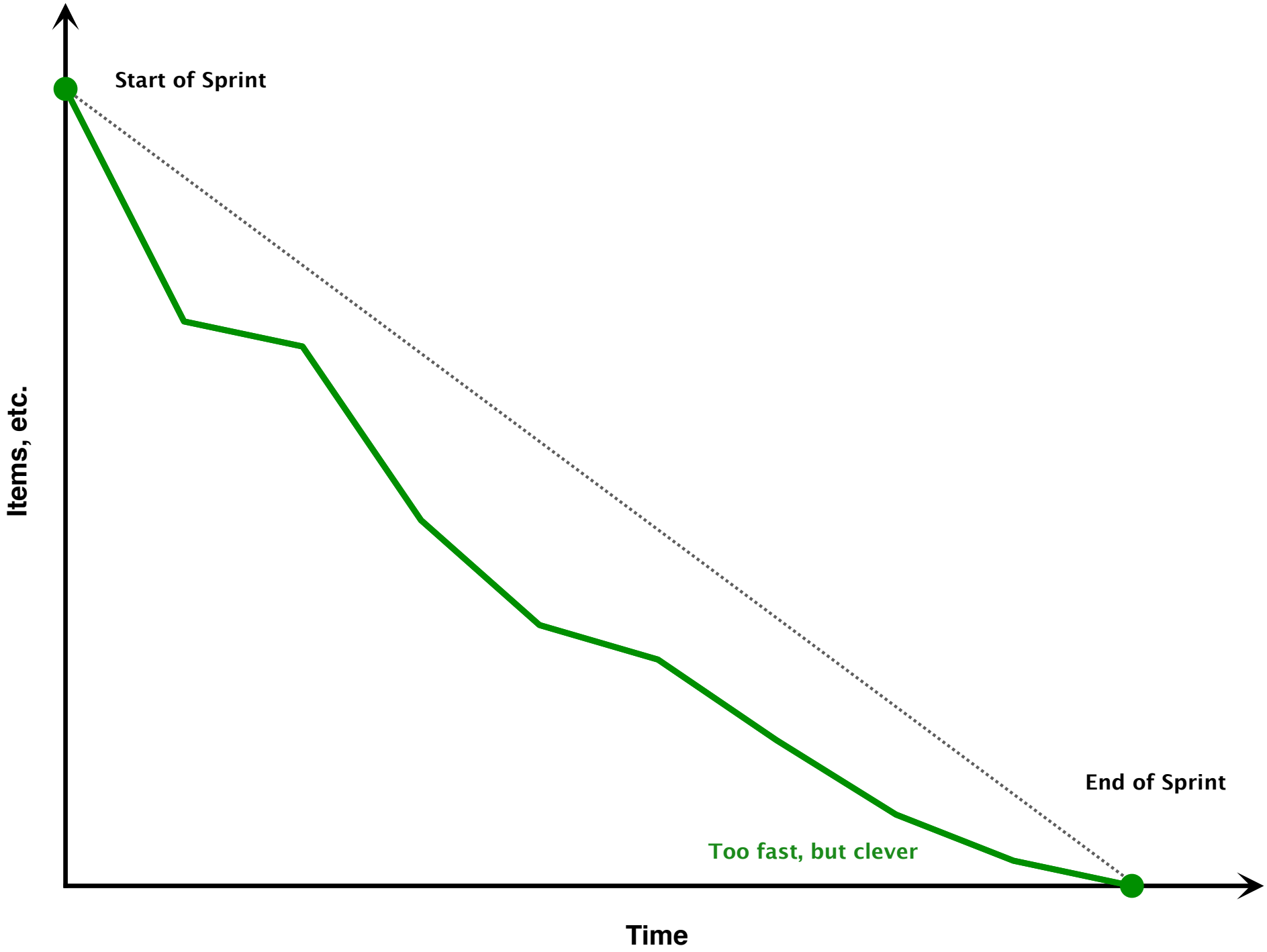
Controlling: Burn Down Charts

- One option to visualise progress
- Y-axis: items/features (e.g. “Story Points”, person days)
- X-axis: time
- Update after each sprint
- Improves team self-assessment
- Realistic estimation of project duration and assessment of changes in project scope
- Velocity (KPI): work per iteration: measured, not estimated (!)

