

SCRUM erleben

Einführung und Praxis

Agenda

- Was ist Agilität? Was ist Scrum?
- Scrum: Rollen, Meetings und Artefakte
- Scrum erleben
- Spezialitäten
- Agile Schlüsselerlebnisse
- Workshop-Retrospektive

Die Wahrscheinlichkeiten sind gegen uns

Größe (\$)	Erfolgreich	Problematisch	Nicht geliefert
< 500.000	38%	44%	19%
< 3 Mio.	27%	52%	21%
< 6 Mio.	16%	55%	29%
< 10 Mio.	4%	57%	39%
> 10 Mio.	0%	66%	34%

Quelle - The Standish Group CHAOS'2000 Survey



Definierte Prozesse

- Festgelegter Kontrollfluss, Rollen und Entscheidungspunkte
- Definierte Abhängigkeiten zwischen einzelnen Tätigkeiten bzw. ihren Ergebnissen
- Kontrolliere, dass das, was getan wird, genau dem entspricht was geplant wurde.
- Anforderungsänderungen werden über ein »Änderungsmanagement« in den Prozess eingebracht.

Einige Einsichten

- Ziv's Unsicherheitsprinzip des Software Engineering:

„Unsicherheit ist im Software-Entwicklungsprozess und den Produkten inhärent und unvermeidlich.“ (Ziv, 1996)

- Humphrey's Prinzip der Anforderungsunsicherheit:

„In einem neuen Software System werden die Anforderungen solange nicht komplett bekannt sein, bis die Anwender damit arbeiten.“ (Humphrey, 1995)

- Wegner's Lemma:

„Es ist unmöglich, ein interaktives System komplett zu spezifizieren.“ (Wegner, 1995)

Definierter Prozess vs. Empirischer Prozess

“It is typical to adopt the defined (theoretical) modelling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.”

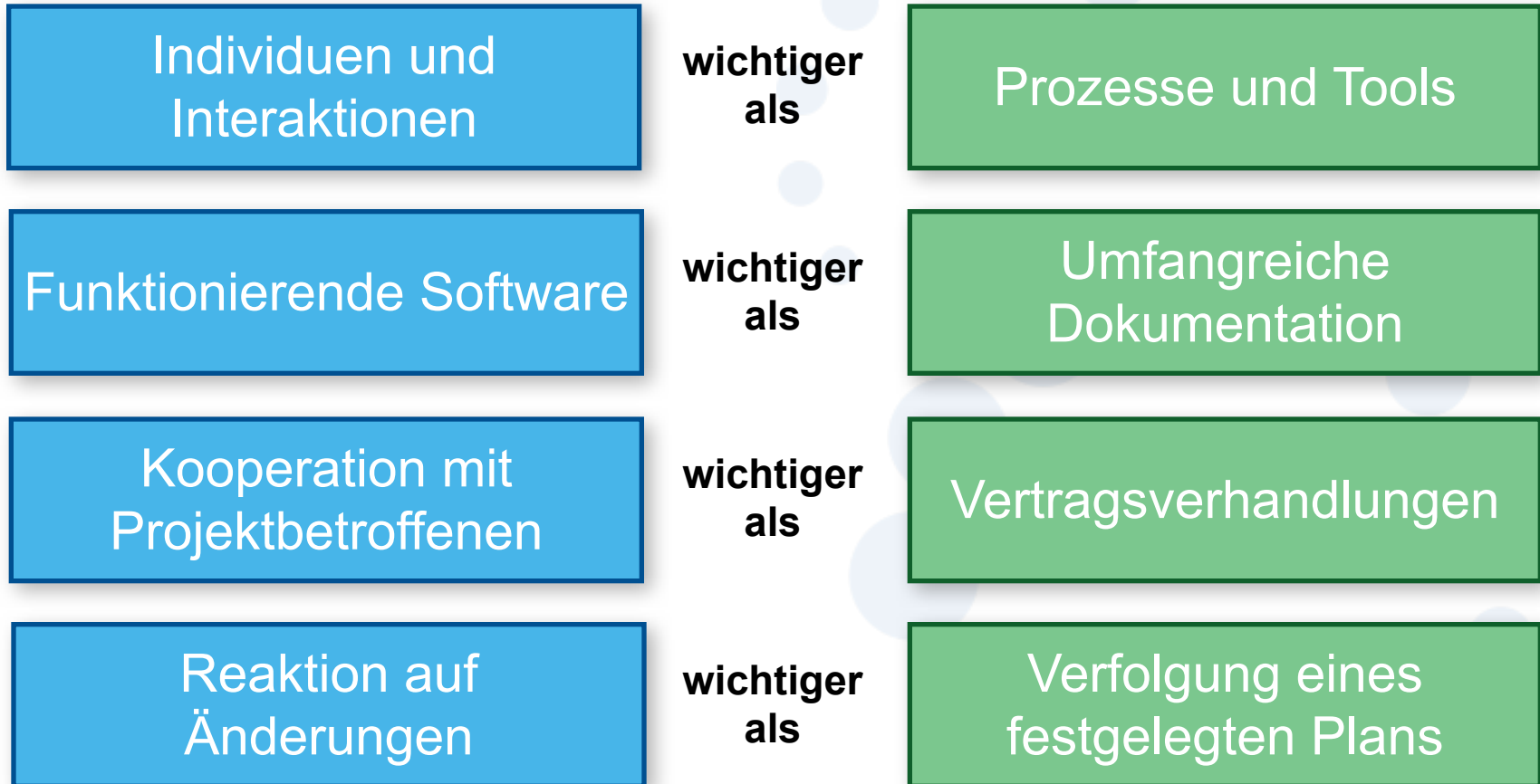
Process Dynamics, Modelling, and Control,
Ogunnaike and Ray, Oxford University Press, 1992



