



SCRUM Product Owner Schulung

SCRUM

SCRUM in a Nutshell

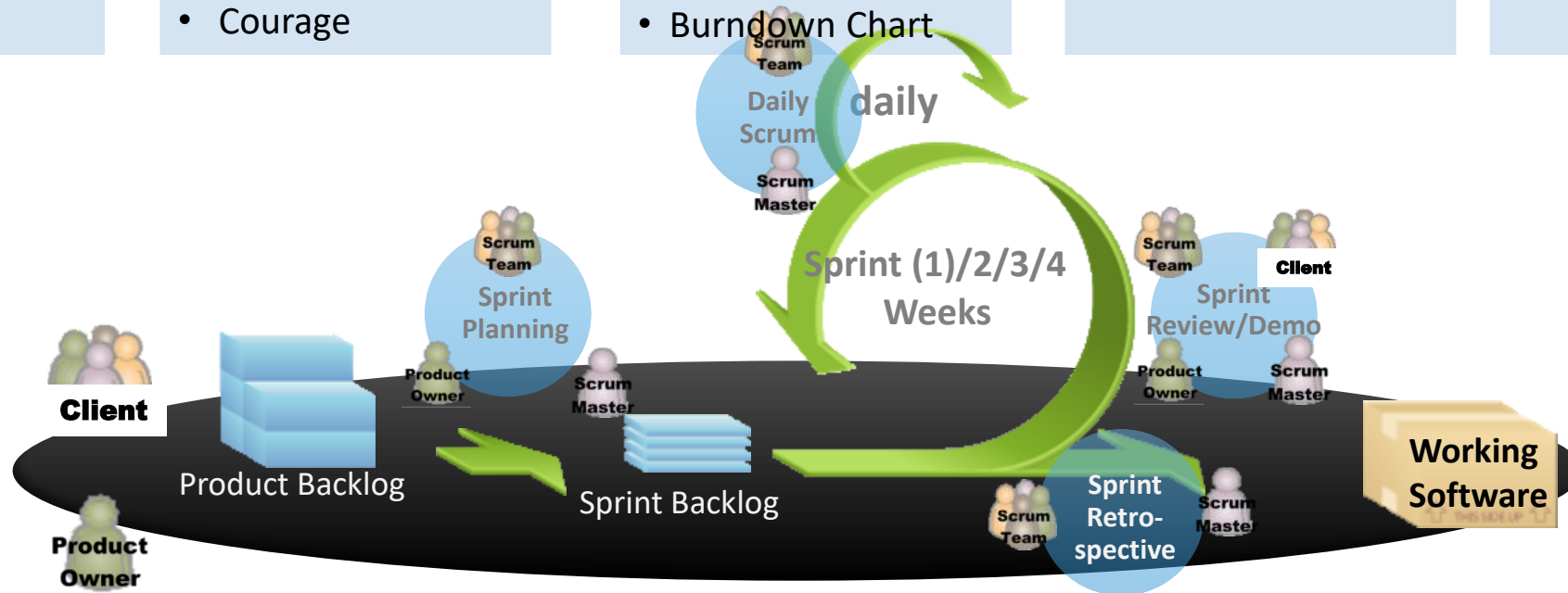
- Der **Product Owner** ist verantwortlich für die Erstellung und Pflege einer priorisierten ToDo Liste mit offenen Features. (**Product Backlog**)
- Vor jedem Durchlauf (**Sprint**) entscheidet das Team wie viele Features es aus der ToDo Liste abarbeiten kann, definiert diese genauer (Tasks) und schreibt sie in das **Sprint Backlog**.
- Während des Sprints stimmt sich das Team ab (**Daily Scrum Meeting**) und protokolliert den Fortschritt im **Burn Down Chart**.
- Der **Scrum Master** coachet das Team und sorgt für ein effizientes und ungestörtes Arbeitsumfeld.
- Das Ergebnis des Sprints wird im **Sprint Review meeting** präsentiert und entspricht einem “potentially shippable product increment”.
- Nach dem Sprint reflektiert das Team die Arbeit und sucht nach Verbesserungsmöglichkeiten (**Retrospektive**).
- Scrum basiert auf der Idee von: transparency, inspection, and adaptation

SCRUM Guide

Das Scrum Guide ist die offizielle Definition (ca. 15 Seiten) von Scrum und unterliegt regelmäßigen Updates.

Das Dokument kann z.B. hier heruntergeladen werden:
<https://www.scrum.org/resources/scrum-guide>

1 Goal	5 Values	6 Artifacts	3 Roles	4 Meetings
• Working Software	• Commitment • Focus • Openness • Respect • Courage	• Product Backlog • Sprint Backlog • Definition of Done • Definition of Ready • Impediment Backlog • Burndown Chart	• [Client] • Product Owner • Scrum Master • Scrum Team • [Consultants/Experts]	• Sprint Planning Meeting • Daily Scrum • Sprint Review/Demo Meeting • Sprint Retrospektive



Product Owner (PO)

Ist die Repräsentanz des Kunden und vertritt deren Interessen!

Ist verantwortlich für die Definition der Anforderungen und entscheidet daher was geliefert wird.

Product Owner - Aufgaben

Definiert Produkt Features befüllt/verantwortet das Product Backlog

Führt Gespräche mit Stakeholdern (Kunden, User,...)

Priorisiert (nach businessvalue) die Einträge im Backlog

Führt die Wünsche der Stakeholder zusammen und vertritt diese

Legt Release Daten und Umfang des Produkts fest

Akzeptiert Ergebnisse oder weist diese zurück (im Review)

Entscheidet **was** geliefert wird

Ist kein Projektmanager im herkömmlichen Sinn!

Team

Setzt die Aufgaben aus dem Backlog um

Folgt der Priorisierung des POs

Schätzt den Aufwand

Ist **selbstorganisierend** (multifunktionale Teammitglieder!)

Größe zwischen 5-9 Personen (osmotische Kommunikation)

Vollzeitmitglieder bevorzugt!

