Course "Softwareprozesse"

Non-standard agile processes: ScrumBut and agile transition

Lutz Prechelt
Freie Universität Berlin, Institut für Informatik

- Observed deviations from Scrum in Scrum teams w.r.t.
  - Roles
  - Daily Scrum, Sprint Planning, Sprint Review
  - (curiously not Retrospectives)
- Discussion when they may be good or bad

- Agile transition as a gradual process in 5 dimensions

Learning objectives

- When and how deviating from Scrum may be a good idea
  - for a would-be Scrum team
- When and how deviating from Scrum may be a bad idea
- How becoming agile is a gradual process
  - with 5 identifiable separate concerns

Scrum parts reinforce each other

- The Scrum Guide states:
  - *"Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum.*
  - *Changing the core design or ideas of Scrum,*
    leaving out elements, or
    not following the rules of Scrum,
    covers up problems and limits the benefits of Scrum,
    potentially even rendering it useless."
  - *"While implementing only parts of Scrum is possible,*
    the result is not Scrum."

- But such changes are made frequently
  - Let's discuss some common ones:

ScrumBut

- "We do Scrum, **but**…" [for example:]
  - "…we don't need Retrospectives"
  - "…the Sprint Backlog is provided by management"
  - "…some people never come to the Daily Scrum"
  - "…we use a flow system instead of Sprint Planning"
  - etc, etc, etc.

- The resulting process may be an improvement or a weakness, sometimes resulting in a pseudo-agile crippling.
  - Pseudo-agile: violates the spirit of the Agile Manifesto.

Sources

- Zainab Masood, Rashina Hoda, Kelly Blincoe:
  "Real World Scrum: A Grounded Theory of Variations in Practice", IEEE Transactions on SW Engineering 2022
  - based on direct observations of 5 teams and 45 interviews
    - India, New Zealand, Pakistan, UK

- Mohamad Mortada, Hamdy Michael Ayas, Regina Hebig:
  "Why do Software Teams Deviate from Scrum? Reasons and Implications",
  Intl. Conf. on SW and Systems Process 2020
  - based on direct observations of 2 teams and
    a quantitative survey among 34 (out of 123) Scrum users
    - Sweden
    - (subsequent percentages are from the survey, >33% is **marked**)
  - found 4/7/2 deviations in
    Daily Scrum/Sprint Planning/Sprint Review

Our question

- Under which circumstances will this Scrum deviation be
  - a substantial mistake?
    - then: What are likely reasons for the deviation?
  - a substantial improvement?

?

- We discuss this question for each deviation

Scrum Master

- No Scrum Master after a previous one left
  - may be OK for a very mature team
  - probably a big mistake for most teams
- Explicitly no Scrum Master
  - probably a big mistake for most teams
- Role handed around among experienced Developers
  - may be OK for a very mature team
  - a dubious solution for most teams
    - may be better to share a full-time Scum Master among 2-3 teams

Developers

Scrum Guide:

- *"Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. […] The specific skills needed by the Developers are often broad"*

- …have different roles
  - such as tester, developer, business analyst

- …have strong specializations
  - such as front-end vs. back-end

- OK iff everybody can contribute fully
  - but that is difficult!

Daily standup

- …often happens later than the scheduled time (**35%**)
  - inefficient
- …takes longer than 15 minutes (**53%**)
  - inefficient
    - and bad for motivation if very long
- Not all team members contribute (**41%**)
  - dangerous
- …is a open discussion, not a structured one (**65%**)
  - inefficient, dangerous

Sprint Planning

Who:

- Sprint Backlog populated by only 1 or a few team members
- Work assigned by team lead, not self-assigned by Developer

What:

- Multiple Sprint Goals specified
- No Sprint Goal specified (**38%**)
  - or only at the end of Sprint Planning (**44%**)
- No Product Backlog, only a Sprint Backlog (**44%**)
- Sometimes no estimate for story (**62%**)