Empirische Prozesse

- Die einzelnen Aktivitäten sind nicht vorhersagbar, nicht-linear und zu komplex, um in genügender Detaillierung definiert werden zu können.
- Die einzelnen Tätigkeiten stehen lose nebeneinander und werden durchgeführt, wenn die Beteiligten es für sinnvoll halten.
- Kontrolle findet über Inspektion der Ergebnisse und Anpassung des Verhaltens statt.
- Befolgung einfacher Regeln führt zu »emergentem Verhalten« (emergent behaviour).

Agiles Manifest als Wertesystem



Quelle: www.agilemanifesto.org



Agile Methoden

- Agil formt eine Vision und unterstützt die Projektbeteiligten, das bestmögliche zu unternehmen, um diese zu erreichen.
- Agil ist „Die Kunst des Machbaren.“
- Typische Vertreter: XP, Crystal, Scrum, FDD

Vorteile agiler Entwicklung

- **Transparenz**
 - ▶ Wo stehen wir? Wo sind unsere Probleme?
- **Flexibilität**
 - ▶ Anforderungen und Prioritäten können sich ändern
- **Ständiges Lernen durch schnelles Feedback**
 - ▶ Automatisierte Tests
 - ▶ Reviews
 - ▶ Retrospektiven
 - ▶ Systemeinsatz

Agile Praktiken

- Herausstellen und Verdeutlichen von:
 - ▶ Konkrete Anforderungen
 - ▶ Fähigkeiten des Teams
 - ▶ Auftretende Hindernisse
- Selbststeuerung und ständige Anpassung an Realität
- Inkrementelle Ergebnisse
- Zusammenarbeit & Kommunikation
- Realitätsnähe, nicht Artefakte

Warum Scrum?