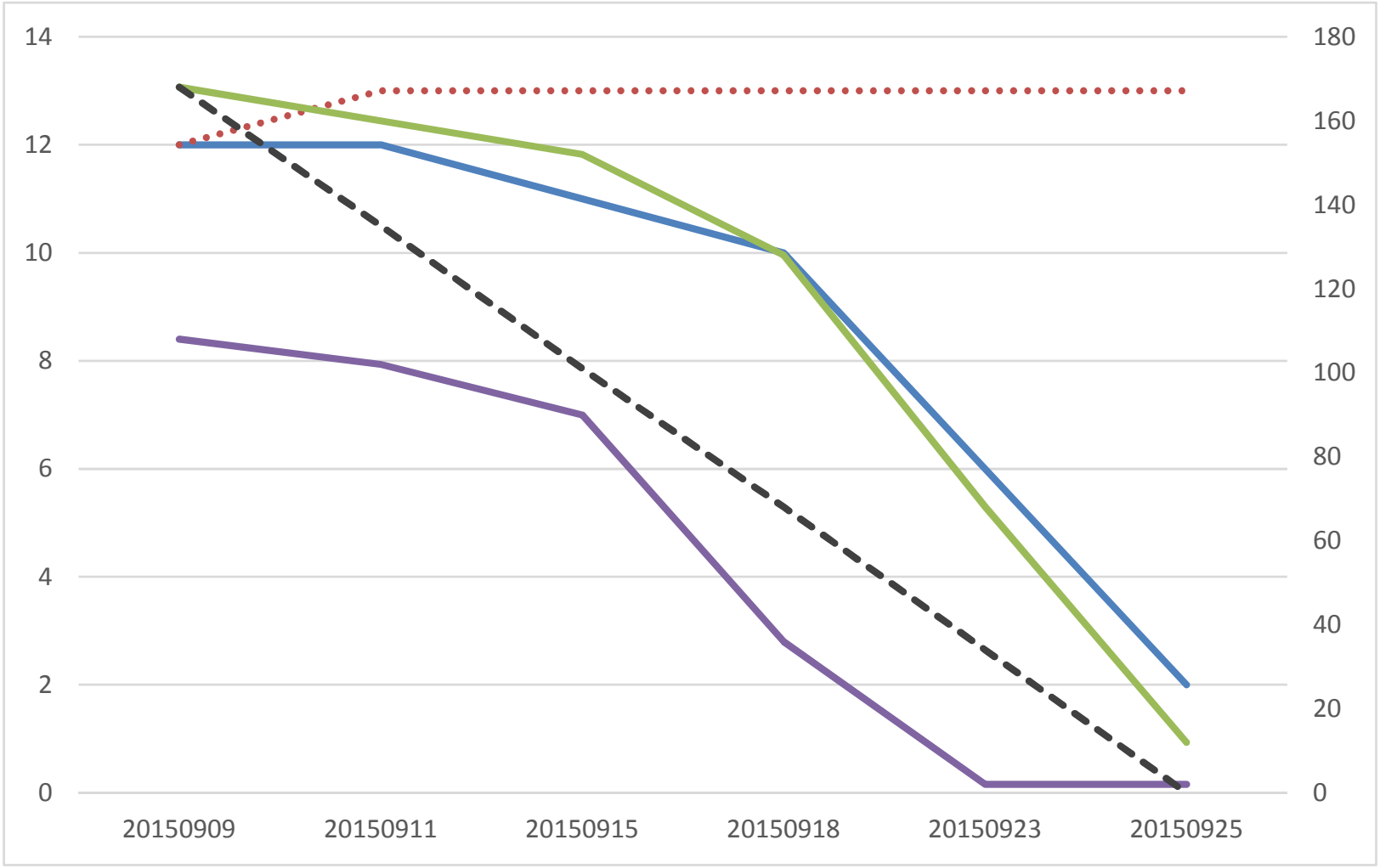

Example: Burndown Charts
Start of Sprint
End of Sprint
Items, etc.
Time


Start of Sprint
End of Sprint