Commitet sich zu den **eigenen** Schätzungen!

Entscheidet wie viel je Durchlauf (Sprint) geliefert wird

Scrum Master (SM)

Nimmt eher eine Mediator und Coach Rolle ein

Aufgaben:

Beseitigt Hindernisse damit das Team produktiv arbeiten kann.

Unterstützt die Zusammenarbeit zwischen Team und PO

Schützt das Team vor externen Störungen

Stellt sicher, dass die SCRUM Regeln und Werte eingehalten werden

The roles according to Mitch Lacey

If the project were a car ...

- the **Scrum team** would be its **engine**.
- the **ScrumMaster** would be the **oil** greasing the engine,
 - providing protection against wear and increasing its power,
 - supporting high performance
 - so that the engine can last longer.
- the **Product Owner** is the **fuel** and the **navigation system**,
 - telling everyone where to go,
 - providing enough fuel for the tank.

Verantwortlichkeiten

Verantwortung	Product Owner	Team	ScrumMaster
Umfang	ja (Release)	ja (Sprint)	-
Zeit	ja (Release)	ja (Sprint)	-
Kosten	ja	-	-
Kommunikation	ja	ja (Sprint)	-
Risiko	ja	ja	-
Qualität	ja (Umfang)	ja (Testing)	ja (Prozess)

Meeting Inhalte

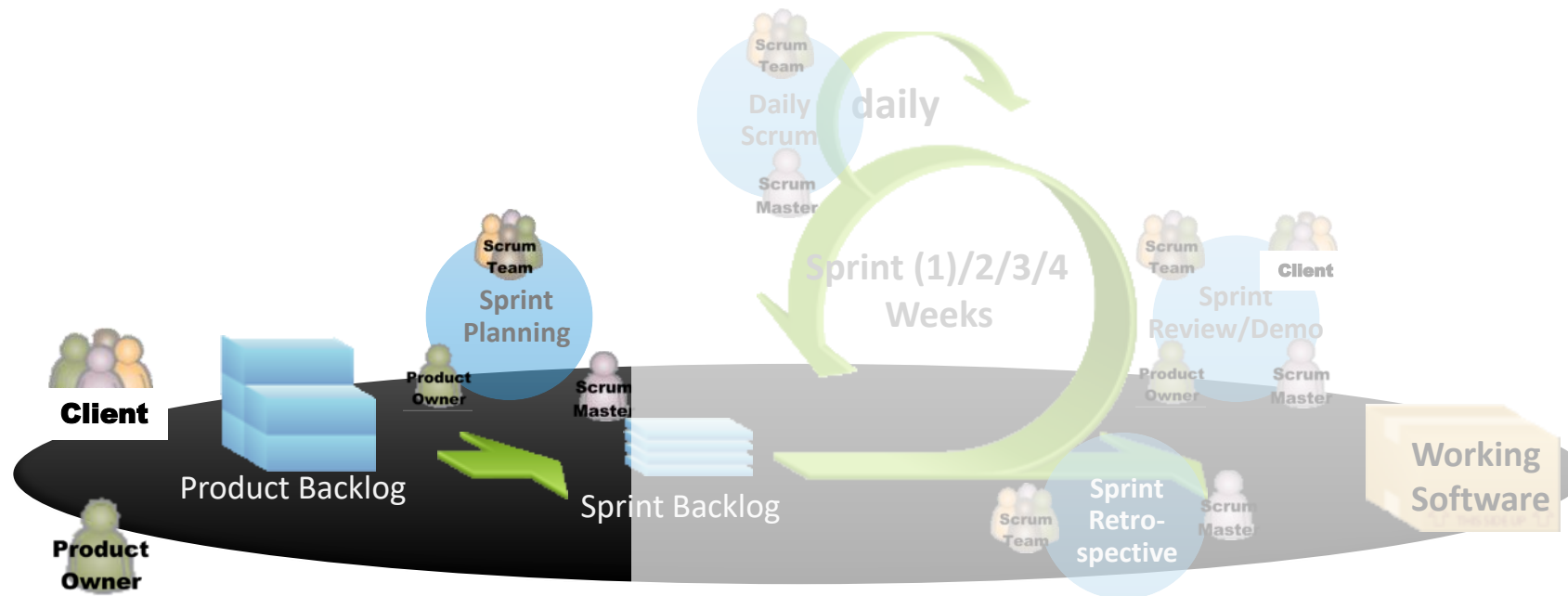
Sprint Planning

Sprint mit Daily Scrums

Review

Retrospektive

Sprint Planning



Sprint Planning

Vor dem Meeting schätzt das Team die Größe der Features

Der Product Owner priorisiert Features aus dem Product Backlog und/oder fügt Neue hinzu/streicht Alte

Der Product Owner beschreibt höher priorisierte Features detaillierter (sofern notwendig/möglich) damit diese der DOR (Definition of Ready) entsprechen

Sprint Planning

Während dem Meeting bestimmt das Team die verfügbare Kapazität (anhand Velocity) und der PO bestimmt die Dauer des Sprints (1-4 Wochen)

PO und Team besprechen die Features aus dem Product Backlog durch

Das Team wählt die am höchsten priorisierten Features und definiert für diese Tasks und schätzt den aufwand der Tasks.

Das wird so lange durchgeführt bis die Kapazität des Sprints erschöpft ist.