How:

- Product Backlog items (stories) not broken down into tasks for Sprint Backlog (**56%**)
- Meetings over 8 hours (26%)
- Available resources not determined (**47%**)

Sprint Review

- Not performed (**44%**)
- Not always feedback from customers or stakeholders (**62%**)
- Depends!
  - Effort may lack direction
  - But with well-informed Product Owners all may go well

Sprint Retrospective

Table 3: Retrospective observation template

Total Time	Used Method	Used Questions	Notes
How long does the team retrospective meeting take?	How does the team decides the important topics to discuss further?	What questions does the team retrospective answers?	For more notes if available

- *"none of the deviations observed during the retrospective was frequently enough confirmed in the online survey to be reported in this paper"*

Agile transition

- The agile, self-organized and self-managed way of working needs to become **accepted** and **learned**.
  - This takes some time and effort.
  - So organizations (and teams) undergo a period of *agile transition*.

- Rashina Hoda, James Noble: "Becoming Agile: A Grounded Theory of Agile Transitions in Practice", Intl. Conf. on SW Engineering 2017
  - based on team observations and 31 interviews
    - India and New Zealand/Australia/USA/India/Portugal
  - finds **5 dimensions of change** that needs to be made
    - teams can be at any point in each dimension
    - sliding back is possible

  (subsequent images are from that article)

Dimension 1: Development practices

Fig. 2. Transitioning Software Development Practices (TSDP)

- Initially (Stage 1, S1):
  - *"getting acquainted with basic agile development practices such as the iterative and incremental delivery"*
  - *"[Teams often retain] many of their earlier ways of working."*
  - *"[Individuals may be] change resistant"*
- S2, S3: *"[Later on, Teams would use e.g.]*
  - *pair programming, user stories and tasks, testing,*
  - *frequent releases, daily stand up meetings, and in some cases, retrospectives."*

Dimension 2: Teams' task-determination practices

Fig. 3. Transitioning Team Practices (TTP)

- Regards
  - *"requirements specification, prioritization, and clarification",*
  - *"effort estimation"*, and
  - in particular *"task assignment"*
- S1: individuals are risk-averse, prefer less responsibility
- S2: they transition or get transitioned:
  - *"I actually set a formal goal for [cross-functionality in] the KPIs [...] You should be […] taking on big development even though you are a good database guy.*
    – P2, Project Manager, New Zealand"
    - KPI: Key Performance Indicator

Dimension 3: Managers' role

Fig. 4. Transitioning Management Approach (TMA)

- Concerns roles or initial roles such as
  - Project Managers, Subject-matter Experts, Team Leads
  - also Scrum Masters
- S2 e.g.
  - "*My team is very easily distracted by what is going on outside the project […] that is where […] I have to sometimes revert to classic-style management [...] command and control mode* – P2, Project Manager, New Zealand"
- S3: "*[Our project manager] clears operational or process-related blocks that hinder [the] completion of a task.* – P16, Senior Developer, USA"

Dimension 4: Reflection and learning

Fig. 5. Transitioning Reflective Practices (TRP)

- S1:
  - Teams strongly implementation-focused.
    Agile trainings seen as a waste of time.
  - Few of them hold retrospectives.
- S2: Agile improvements become viewed as a to-do
  - "*We are now working on process improvements*"
- S3:
  - Nearly all teams hold retrospectives.
  - "*Somebody who just learned new technology […] speaks on Saturdays once a month. First half is theory and second half is hands-on [...] everybody would come and learn.*
    – P22, Senior Developer, India"

Dimension 5: Operational culture

Fig. 6. Transitioning Culture (TC)

- Operational work culture, resulting from
  - organizational culture, team culture, individual traits
- S1: Decisions and communication along a hierarchy:
  - "*if we [cannot] deliver [on time, we'll] document the reason [...] After getting the approval from the managers, we [send it] to the product owner who [explains the problem] to the clients.*
    – P6, Developer, India
- S3: "*So we can decide whether we can make this addition or to reject it [...] This is not done by any external influence but it is purely us.*
  – P7, Tester, India