Items, etc.
Time

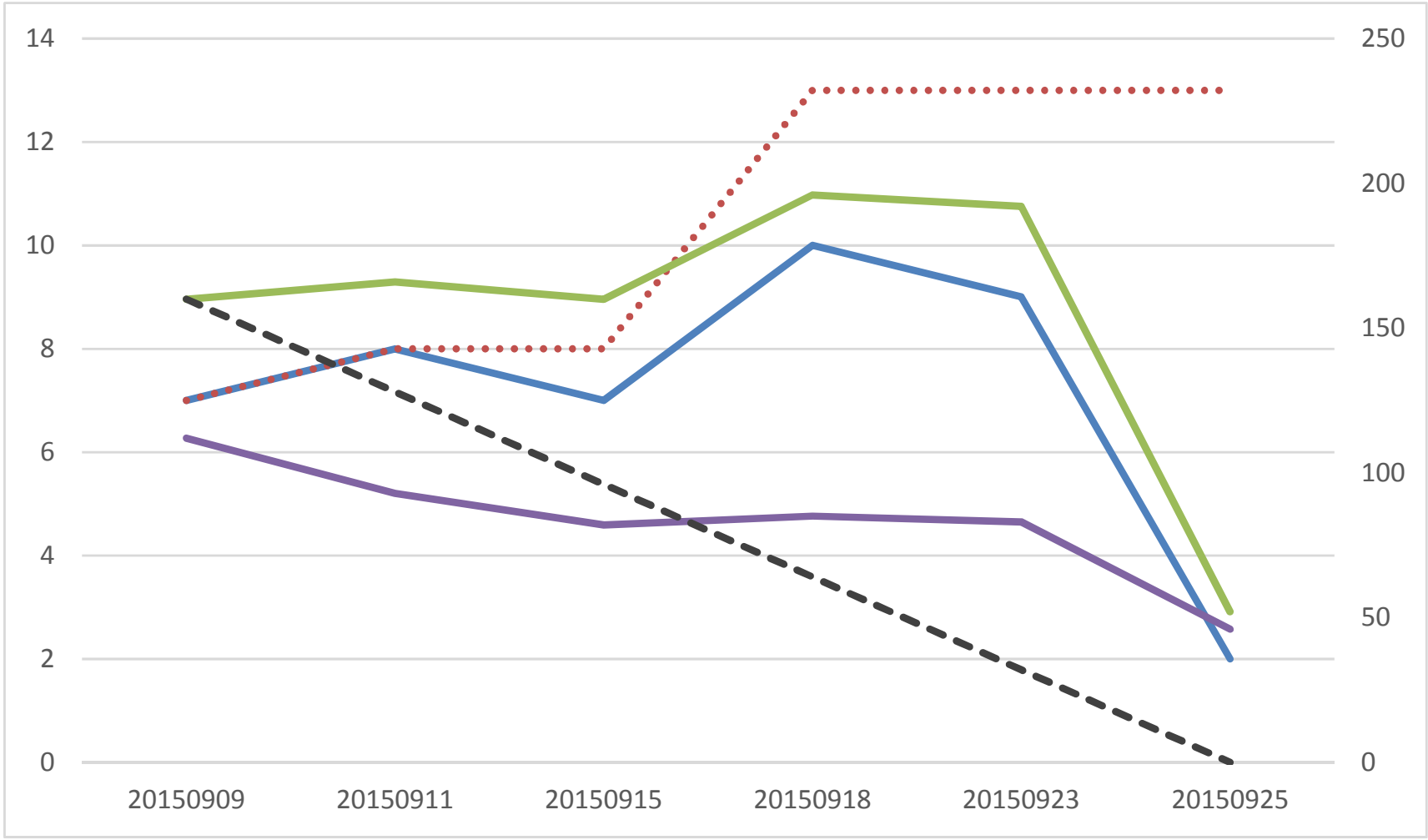

Start of Sprint
Items, etc.
Too slow
"Too fast"
End of Sprint
Time


Start of Sprint
