

GOOD PRACTICE
Warum „good“ und nicht „best“?

Eine komplexe Software-Entwicklung besteht aus vielen unterschiedlichen Puzzleteilen, die immer wieder eine andere Form und Farbe annehmen. Deswegen möchten wir Anstöße geben, um in bestimmten Situationen erfolgreich zu sein.

Dies heißt jedoch nicht, dass deine Probleme automatisch, auch wenn sie ähnlich sind, gelöst werden können. Das Umfeld, die Mitarbeitenden und die Infrastruktur spielen eine sehr große Rolle, die ebenfalls betrachtet werden muss.

Viel Spaß beim Ausprobieren, wir freuen uns auf Feedback jeglicher Art.

GOOD PRACTICE
Testautomatisierung (Regressionstests)

„Fall fast.“ – Autor unbekannt

Wir empfehlen automatisierte Tests, da sie es ermöglichen, Fehler im Code und der Funktionalität schnell zu erkennen und auszubessern. Getestet werden nicht nur Neuentwicklungen, sondern auch Auswirkungen auf das gesamte Produkt. Dabei ist der Kosten-/Nutzen-Faktor zu betrachten: Da eine Testautomatisierung aufwändig sein kann, konzentriert man sich auf die wichtigen Funktionen, die entscheidend für die Nutzung des Produktes sind (risikobasiert).

Automatisierte Tests ersetzen manuelle Tests nicht vollständig. Wir empfehlen, automatisierte Tests möglichst früh in der Entwicklung zu berücksichtigen und Akzeptanzkriterien als Grundlage zu nutzen.

GOOD PRACTICE
DevOps

„You build it, you run it.“ – Werner Vogels

Wir lieben DevOps, da sich das Team ganzheitlich um die Anwendung kümmert und komplexe Software-Überabgabeprozesse nicht mehr notwendig sind. So liegt die Verantwortung bei den Experten, die sie entwickelt haben.

Von der Entwicklung, über die Integration der Komponenten, zum Go-Live und weiter zur Überwachung der laufenden Anwendung – alles liegt in der Verantwortung des Teams.

GOOD PRACTICE
Continuous Integration & Delivery

„The most powerful tool we have as developers is automation.“ – Scott Hanselman

Wir nutzen Continuous Integration, um sicherzustellen, dass unser Code nach jedem Commit noch lauffähig ist und unsere Qualitätskriterien (z.B. Testabdeckung) weiterhin erfüllt sind. Eine automatisierte Pipeline hilft uns dabei.

Continuous Delivery ist der automatisierte Schritt des Deployments der Anwendung in die Produktion, sobald alle notwendigen Qualitätskriterien erfüllt sind.

GOOD PRACTICE
BDD (Behaviour Driven Development)

„Es hört doch jeder nur, was er versteht.“ – J.W. Goethe

Wir nutzen Behaviour Driven Development in Verbindung mit einer „abstrakten language“, also einer allgegenwärtigen, für alle verständlichen und auf der Fachdomäne basierenden Sprache, um Verständigungsprobleme zwischen Fachexperten, Entwicklern und anderen Stakeholdern zu minimieren und ein gemeinsames Verständnis von der Anwendung zu entwickeln.

Da die Tests während oder besser noch vor der Entwicklung des Codes erstellt werden und das Verhalten der Anwendung in einer semi-formalen Syntax (z.B. gherkin) beschreiben, ist allen Beteiligten von Anfang an klar, wie die Anwendung nach der Implementierung eines neuen Features funktionieren soll.

GOOD PRACTICE
Mob Programming/Pair Programming

„Erfolg ist mehr als die Summe seiner Teile.“ – Aristoteles

Für uns ist Mob Programming zu einem wichtigen Bestandteil des Entwicklungsalltags geworden. Wir finden es klasse, wenn mehrere Entwickler gemeinsam an einem Problem arbeiten können. Nicht nur die Qualität des Codes wird (bereits während seiner Erstellung) gesichert, auch Wissenssinn innerhalb des Teams werden vermieden.