- Leichtgewichtiger Management-Rahmen
- Zeigt schnell Resultate
- Einfach zu erlernen
- Guter Startpunkt für agiles Vorgehen

Gefahr: Vernachlässigung der technischen, entwicklungsnahe Aspekte

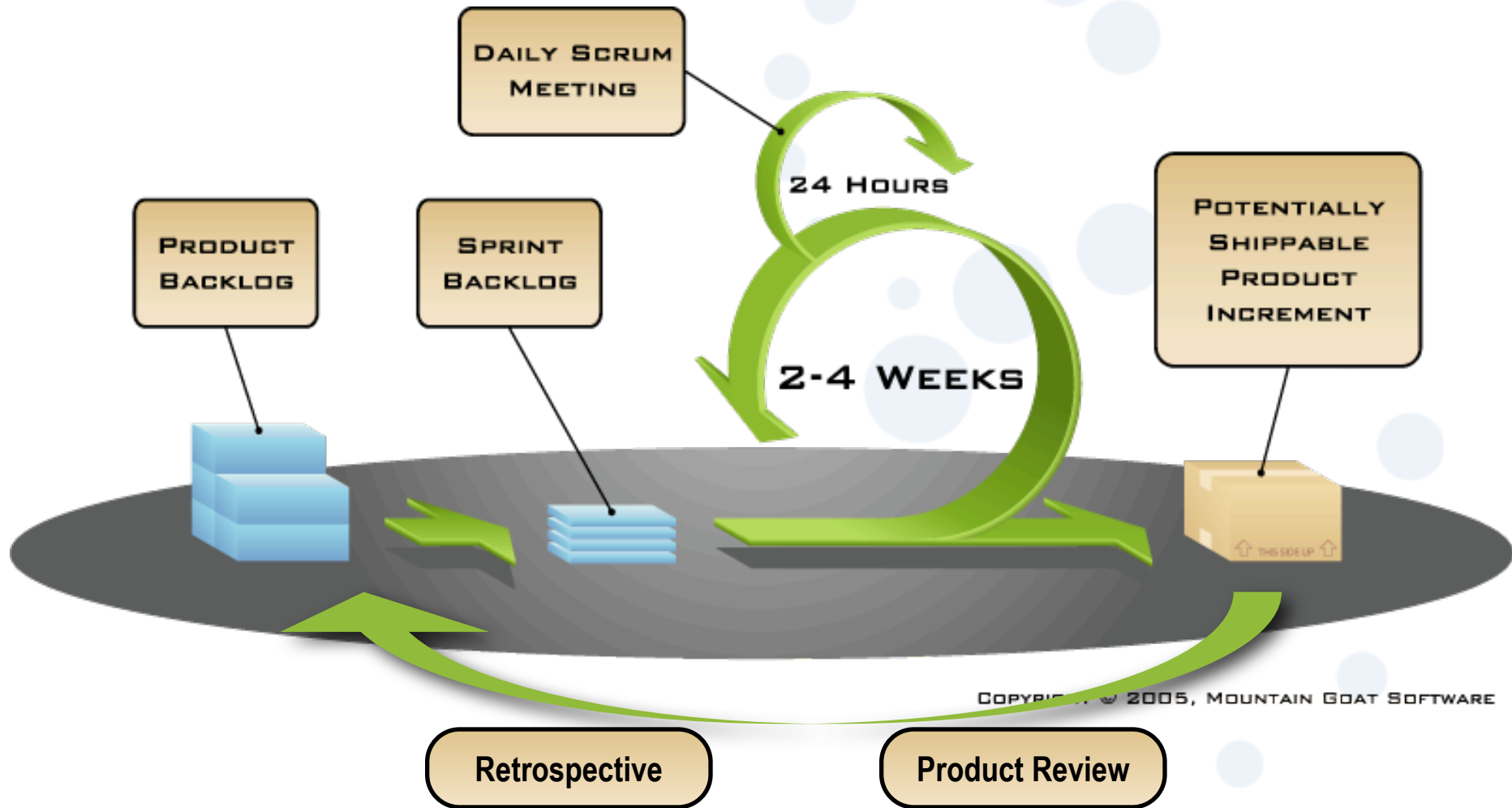
Scrum in 83 Worten

- Scrum ist ein agiler Prozessrahmen, der es erlaubt, sich auf die Auslieferung der wichtigsten Geschäfts-Anforderungen innerhalb kürzester Zeit zu fokussieren.
- Scrum gestattet es schnell und in regelmäßigen Abschnitten (von zwei Wochen bis zu einem Monat) tatsächlich lauffähige Software zu inspizieren.
- Das Business setzt die Prioritäten. Selbstgesteuerte Entwicklungsteams legen das beste Vorgehen zur Auslieferung der am höchsten priorisierten Features fest.
- Alle zwei Wochen bis zu einem Monat kann jeder lauffähige Software sehen und entscheiden, diese so auszuliefern oder in einem weiteren Abschnitt zu ergänzen.

Scrum Charakteristika

- Selbst-gesteuerte Teams
