
Einführung in das Scrum Framework

&

welche 10 Praktiken helfen, Scrum
wirklich gut zu machen

Wer bin ich

Jean Pierre Berchez

HLSC
Dornhalde 1
70597 Stuttgart
0711-766408

Mobile +49-160-4704532

Email: jp.berchez@scrum-events.de

Web: www.scrum-events.de
www.scrum-dav.de



training and certification programs by Scrum.org™

SCRUMevents

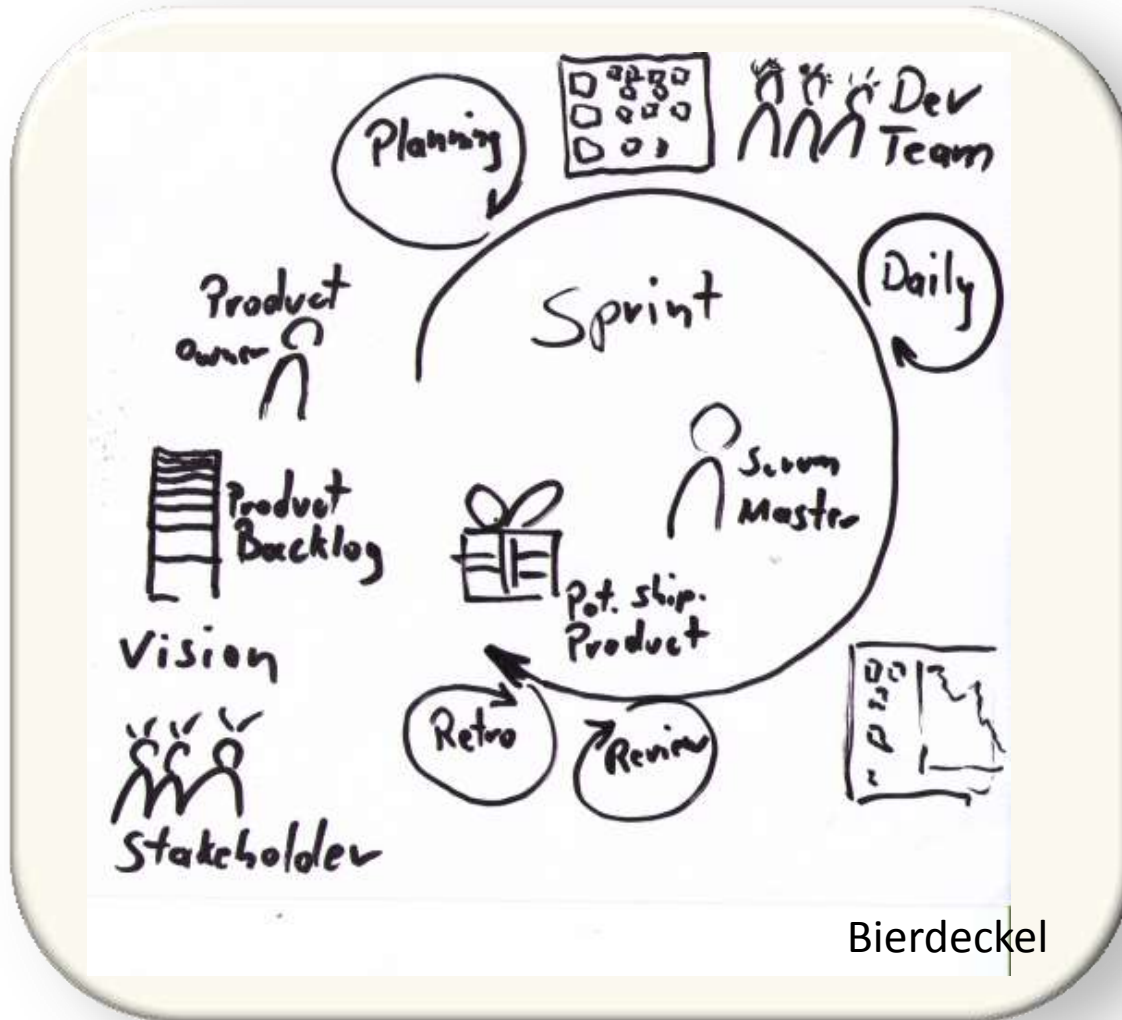


- Kurse und Vorträge mit Jeff Sutherland und Ken Schwaber
- Verschiedene Kurse der Scrum.org
 - Professional Scrum Master
 - Professional Scrum Product Owner
 - Scrum Developer
 - Professional Scrum Foundations
- Management 3.0 Kurse
- Seminare zum „Scaled Agile Framework“ (Dean Leffingwell)
- Scrum Consulting & Coaching
- Veranstalter Scrum-Day (Konferenz rund um Scrum)
- Weitere Infos unter www.scrum-events.de und www.scrum-day.de

Agenda

- Was ist Scrum?
 - (oder Scrum auf einem Bierdeckel)
- 10 Tipps die helfen Scrum richtig gut zu machen