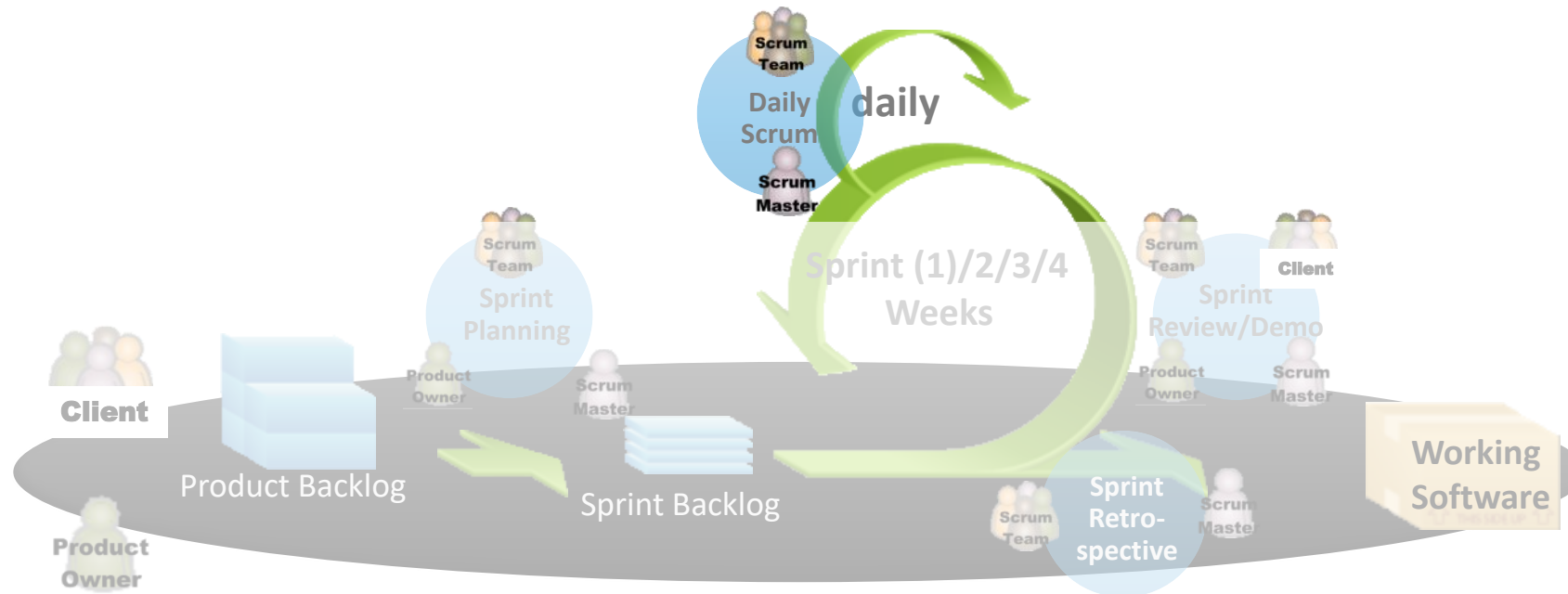
Sprint Planning ... und danach

Nach dem Meeting wird das Sprint Backlog vom Scrum Master erstellt

Das Team geht an die Arbeit und setzt die Aufgaben um (Sprint)

Bis zum nächsten Sprint Planning Meeting können jetzt keine Änderungen mehr an das Team übermittelt werden.

Daily Scrum Meeting



Daily Scrum Meeting

Aufbau (Empfehlung) ist 3-phasig:

- Was ist seit dem letzten mal erledigt worden?

- Was plane ich bis zum nächsten mal?

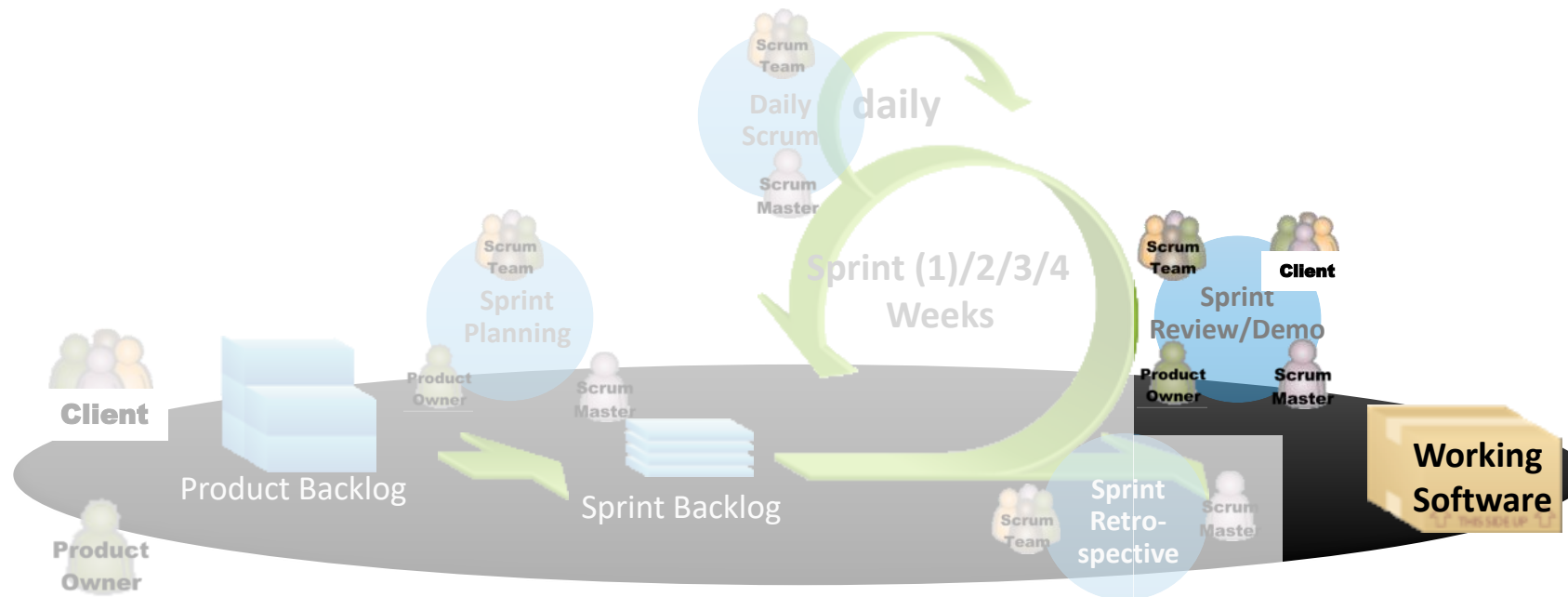
- Welche Probleme habe ich um das Ziel zu erreichen?