- Produkt schreitet in Serien von Sprints einer festen Dauer fort
- Anforderungen sind als Einträge im Product Backlog festgehalten
- Keine spezifische Entwicklungsmethode vorgeschrieben, stattdessen:
 - ▶ Generative Regeln um ein agiles Umfeld für die Auslieferung von Produkten zu schaffen
- Einer der agilen Prozesse

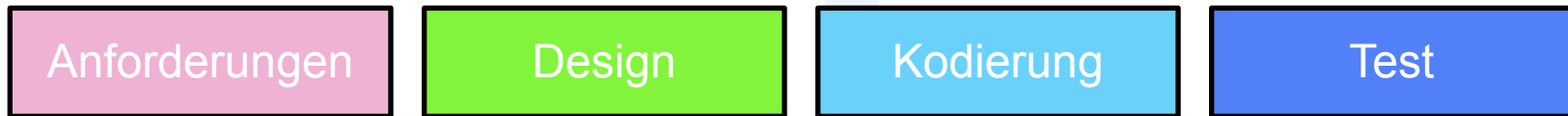
Scrum im Überblick



Iterationen in Scrum: Sprints

- Scrum-Projekte schreiten in Serien von Sprints voran
- Die typische Sprintdauer beträgt 2 – 4 Wochen
- Das Produkt wird während des Sprints entworfen, kodiert und getestet
- Am Ende steht ein auslieferbares Stück Software

Sequenzielle vs. überlappende Entwicklung



Anstatt alles im Ganzen hintereinander ...

... tun Scrum-Teams ein bisschen von allem die ganze Zeit über

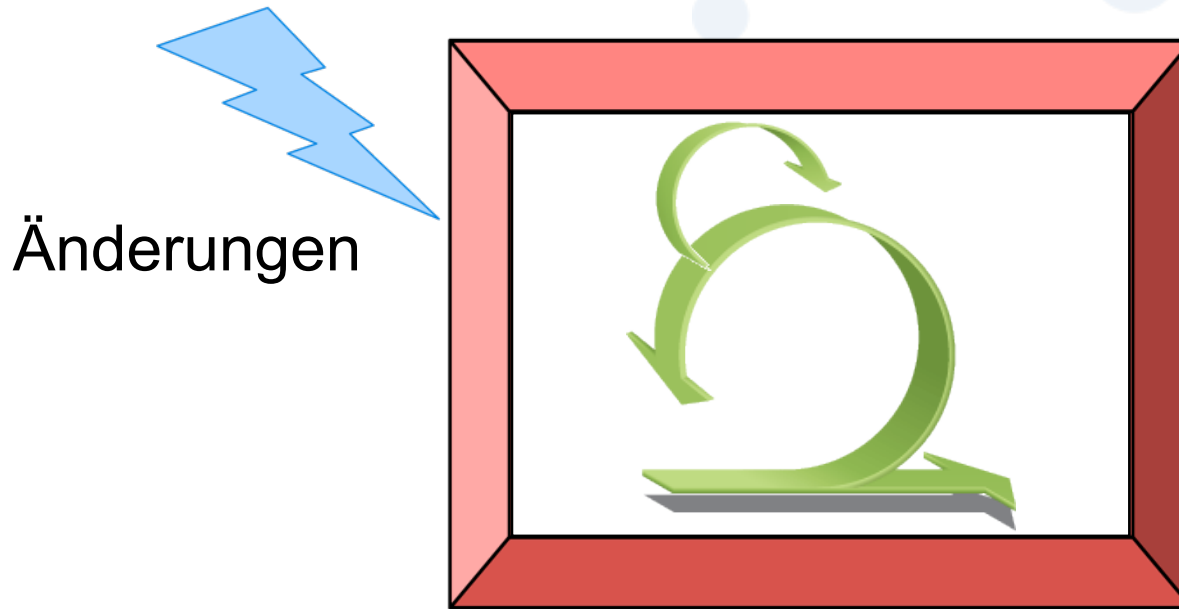
Quelle: "The New New Product Development Game" von Takeuchi und Nonaka. Harvard Business Review, January 1986.