Items, etc.
End of Sprint
Too fast, but clever
Time

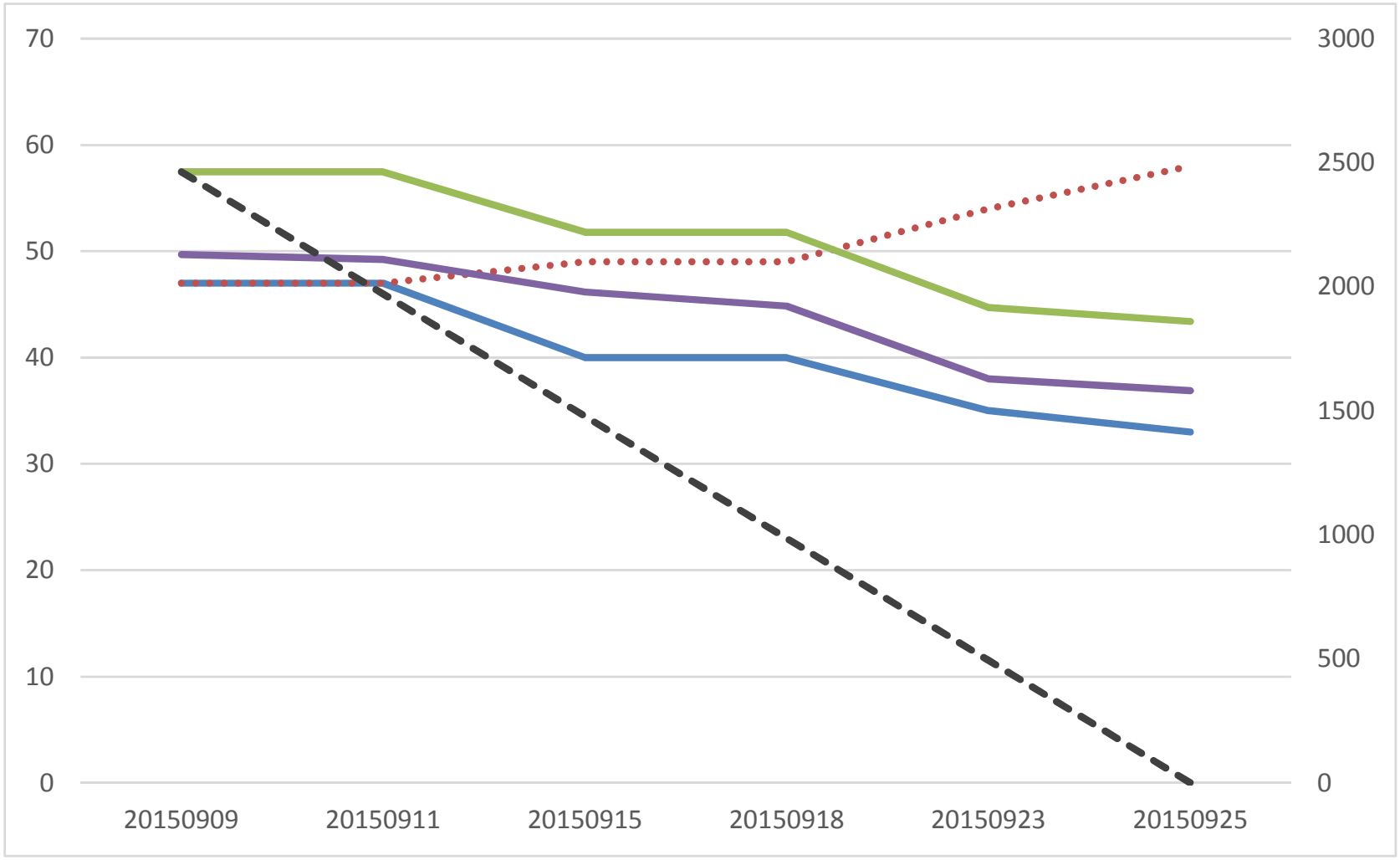
Example Burn Down: Team 1