GOOD PRACTICE
Burndown Chart

„Erfolg ist eine Treppe, keine Tür.“ – Dottie Walters

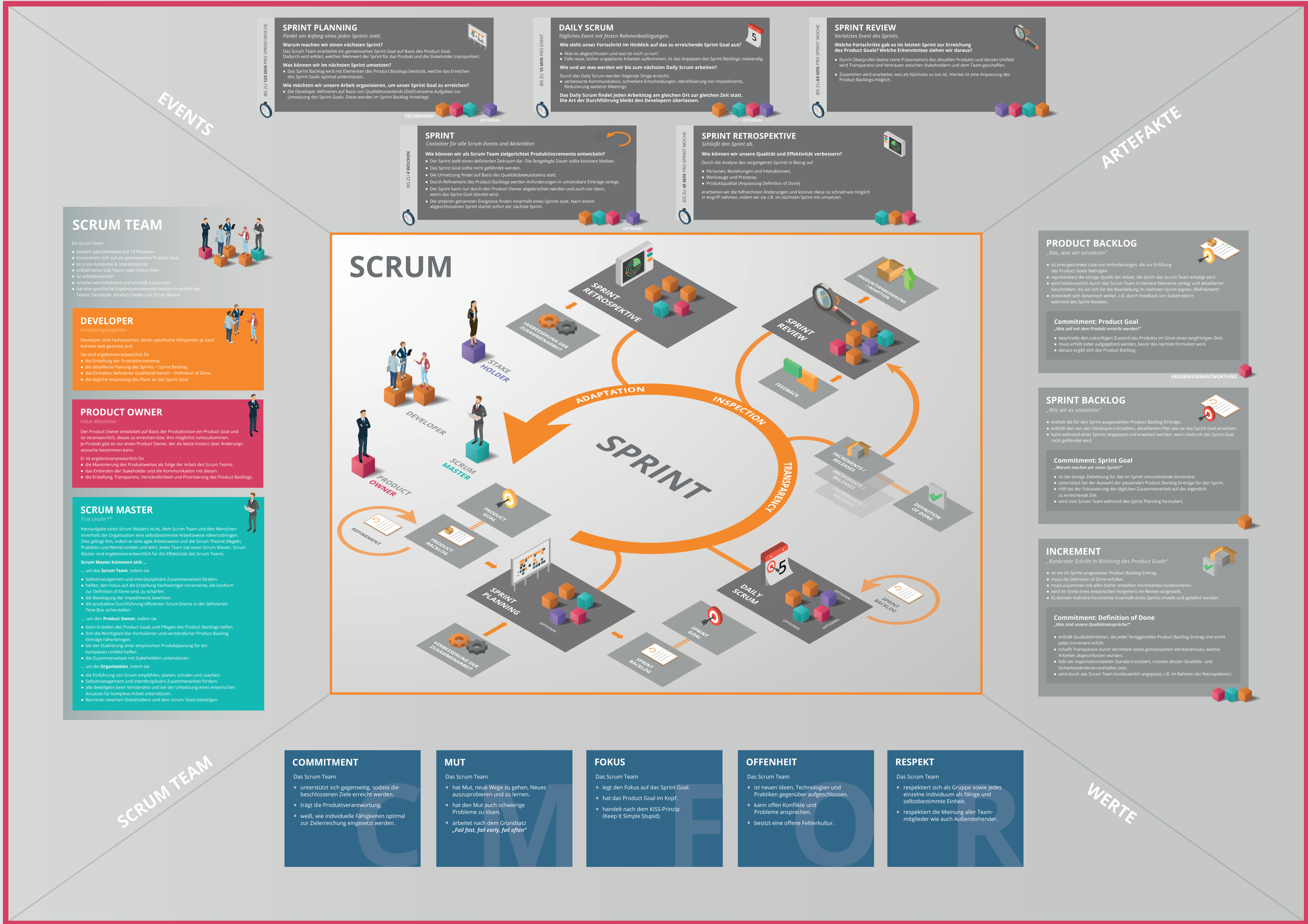
Wir empfehlen, den Fortschritt des Teams während eines Sprints mit einem Burndown Chart zu visualisieren. Dabei wird jeden Tag die geleistete Arbeit in Relation zu dem Zeitverlauf des Sprints gesetzt, beginnend mit der voraussichtlichen Gesamtarbeitsmenge. Dies bietet einen schnellen Überblick auf die noch zu leistende Arbeit und hilft zu erkennen, ob Impediments das Team blockieren und wie gut die Arbeitspläne aufgeteilt sind.