Sprint als strenge Time-Box

- Eine konstante Dauer führt zu einem besseren Rhythmus:
 - ▶ Verlässliche Termine (mit variablem Funktionsumfang)
 - ▶ Vergleichbarkeit (zur Geschwindigkeitsmessung)
 - ▶ Priorisierung
(das Beste aus der verfügbaren Zeit rausholen)

Keine Anforderungsänderungen während des Sprints



Zu viele Änderungen führen zu geringerer Produktivität und machen jegliche Planung überflüssig.

Länge der Sprints

- **Bestimmung der Obergrenze:**
Wie lange können Sie geplanten Prioritäten stabil halten?
- **Bestimmung der Untergrenze:**
Wie lange benötigen Sie, um alles, was zur Fertigstellung eines Features nötig ist, umzusetzen?
- **Faustregel:**
Mit kürzeren Sprints beginnen und diese erst verlängern, wenn sich ein Rhythmus eingestellt hat.

Scrum - der Rahmen

Rollen

- Produkt-Owner
- ScrumMaster
- Team

Meetings

- Sprint-Planung
- Sprint-Review
- Sprint-Retrospektive
- Tägliches Scrum-Meeting

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Scrum - der Rahmen

Rollen

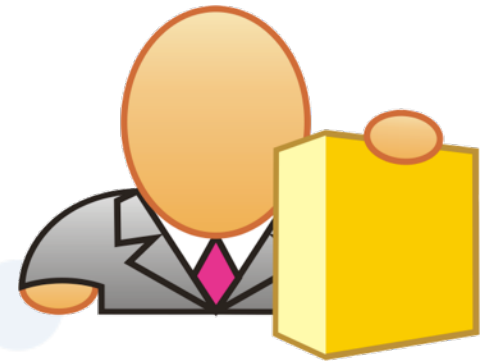
- Produkt-Owner
- ScrumMaster
- Team

- Sprint-Planung
- Sprint-Review
- Sprint-Retrospektive
- Tägliches Scrum-Meeting

Artefakte

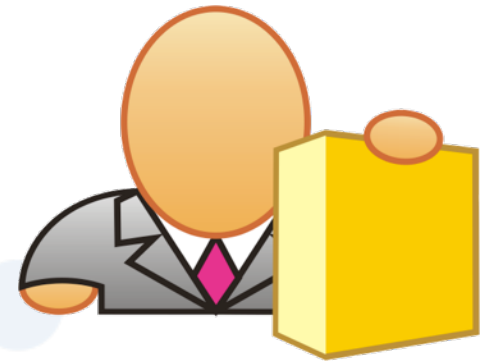
- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Der Product Owner (1)



- erfasst Bedürfnisse der Kunden und Stakeholder
- bestimmt Auslieferungsdatum und Inhalt
- ist verantwortlich für Produkt und Projekterfolg (u.a. ROI)
- definiert und priorisiert Features abhängig von deren *Geschäftswert*
- passt Features und Prioritäten für jeden Sprint an
- akzeptiert oder weist Arbeitsergebnisse zurück