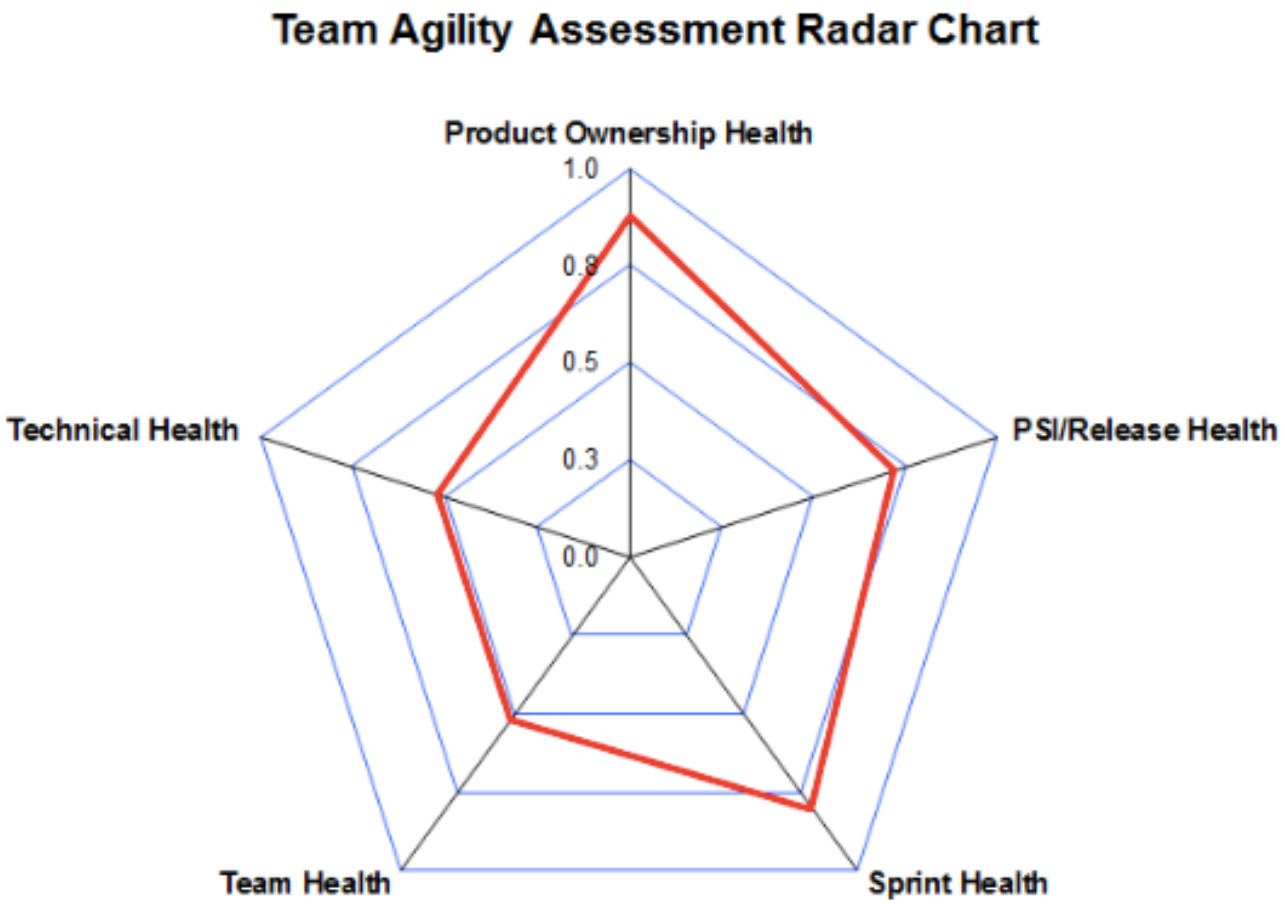
Example Burn Down: Team 2



Example Burn Down: Team 3



SAFE Self-Assessment Metrics



Conclusion
Agile Metrics (KPIs)