Meeting ist extrem wichtig für Kommunikationsaustausch zwischen dem Team. Fokus liegt immer auf dem Ziel (Goal) des Sprints

Soll nicht länger als 15 Minuten dauern (Timeboxed)

Scrum Master kümmert sich um die Behebung der Probleme (Impediment Backlog)

Sprint Review



Sprint Review

Team präsentiert das **Product Increment** dem Product Owner (und dem Kunden)

Vergleichbar mit einem User Acceptance Test

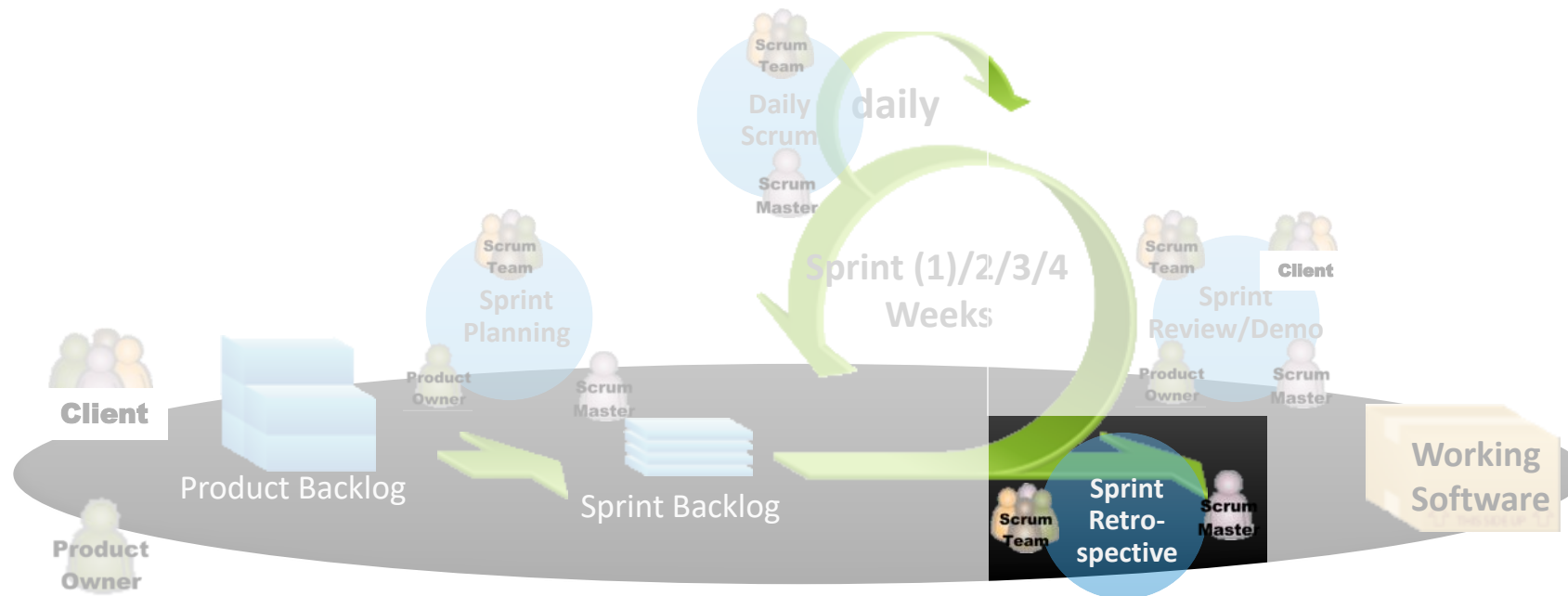
Nur fertige Features (inkl. Testing) werden vorgezeigt (Done is Done)

Vor dem Meeting berechnet der SCRUM Master die **Velocity** des Teams

Dear Feature Zuwachs wird anhand der Software gezeigt (keine Slides)

Der PO kann einzelne Stories abnehmen oder annehmen

Sprint Retrospective



Retrospektive

Scrum Team versucht besser zu werden und hinterfragt und reflektiert die eigene Arbeit

Teilnehmer sind: PO, SM, Team und andere sofern extra eingeladen

SM übernimmt Aufgaben in das Impediment Backlog

Keine Leistungsbeurteilung und Schuldzuweisungen!

Dokumente in SCRUM

6 Artifacts

- Product Backlog
- Sprint Backlog
- Definition of Done
- Definition of Ready
- Impediment Backlog
- Burndown Chart

Dokumente in SCRUM

Product Backlog

User Story 1
User Story 2
User Story ...
User Story ...
User Story ...
User Story ...
Epic

Sprint Backlog

User Story 1	Task 1
	Task 2
	Task ...
User Story 2	Task 1
	Task 2

Product Backlog

Beinhaltet **priorisierte Auflistung** der Anforderungen in Form von User Stories oder Use Case Definitionen (idealerweise anhand der INVEST Kriterien formuliert)
Weiters können spikes, bugs gelistet werden.

PO ist Eigentümer des Product Backlogs und dieser priorisiert und befüllt es.
Das Team kann durch Input unterstützen und ebenfalls Einträge hinzufügen.

Kann zu jeder Zeit ergänzt/abgeändert werden

Team schätzt den Aufwand in Relation zu einer Referenzstory. (z.B: Planning Poker)
Die Schätzungen sind zu diesem Zeitpunkt aber nur grobe Richtwerte

Product Backlog

INVEST Kriterien:

Letter	Meaning	Description
I	Independent	The user story should be self-contained, in a way that there is no inherent dependency on another user story.
N	Negotiable	User stories, up until they are part of an iteration, can always be changed and rewritten.
V	Valuable	A user story must deliver value to the end user.
E	Estimable	You must always be able to estimate the size of a user story.
S	Small	User stories should not be so big as to become impossible to plan/task/prioritize with a certain level of certainty.
T	Testable	The user story or its related description must provide the necessary information to make test development possible.