Der Product Owner (2)



- darf das Team bis zu 10% seiner Zeit in Anspruch nehmen
 - ▶ für konzeptionellen Arbeiten und Schätzklausuren
 - ▶ *zusätzlich* Sprint Planning und Sprint Review
- muss mind. 80% seiner Zeit für das Projekt zur Verfügung stehen
- kann maximal zwei Projekte parallel betreuen
- ist *genau eine Person*
 - ▶ beeinflussbar durch Komitees, Management, Kunden, Vertrieb

Warum nur eine Person?

- Sicherstellung von Klarheit der Anforderungen gegenüber den Entwicklern
- Eliminierung von Verwirrung durch
 - ▶ mehrere Chefs
 - ▶ unterschiedliche Meinungen
 - ▶ Einmischung von Außen

Der Scrum Master (1)



- ist verantwortlich für die Einhaltung von Scrum-Werten und -Techniken
- ist ein Coach für die eingesetzten Techniken
- hilft beim Beseitigen von Hindernissen
- unterstützt die enge Zusammenarbeit zwischen allen Rollen und Funktionen
- schützt das Team vor äußeren Störungen

Der Scrum Master (2)



- bringt idealerweise Ingenieur- und Entwicklerfähigkeiten mit
- versucht Lernprozess und Selbstorganisation des Teams anzustoßen
- moderiert die diversen Meetings und hilft bei deren Vorbereitung
- repräsentiert das Team gegenüber dem Management

Der Scrum Master (3)



- darf **nicht** inhaltlich arbeiten
- ist **nicht** Teil des (Umsetzungs-)Teams
- hat **keine** Weisungsbefugnis gegenüber dem Team

Das Team

- Typischerweise 5-9 Personen
- Funktionsübergreifend:
 - ▶ QS, Programmierer, UI-Designer, etc.
- Mitglieder sollten Vollzeitmitglieder sein
 - ▶ Wenige Ausnahmen (z.B. Systemadministratoren)
- Das Team steuert sich selbst
 - ▶ Mitglieder übernehmen Aufgaben, bekommen sie nicht zugewiesen
 - ▶ Es darf alles tun, was für die Zielerreichung notwendig ist
 - ▶ Ideal: keine Titel (aber manchmal nicht vermeidbar)
- Mitgliedschaft kann sich nur zwischen Sprints verändern



Was macht ein Team aus?

- Eine Gruppe: 2 - 20 Personen mit organisatorischem Zusammenhalt
- Ein Team:
 - ▶ gemeinsames Ziel
 - ▶ selbstloses Handeln
- Ein Team zeigt Commitment
 - ▶ Jeder arbeitet 100%ig an der gemeinsamen Aufgabe
 - ▶ Jeder setzt die Teamentscheidungen um, auch wenn er persönlich anderer Meinung ist
 - ▶ Keine Sabotage, kein Schlechtreden, kein „Ich habe es ja vorher gewusst“

Selbst-organisiert, -gesteuert und -bestimmt

- Jede Gruppe organisiert sich - irgendwie - selbst
- **Selbstgesteuerte** Teams arbeiten zielgerichtet
- **Selbstbestimmte** Teams setzen sich ihre Ziele selbst
- Kosten der Selbststeuerung
 - ▶ **Fremdgesteuerte** Gruppen arbeiten oft **effizienter**
 - ▶ Die Entscheidungen selbstgesteuerter Teams haben eine **höhere Qualität**
 - ▶ Selbstgesteuerte Teams können **effektiver** sein