Reminder:
Definition "Culture" [ShwBel15]

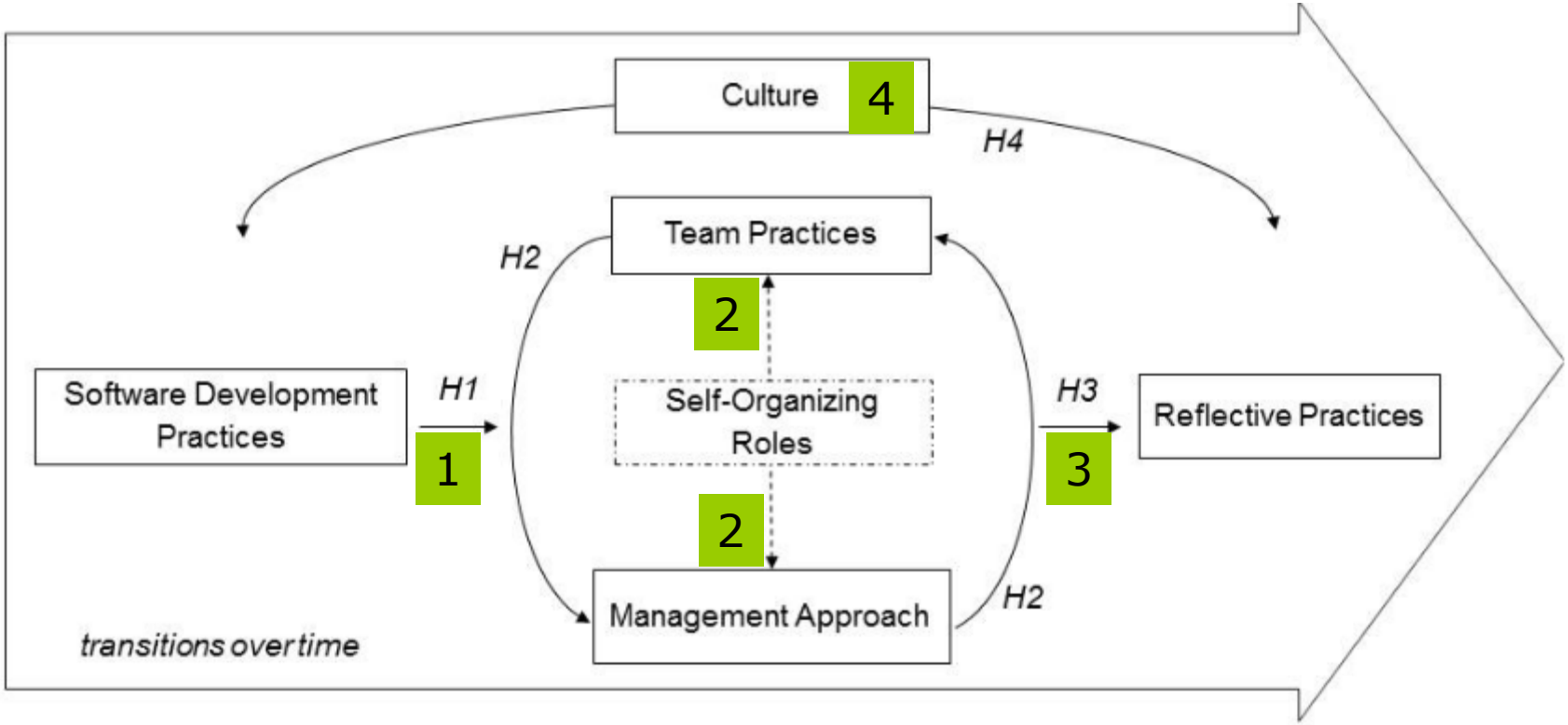
Freie Universität Berlin

- *"Explicit and implicit patterns of behavior"*
  - that constitute achievements of a human group and
- *"traditional ideas and […] their attached values"*.
  - *"Conventional understandings manifest in act and artifact."*
- ***"'culture' refers to community-specific ideas about what is true, good, beautiful, and efficient."***