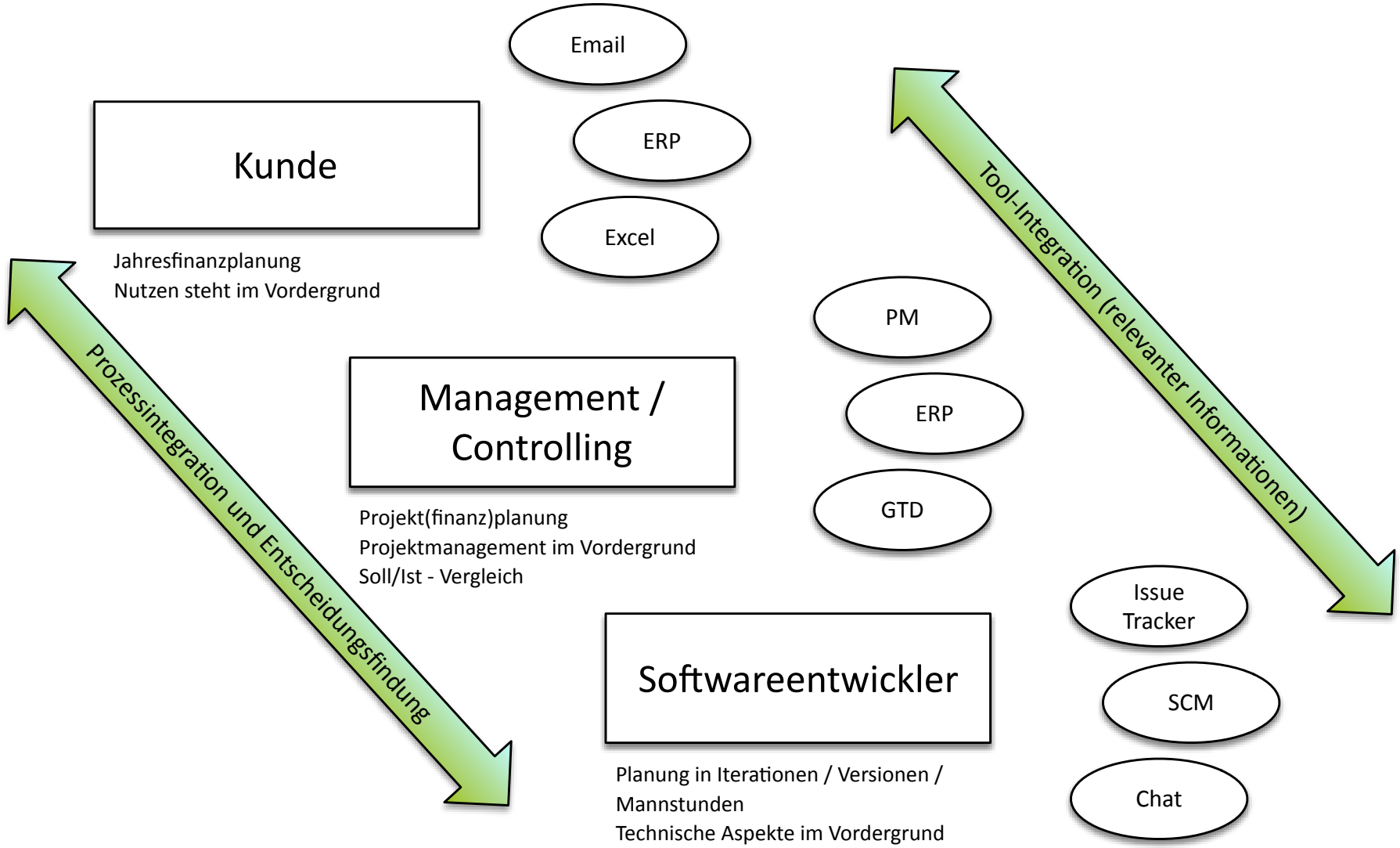
- Different options to visualise progress
  - Burn-Down Charts
  - Prozess-flow visualisation
  - Cumulative Flow Diagrams
- Various KPIs
  - Velocity (aggregated): *items per iteration*
  - Velocity per work type
  - Cycle Time/Lead Time: *Average completion time of one item*
  - Identification of bottlenecks (queue length)
  - (Defect Rates)
- Usefulness of metrics/KPIs depending on
  - Project environment
  - Concrete process implementation
  - Maturity of teams / process implementation
- Cross team metrics / KPIs, e.g. story points?
  - Can be used but has risks

Reviews / Audits

- Process quality
- QA/reporting discipline
- Code quality
- Test quality
- Documentation quality
- Interviews with
  - Product owner
  - Customer
  - Team members

Challenges and Risks

- Introduction / change management
- Process not implemented properly / consequently
- Lack of transparency
- Role of product owner / customer
- Team structure not clear enough
- Team roles still focused on “silos”
- Inadequate technical implementation


Email
ERP
Excel
Kunde
Jahresfinanzplanung
Nutzen steht im Vordergrund
Tool-Integration (relevanter Informationen)
Prozessintegration und Entscheidungsfindung
PM
ERP
GTD
Management / Controlling
Projekt(finanz)planung
Projektmanagement im Vordergrund
Soll/Ist - Vergleich
Issue Tracker
SCM
Chat
Softwareentwickler
Planung in Iterationen / Versionen / Mannstunden
Technische Aspekte im Vordergrund

Confidence Building

- Claim and change management as part of the process
- Frequent releases and demos
- Customer relationship
- Backlog transparency
- Testing and QA
- Agile metrics / KPIs
- Audits

Conclusion: Cornerstones

- Agile or plan-driven?
- Agile is the opposite of sloppy!
- Transparency over following a (long term, improbable) plan
- Agile – not only for small teams
- Requirements and customers
- Roles and processes

Agile Softwareentwicklung im Konzernumfeld

Dr. Alexander Schatten
biac (Vienna Insurance Group)
alexander.schatten@biac.at
alexander@schatten.info
http://www.schatten.info