Scrum - der Rahmen

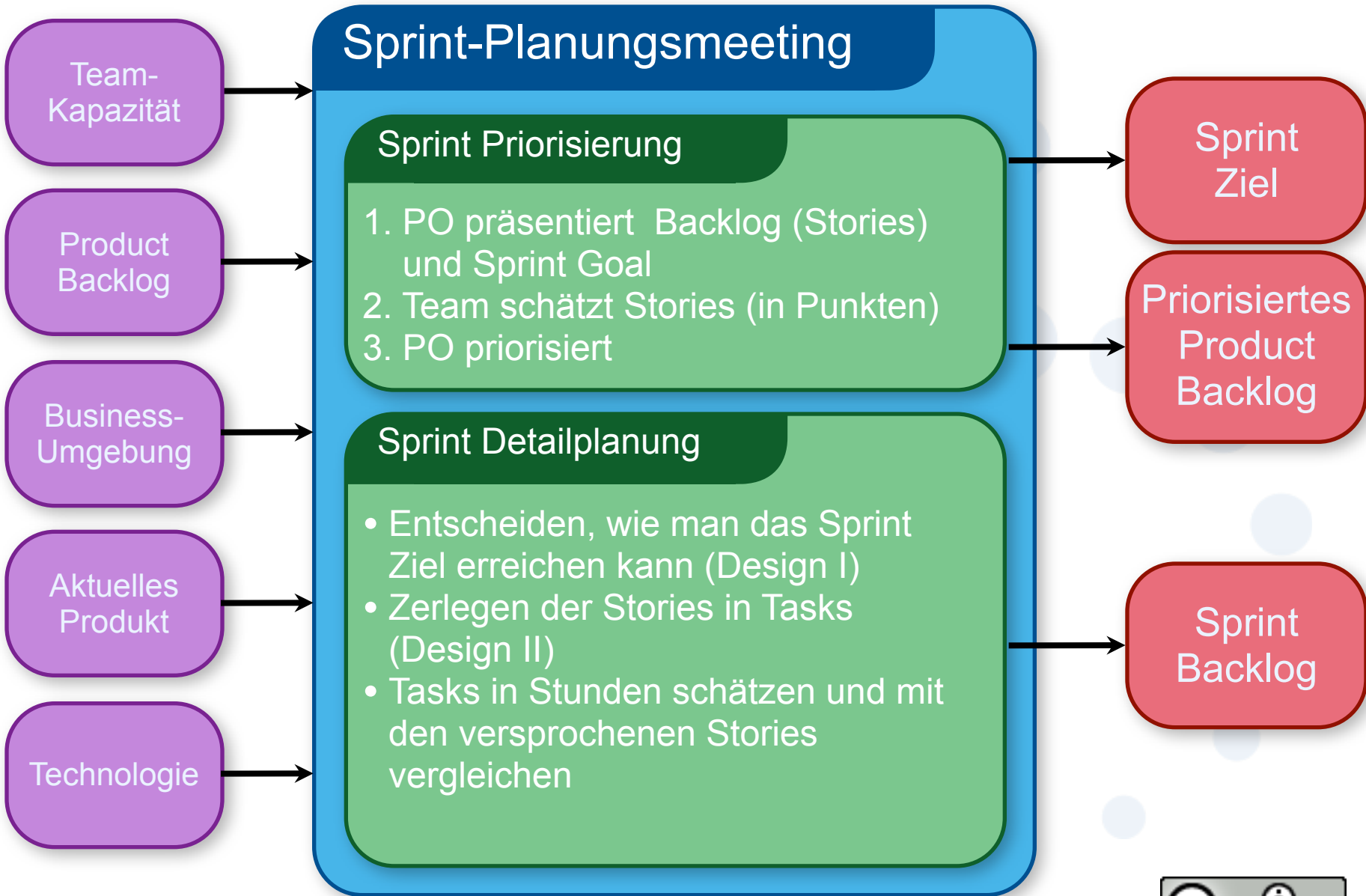
Rollen

- Produkt-Owner
- ScrumMaster
- Team

Meetings

- Sprint-Planung
- Sprint-Review
- Sprint-Retrospektive
- Tägliches Scrum-Meeting