Scrum auf einem Bierdeckel



10 Tipps die helfen Scrum richtig gut zu machen

1. Selbst Organisation
2. Backlog – Items sind “READY” bei Sprint Start
3. PSPI’s sind „DONE“ am Sprint Ende
4. Product Backlog Grooming
5. Puffer für Refactoring, Bugs etc. einplanen
6. Visualisiere deinen Fortschritt
7. Häufige Collaboration und Kommunikation
8. Pair „Developing“
9. Wenig Störungen
10. „Inspect & Adapt“ vs. „Adapt & Inspect“

Selbst Organisation



- ✓ Teams sind X-Functional
- ✓ Ein Team arbeitet als Team und nicht als Ansammlung von Personen (Spezialisten, Egoisten etc.)
- ✓ Ein Team erledigt die höchst priorisierte Story zuerst (und zwar gemeinsam) und testet sofort (und nicht am Ende des Sprints)
- ✓ Ein Team “committed” sich zu einem “Sprint Goal”
- ✓ Alle Arbeit (um ein “fertiges Product zu liefern) gehört dem Team und muss vom Team erledigt werden.

Backlog – Items sind “READY” bei Sprint Start



- ✓ Stories Identifiziert, Priorisiert, und Geschätzt
- ✓ Alle ausgewählten Stories haben Akzeptanz Kriterien
- ✓ Alle Team Mitglieder (Analysten, Codierer, Tester etc.) und der Product Owner haben ein gemeinsames Verständnis über die Story bevor diese in den Sprint Backlog geht.

PSPI's sind „DONE“ am Sprint Ende

- ✓ Potentially Shippable Product Increment
- ✓ Alle Tests
 - ✓ Unit Tests
 - ✓ Akzeptanz Tests
- ✓ Live Demo für Product Owner und Stakeholder im Review Meeting



Product Backlog Grooming während des Sprints



- ✓ Allokierte Zeit vom Team und Product Owner um über zukünftige Dinge bzgl. des Products zu kommunizieren
- ✓ Neue Stories Epics im Backlog
- ✓ Die „oberen“ Stories werden verfeinert bis diese „Ready“ sind

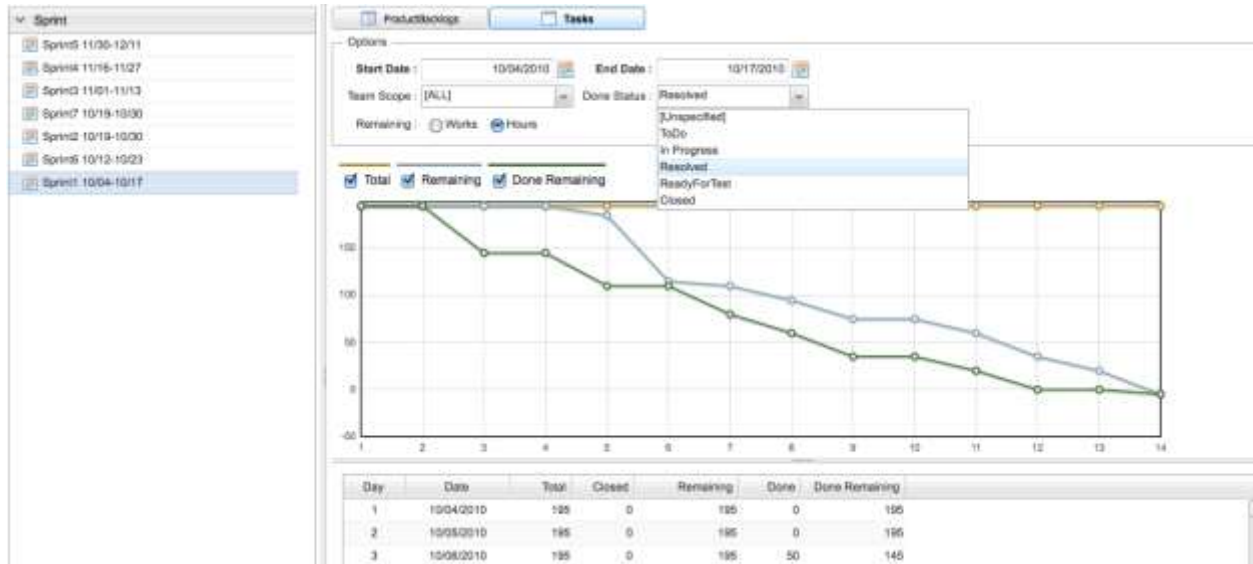
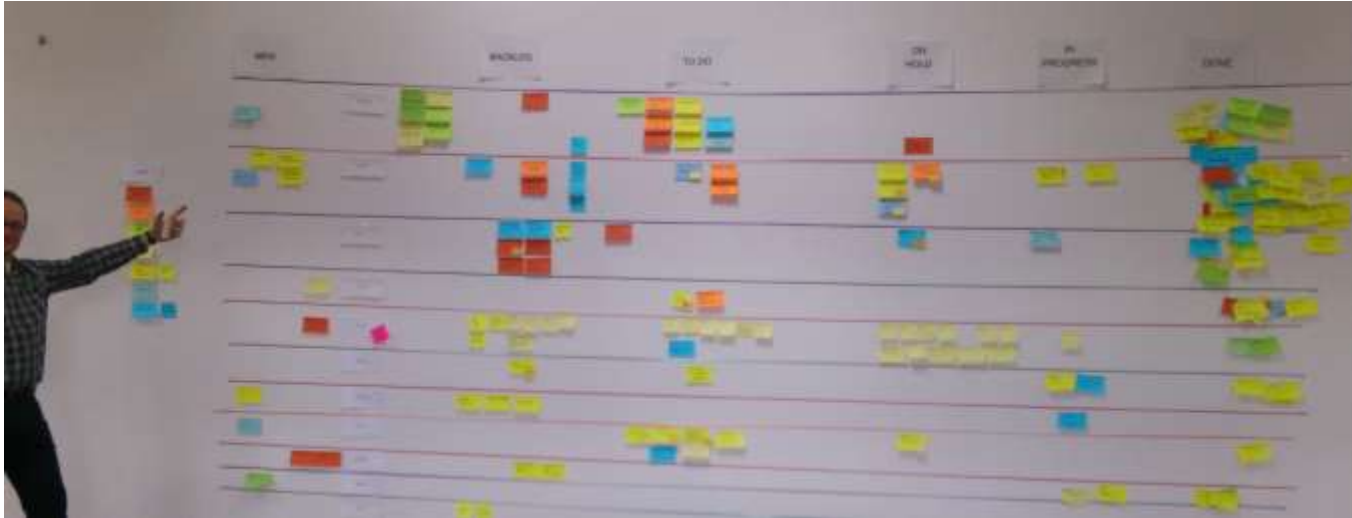
Puffer für Refactoring, Bugs etc. einplanen