Der Graph wird jeden Tag aktualisiert und ähnelt im Idealfall einer Treppe oder einer Treppe (bei Ja).

***SELBSTBESTIMMT**
Das Scrum Team ist in allen produktbezogenen Aktivitäten selbstbestimmt.

Wir verstehen darunter, dass die Organisation sämtlicher Arbeit (wie, wann, von wem und wo) komplett in der Teamverantwortung liegt. Leitplanken und grundlegende Qualitätskriterien werden in Zusammenarbeit mit Stakeholdern definiert, um eine gemeinsame Vision/Richtung zu entwickeln.

Sämtliche Tätigkeiten, die dazu notwendig sind das Produkt zu entwickeln und zu pflegen, obliegen der Teamhoheit.



SCRUM TEAM

Ein Scrum Team

- besteht typischerweise aus 10 Personen
- konzentriert sich auf ein gemeinsames Product Goal
- ist cross-funktional & interdisziplinär
- erhält keine Sub-Teams oder Hierarchien
- ist selbstbestimmt
- arbeitet wertschöpfend und sinnhaft zusammen
- hat drei spezifische Ergebnisverantwortlichkeiten innerhalb des Teams: Developer, Product Owner und Scrum Master

DEVELOPER
 Umsetzungsexperten

Developer sind Fachexperten, deren spezifische Fähigkeiten je nach Inkrement weit getrennt sind, aber selbst ergebnisverantwortlich für:

- die Erstellung der Produktinremente
- die detaillierte Planung des Sprints - Sprint Backlog
- die Erreichung definierter Qualitätskriterien - Definition of Done,
- die tägliche Anpassung des Plans an das Sprint Goal

PRODUCT OWNER
 Hobst Maximierer

Der Product Owner entwickelt auf Basis der Produktion ein Product Goal und ist verantwortlich, dieses zu erreichen bzw. ihm möglichst nahezukommen. Ihr Produkt gibt es nur einen Product Owner, der als letzte Instanz über Änderungswünsche bestimmen kann.

Er ist ergebnisverantwortlich für:

- die Maximierung des Produktwertes als Folge der Arbeit des Scrum Teams,
- das Einbeziehen der Stakeholder und die Kommunikation mit diesen,
- die Erstellung, Transparenz, Verständlichkeit und Priorisierung des Product Backlogs.

SCRUM MASTER
 Team Leader*

Kernaufgabe eines Scrum Masters ist es, dem Scrum Team und den Menschen innerhalb der Organisation eine selbstbestimmte Arbeitsweise näherzubringen. Dies gelingt ihm, indem er eine agile Arbeitsweise und die Scrum Theory (Ritual, Praktiken und Werte) vorlebt und lehrt. Jedes Team hat einen Scrum Master. Scrum Master sind ergebnisverantwortlich für die Effektivität des Scrum Teams.

Scrum Master kümmern sich...

- um das Scrum Team, indem sie
 - Selbstmanagement und interdisziplinäre Zusammenarbeit fördern,
 - helfen, den Fokus auf die Erstellung hochwertiger Increments, die konform zur Definition of Done sind, zu schaffen,
 - die Beseitigung der Impediments bewirken,
 - die produktive Durchführung effizienter Scrum Events in der definierten Time Box sicherstellen.
- um den Product Owner, indem sie
 - beim Erstellen des Product Goals und Pflegen des Product Backlogs helfen,
 - ihm die Wichtigkeit klar formulieren und verständlicher Product Backlog Einträge nahebringen,
 - bei der Stabilisierung einer empirischen Produktplanung für ein komplexes Umfeld helfen,
 - die Zusammenarbeit mit Stakeholdern unterstützen.
- um die Organisation, indem sie
 - die Einführung von Scrum empfehlen, planen, schulen und coachen,
 - Selbstmanagement und interdisziplinäre Zusammenarbeit fördern
 - alle Beteiligten beim Verständnis und bei der Umsetzung eines empirischen Ansatzes für komplexe Arbeit unterstützen,
 - Barrieren zwischen Stakeholdern und dem Scrum Team beseitigen.