Discussion:

- So the ideas change during an agile transition.
  What about the underlying facts?
- Whether one culture is preferable over another depends on
  - context (e.g. team, code base, rqmts., surrounding organization)
  - the values held by the observer.

Conjecture(!) how agile transition happens

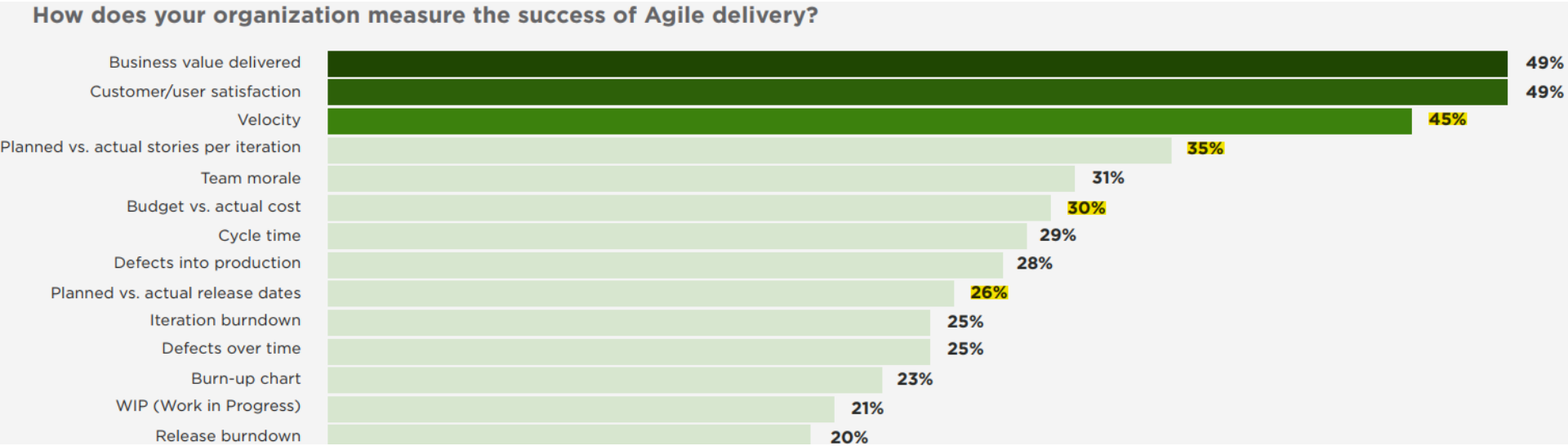


1. SDPs must change first
2. TPs and MA influence each other, mediated by self-org roles
3. RPs transition only after TPs and MA have transitioned somewhat
4. Org., team, and indiv. culture mediate everything ?