- ✓ Puffer einplanen für:
 - Refactroing
 - Bugs
 - Backlog-Grooming/Refinement
 - „Unplanned Items“
- ✓ Puffer visualisieren (Transparenz schaffen)



Visualisiere den Fortschritt



Häufige Collaboration und Kommunikation

- ✓ Alle Scrum Meetings
 - ✓ Daily, ✓ Planning
 - ✓ Review, ✓ Retrospective
- ✓ Grooming
- ✓ User Story Workshops
- ✓ Alle Scrum-Team Mitglieder arbeiten eng zusammen während Sprints -> Pair Programming
- ✓ Kurze Sprintdauer hilft (zwei Wochen)
- ✓ Wikis, Tracker, Instant Messaging, Continuous Integration



“Pair Developing”



- ✓ Team Mitglieder arbeiten in Pairs bis eine Story DONE ist.
- ✓ Wissensvermittlung während “produziert” wird.
- ✓ Hilft Konsistent zu sein.
- ✓ Personal Risiken werden vermieden.
- ✓ Ideal bei Spezialistentum.

Wenig Störungen

- ✓ Prioritäten ändern sich nicht innerhalb eines Sprints.
- ✓ Störungen von aussen werden auf ein Minimum reduziert (z.B. Produktions Support)
- ✓ Änderungswünsche werden über den Product Owner kanalisiert.
- ✓ Ungeplante Aufgaben transparent machen (ggf. Entstehende Kosten aufdecken)



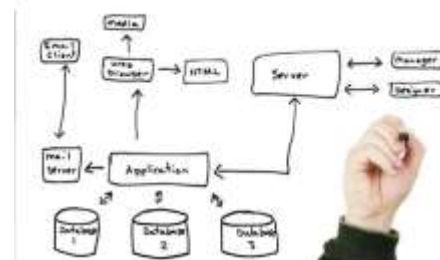
“Inspect & Adapt” vs. “Adapt & Inspect”



- ✓ Scrum sollte so angewandt werden wie es das Framework vorsieht.
- ✓ Scrum „And“ ist gut Scrum „But“ ist schlecht.
- ✓ Scrum Teams müssen gut geschult sein
 - ✓ ScrumMaster Training für agiles coaching
 - ✓ Product Owner Training für Product Owner
 - ✓ XP / Engineering Training für Dev und QA Teams etc.

Zugabe

- ✓ Gute Engineering Practices helfen:
 - ✓ Continuous Integration
 - ✓ Automated Unit and Acceptance Testing
 - ✓ Push-Button Releases
 - ✓ Test-Driven Development
 - ✓ Clean Code
 - ✓ Refactoring



Tipps zur Anwendung



- ✓ Auf wenig Anpassungen fokussieren
- ✓ Ggf. in kleinen Schritten agieren. Eine Änderung pro Sprint danach “Inspect & Adapt”
- ✓ Wenn nötig Experten hinzuziehen
- ✓ “Educate yourself” (Web, Konferenzen, Trainings)

Danke

Q&A