Quelle: [https://en.wikipedia.org/wiki/INVEST_\(mnemonic\)](https://en.wikipedia.org/wiki/INVEST_(mnemonic))

Sprint Backlog

Gehört dem Team und nur das Team darf es befüllen/ändern

Enthält Tasks (im Gegensatz zu Product Backlog) die in einem Sprint umgesetzt werden

Einzelne Entwickler können Tasks auswählen und diese umsetzen (selbststeuernde Teams)

Erlaubt das Ablesen des Fortschritts im Sprint

DoD (Definition of Done)

Beschreibt eine Checkliste die erfüllt sein muss damit es als „fertig“ bezeichnet werden kann.

Oft ausreichend wenn diese auf Projektebene definiert werden (nicht auf User Story Ebene vgl. Acceptance Criteria)

A handwritten table titled 'Definition of done' with two main columns: 'Artefakt' and 'Kriterien'. The table is divided into three rows of artifacts: 'Dialoge', 'Business - Services', and 'Iteration'. Each row lists specific criteria for completion, such as 'Quellcode dokumentiert', 'Performance-Kriterium erfüllt', and 'Inkrement auf Test Stage deployed und lauffähig'.

Artefakt	Kriterien
Dialoge	Quellcode dokumentiert, 100% Unittests erfolgreich, Usability Test erfolgreich, ...
Business - Services	Performance-Kriterium erfüllt, Sicherheitsanforderungen erfüllt, 100% Unittests erfolgreich, ...
Iteration	Inkrement auf Test Stage deployed und lauffähig, Acceptance Test abgeschl., alle neuen Funktionen dokumentiert, ...

Team "Done" List

...With a Story

- All Code (Test and Mainline) Checked in
- All Unit Tests Passing
- All Acceptance Tests Identified, Written & Passing
- Help File Auto Generated
- Functional Tests Passing

...With a Sprint

- All Story Criteria, Plus...
- Product Backup Updated
 - Performance Testing
 - Package, Class & Architecture Diagrams Updated
 - All Bugs Closed or Postponed
 - Code Coverage for all Unit Tests at 80% +

...Release to INT

- All Sprint Criteria, Plus...
- Installation Packages Created
 - MOM Packages Created
 - Operations Guide Updated
 - Troubleshooting Guides Updated
 - Disaster Recovery Plan Updated
 - All Test Suites Passing

...Release to Prod

- All INT Criteria, Plus...
- Stress Testing
 - Performance Tuning
 - Network Diagram Updated
 - Security Pass Validated
 - Threat Modeling Pass Validated
 - Disaster Recovery Plan Tested

DoD (Definition of Done)

Wie könnte ihr DOD aussehen?



DoR (Definition of Ready)

Beschreibt Kriterien die eine Anforderung (aus dem Product Backlog) erfüllen muss um in das Sprint Planing übernommen zu werden

DoR (Definition of Ready)

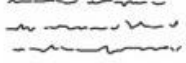
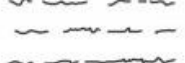
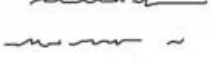
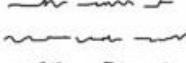
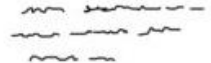
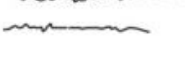
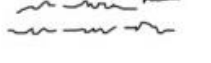
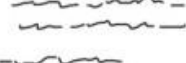
Wie könnte ihr DoR aussehen?



Impediment Backlog

Entspricht der ToDo Liste des Teams um nicht technische Themen abzubilden.

Wird gewartet und gepflegt vom Scrum Master. Und bezieht sich immer auf Themen die ein effizientes Arbeiten stören

EIGNER	PROBLEM	WORKAROUND	LÖSUNG	ZIELDATUM	STATUS
					FERTIG
					IN ARBEIT

Reporting

Burn Down Chart:

Visualisiert die Arbeitsschritte und die Velocity des Teams über die Sprints hinweg

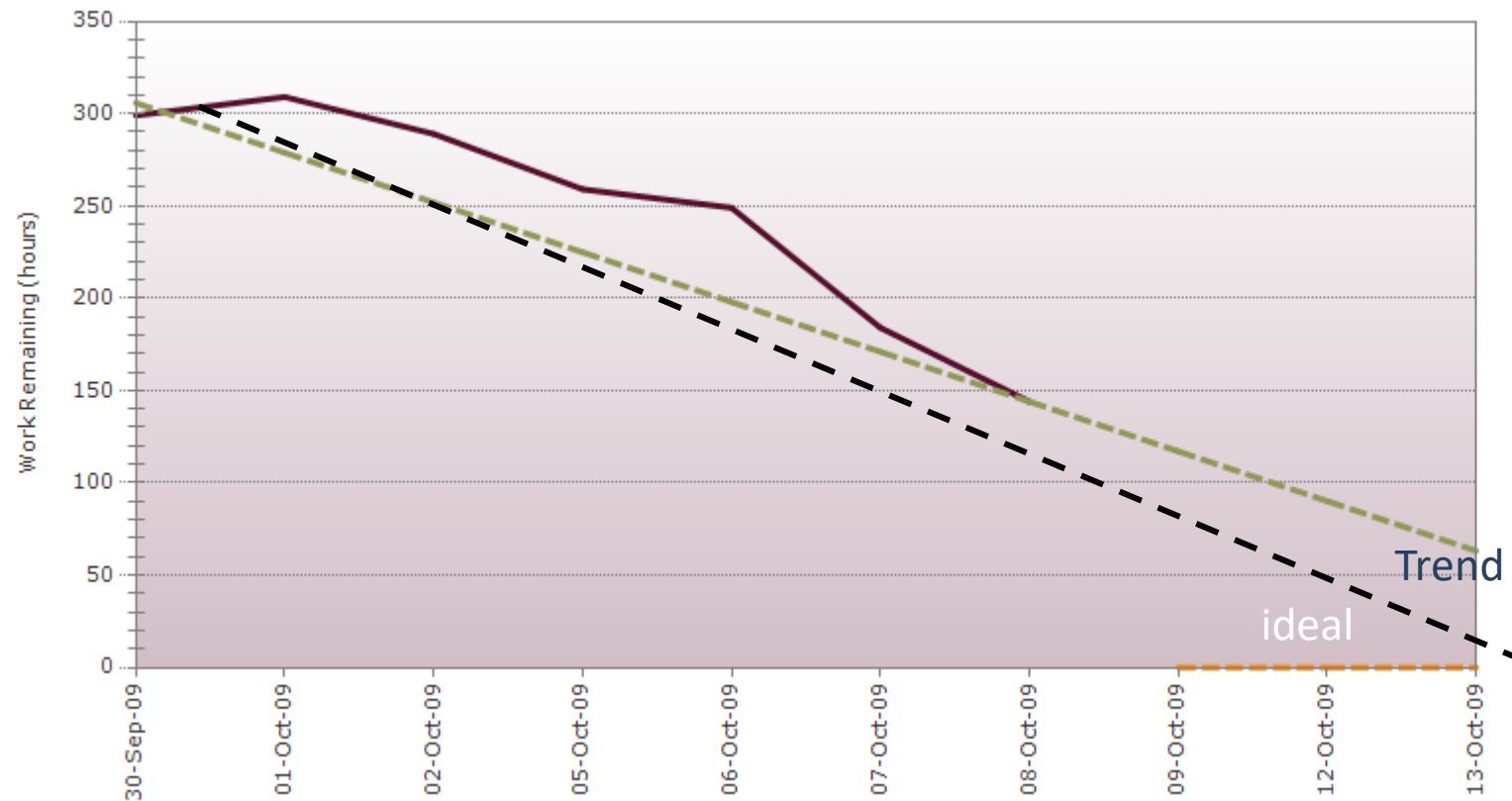
Sprint Task board:

Gibt einen schnellen und intuitiven Überblick über den Projektfortschritt

Product Cumulative Flow

Stellt die Stadien der Arbeiten in einer Aggregierten Form dar

Burn-down instead of burnout



Sprint Task board

Sprint Task Board

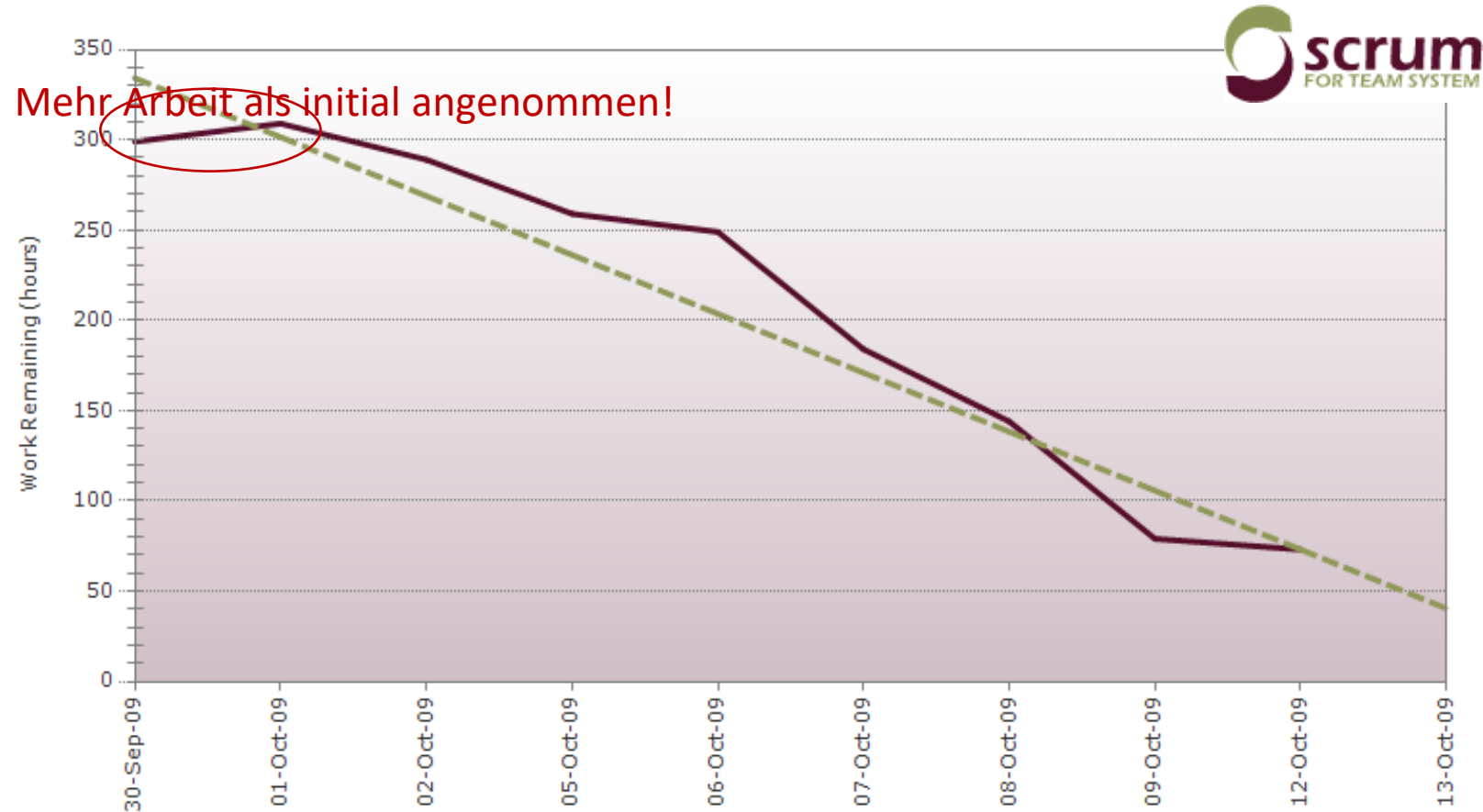


A typical project Release 1 Sprint 1

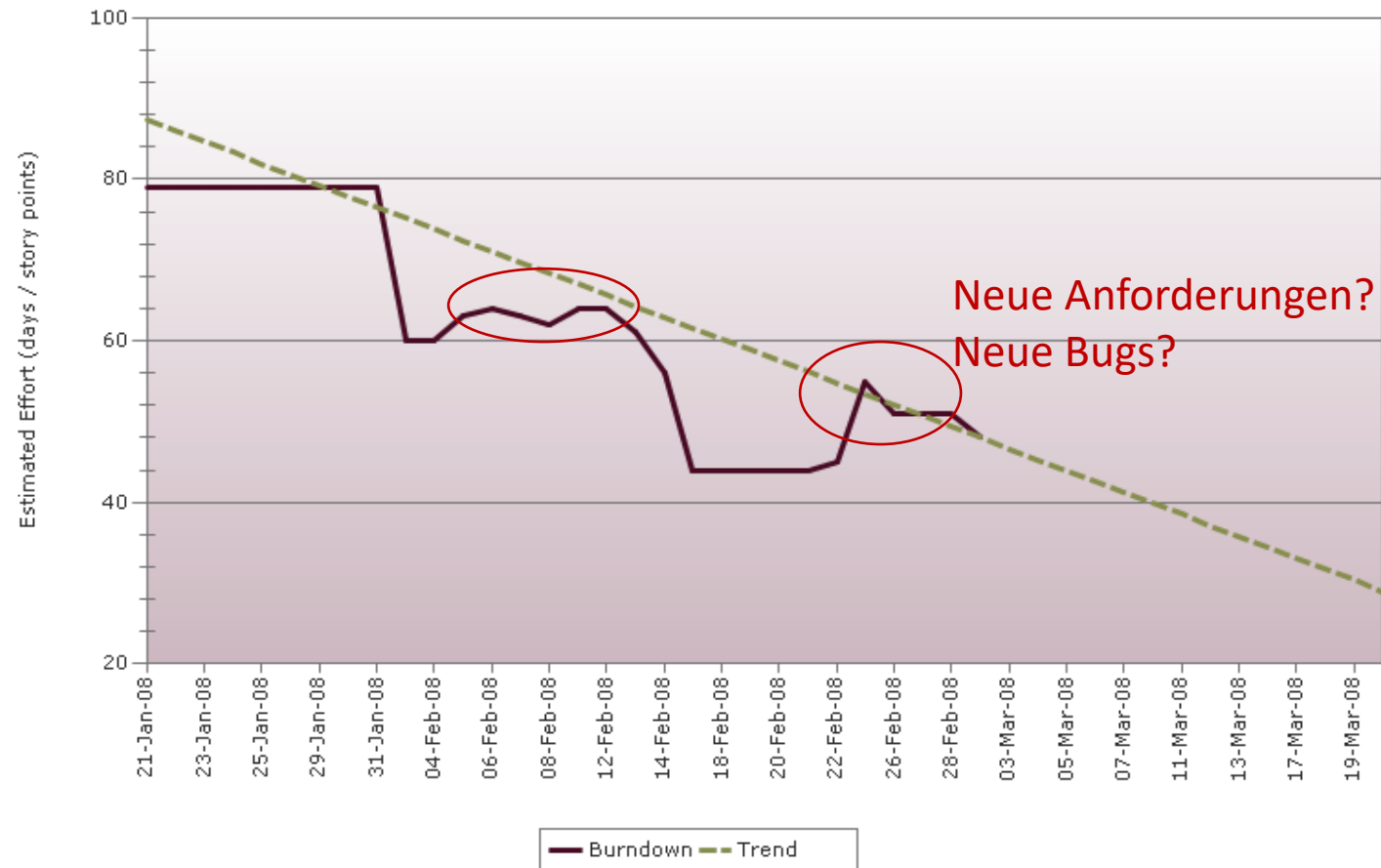
Last Data Refresh: 01-Oct-2009 15:16, Report Run: 01-Oct-2009

User Stories	Not Done	In Progress	Ready For Test	Done
 PB #110: User Story 1 Team: Type: Product Backlog Item	SB #142: Related to work item 110 - User Story 1 Team: Work \ Estimated: 60 \ 60	SB #140: Related to work item 110 - User Story 1 Team: Work \ Estimated: 20 \ 60 SB #141: Related to work item 110 - User Story 1 Team: Work \ Estimated: 10 \ 40		
 PB #112: User Story 3 Team: Type: Product Backlog Item	SB #132: Related to work item 112 - User Story 3 Team: Work \ Estimated: 1 \ 1 SB #135: Related to work item 112 - User Story 3 Team: Work \ Estimated: 16 \ 16 SB #136: Related to work item 112 - User Story 3 Team: Work \ Estimated: 12 \ 12			

Burn-down instead of burnout



Motivation for the team: things are getting done and everyone can see it (also the client)!