COMMITMENT

Das Scrum Team

- unterstützt sich gegenseitig, sodass die beschlossenen Ziele erreicht werden,
- trägt die Produktverantwortung,
- weiß, wie individuelle Fähigkeiten optimal zur Zielerreichung eingesetzt werden.

MUT

Das Scrum Team

- hat Mut, neue Wege zu gehen, Neues auszuprobieren und zu lernen,
- hat den Mut auch schwierige Probleme zu lösen,
- arbeitet nach dem Grundsatz *„Fail fast, fail early, fail often“*

FOKUS

Das Scrum Team

- legt den Fokus auf das Sprint Goal,
- hat das Product Goal im Kopf,
- handelt nach dem KISS-Prinzip (keep It Simple Stupid).

OFFENHEIT

Das Scrum Team

- ist neuen Ideen, Technologien und Praktiken gegenüber aufgeschlossen,
- kann offen Konflikte und Probleme ansprechen,
- besitzt eine offene Fehlerkultur.

RESPEKT

Das Scrum Team

- respektiert sich als Gruppe sowie jedes einzelne Individuum als fähige und selbstbestimmte Einheit,
- respektiert die Meinung aller Teammitglieder wie auch Außenstehender.

GOOD PRACTICE
Refinement Event

„Ein Backlog ist nie komplett und nie vollständig.“ – Tim Buchner

Wir empfehlen regelmäßige Refinement Events, wenn das Team es noch nicht gewohnt ist, sich ausreichend mit der Weiterentwicklung des Backlogs zu beschäftigen. Es stellt ein gemeinsames Zeitfenster dar, in dem sich das Scrum Team mit dem Backlog auseinandersetzen kann. Dieses kann um Details wie Beschreibung, Reihenfolge und Größe ergänzt werden. Die Attribute variieren oft je nach Arbeitsumfeld.

Ein solches Event ist auch dann sinnvoll, wenn man Stakeholder außerhalb des Teams in das Refinement miteinbeziehen möchte.

GOOD PRACTICE
Referenz Product Backlog Einträge

„Denken heißt vergleichen.“ – Wolther Rathenau

Wir nutzen Referenz Product Backlog Einträge, um für die Schätzung des Aufwands neuer Einträge eine gesunde Basis zu schaffen. Dadurch fällt es Teams deutlich leichter, neue Anforderungen in Relation zu diesen einzuschätzen.

Wichtig ist es immer wieder zu überprüfen, ob bisherige Referenz Product Backlog Einträge noch als solche sinnvoll sind; andere technologische Schwerpunkte oder ein tieferes Verständnis des Produktes und der verwendeten Technologien können Komplexitäten verändern.

GOOD PRACTICE
Planning Poker

„Keep it simple and get rid of the big cards.“ – Autor unbekannt

Wir benutzen Planning Poker für die Schätzung unserer Product Backlog Einträge, um Meinungen und Experten einzelner Teammitglieder besser zur Geltung zu bringen.

Während eines Refinement Events werden Diskussionen durch sehr unterschiedliche Schätzungen angestoßen und vornehmlich unabhängige Gedankengänge im Team zugelassen.

****LEADER**

Leader sind für uns Menschen, die durch ihr Verhalten und ihre Taten andere Leute motivieren, ein gemeinsam definiertes Ziel zu verfolgen. Sie inspirieren und begeistern durch den Drang etwas zu verändern, zu verbessern und zu bewegen.

Jeder kann ein Leader in einem gewissen Kontext sein.