Success measures in practice

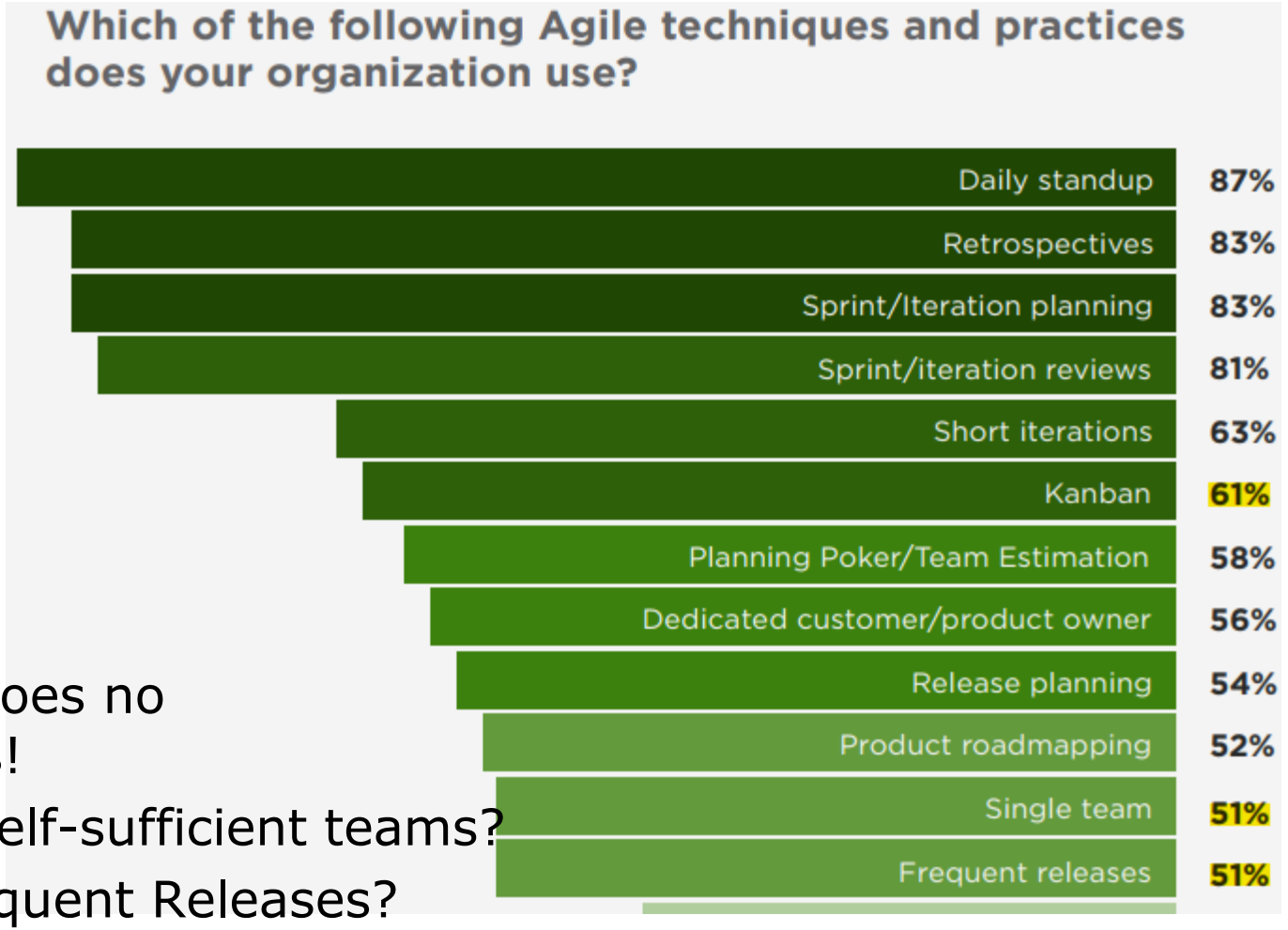
Freie Universität Berlin

Data from the 2021 "State of Agile" report



- The marked metrics are questionable at best, yet common

Practices used in practice



- 1 in 6 teams does no Retrospectives!
- Half have no self-sufficient teams?
- Half do no Frequent Releases?

("Kanban" is probably often just a Kanban board)

Summary

- Deviating from Scrum is often a bad idea
  - because it breaks how Scrum is supposed to achieve agility
- Deviating from Scrum can sometimes be a sensible idea
  - if it is well-matched to the context
- Becoming agile involves a gradual change of
  - work practices,
  - management practices (on teams' as well as managers' sides),
  - reflection and learning practices, and
  - operational culture

Thank you!