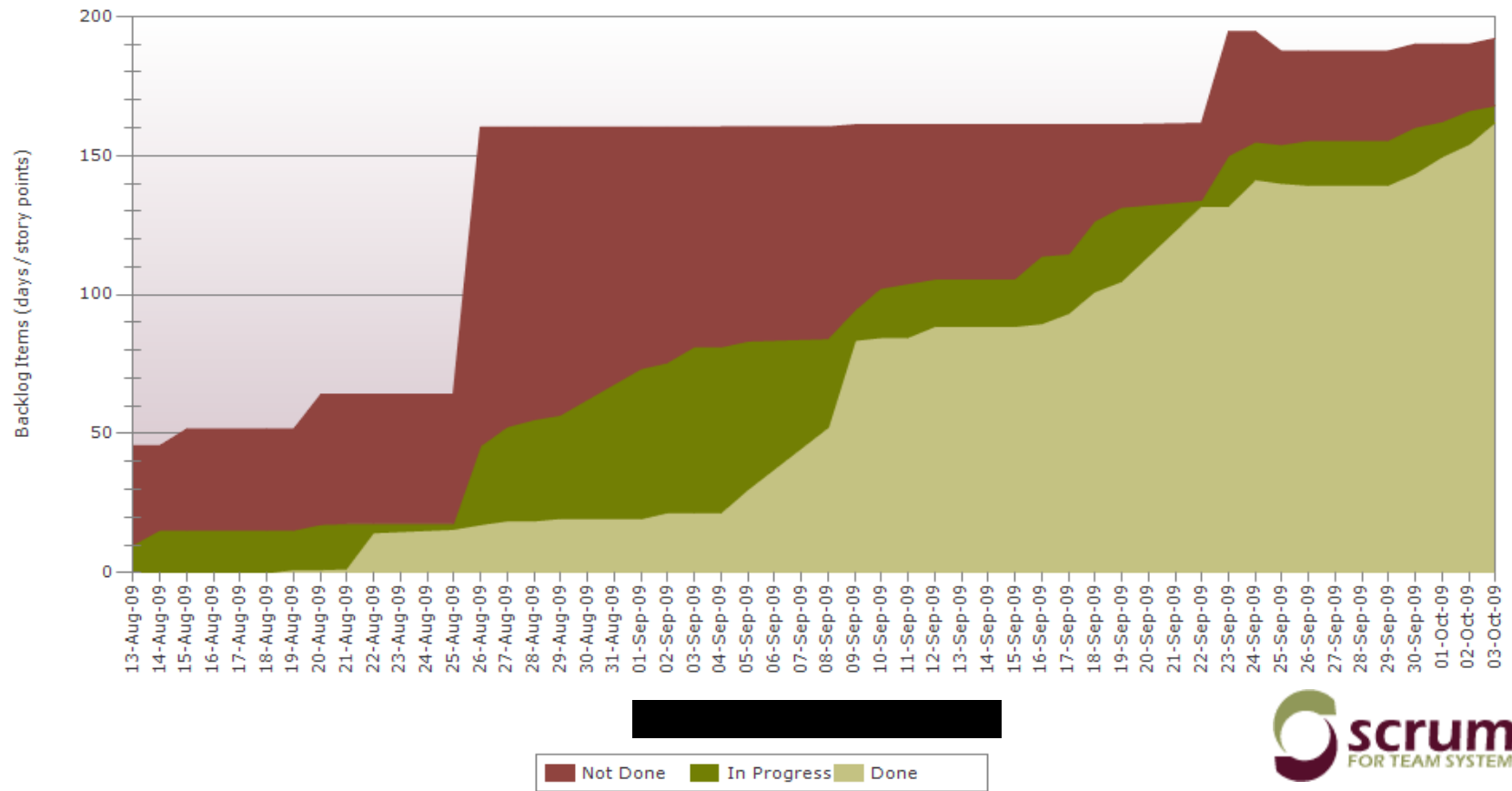


Soll/ Ist Vergleich

Prognosen möglich

Alles auf Basis der Velocity zu berechnen

Product Cumulative Flow



Argumente für und gegen SCRUM

Vorteile

- Transparent und/durch viel Kommunikation
- kurze Feedbackzyklen
 - Keine bösen Überraschungen am Ende des Projektes
 - Funktionierende Software zu Testzwecken, Review oder sogar Produktion nach jedem Sprint
- Inhärentes Veränderungsmanagement
- Time boxing Modell
- Selbst organisierendes, hoch produktives Team
- Hoher Zufriedenheitsfaktor für alle Beteiligten
- Fail-early Prinzip
- Fertig ist fertig Prinzip

Vorteile

- Business-value-driven Entwicklung
- Risikominimierung (Kosten eines Sprints)
- Feature-driven development & test-driven development möglich
- Vordefinierte Möglichkeiten Anforderungen vor jedem Sprint anzupassen. $\text{business value} = f(\text{cost, time, functionality, quality})$

Vorteile

- Kontinuierliches Monitoring durch Management/Kunden möglich (Demos, Daily Scrums)
 - Burn Down Chart
 - Probleme können nicht unter den Teppich gekehrt werden => werden in kürzester Zeit sichtbar. (Grüne Ampel Syndrom)

Nachteile

- Oft mit starkem Widerstand bei Einführung zu rechnen
 - Management Rolle dreht sich um (Team ist verantwortlich)
 - Mehr als eine Person übernimmt die Rolle des Projektmanagers
- Wenn es schief geht, geht es richtig schief
- Team Mitglieder müssen immer Vollzeit an einem Projekt arbeiten (zumindest 80% der Zeit, besser 90%)

Nachteile

- Das Team **muss** sich zu 100% zum Projekt und eigenen Schätzungen „**commiten**“.
- Der Kunde muss regelmäßiges Feedback geben (und sich die Zeit dafür zu nehmen)
- Entwickler verspüren bei den ersten Sprints mehr Stress
- Hoch spezialisiertes Vorgehensmodell für Software Projekte (z.B: Hausbau??)
- Unglücklicherweise hat der Trend der Agilität viel ruiniert da falsch eingesetzt/umgesetzt.
- SCRUM muss im Unternehmen möglich sein (Support von Oben)

Fehler bei der Einführung von SCRUM

- Nur Teile oder etwas von SCRUM einführen geht nicht.
- Rollen zusammenlegen (PO und SM)
- Nach den ersten Sprints aufgeben, da Schätzungen nicht stimmen und alle gestresst sind
- Features von Sprint zu Sprint verschieben
- Versuch SCRUM in Unternehmen einzuführen dessen Kultur und Prinzipien es nicht zulassen

Einführung von SCRUM - Überlegungen

Nicht gleich in einem großen Projekt implementieren, lieber in einem kleinem unwichtigen Projekt das erste mal Implementieren

Zumindest PO und SM benötigen vorab ein Training!

Kunde wird oft zu wenig eingebunden

Wechsel im Team und Teilzeit MitarbeiterInnen sind nicht gut