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm



Sprint-Planungsmeeting

Sprint Priorisierung

1. PO präsentiert Backlog (Stories) und Sprint Goal
2. Team schätzt Stories (in Punkten)
3. PO priorisiert

Sprint Detailplanung

- Entscheiden, wie man das Sprint Ziel erreichen kann (Design I)
- Zerlegen der Stories in Tasks (Design II)
- Tasks in Stunden schätzen und mit den versprochenen Stories vergleichen

Sprint Ziel

Priorisiertes Product Backlog

Sprint Backlog

Die Sprint Detailplanung

- Team wählt Stories, zu deren Implementierung es sich verpflichten kann, aus dem Product Backlog aus.
 - ▶ Grundlage: Die fertiggestellte Arbeit des letzten Sprints
- Sprint Backlog wird erstellt
 - ▶ Tasks werden identifiziert und geschätzt (1-8 Stunden)
 - ▶ Dieses wird gemeinschaftlich getan, nicht vom ScrumMaster allein
- Highlevel-Design wird berücksichtigt

As a vacation planner,
I want to see photos
of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

Das tägliche Scrum-Meeting

- Parameter:
 - ▶ Täglich, selber Ort, gleiche Zeit
 - ▶ <15 Minuten
 - ▶ Stand-up
- Ziel: Abstimmung im Team, nicht Problemlösung
 - ▶ Alle sind eingeladen
 - ▶ Aber nur Team-Mitglieder und der Scrum-Master dürfen reden
- Die Hindernisliste wird aufgefüllt
- Hilft dabei, andere Meetings zu vermeiden



A chicken and a pig are walking down the road. The chicken asks the pig: “Would you like to open a restaurant with me?”

The pig considers the question and replies, “Yes, I’d like to. What do we call the restaurant?” The chicken answers: “Ham and Eggs!”

The pig stops, pauses, and says: “On second thought, I don’t think I want to open this restaurant with you. I’d be committed, but you’d only be involved.”

Jeder beantwortet 3 Fragen

1

Was hast du seit dem letzten mal geschafft?

2

Was wirst du bis zum nächsten mal tun?

3

Was behindert dich beim vorwärts kommen?

- Keine Statusberichte für den Scrum-Master, sondern Abstimmungsinformationen für die Kollegen
- Weiterführende Diskussionen werden im Anschluss geführt

Das Sprint-Reviewmeeting (1)

- Das Team präsentiert, was es während eines Sprints erreicht hat
- Typischerweise in Form einer Demo der neuen Features oder der zugrunde liegenden Architektur
- Informell
 - ▶ Grobe Regel:
Zwei Stunden zur Vorbereitung
 - ▶ Keine Folien, sondern **echte** Artefakte
- Das ganze Team nimmt teil
- Laden Sie die ganze Welt ein!



Das Sprint-Reviewmeeting (2)

- Product Owner und andere geben Feedback
- Darauf basierend kann der Kunde:
 - ▶ Neu priorisieren
 - ▶ Neue Anforderungen ins Product Backlog einfügen
 - ▶ Die Auslieferung des Produkts anstoßen
 - ▶ Die Qualitätsansprüche ändern
 - ▶ Das Projekt beenden, da weitere Iterationen keinen ROI bringen

Die Sprint-Retrospektive (1)

- Wichtigste Rückkopplungsschleife für den Entwicklungsprozess:
 - ▶ Prüfen Sie regelmäßig, was gut und nicht so gut funktioniert
- Typischerweise 30-60 Minuten lang
- Nach jedem Sprint
- Das ganze Team nimmt teil
 - ▶ Vorsicht bei externen Teilnehmern

Die Sprint-Retrospektive (2)

- Frage an jeden einzelnen:
 - ▶ Was lief gut während des Sprints?
 - ▶ Was kann im nächsten Sprint verbessert werden?
- Diskussion der identifizierten Probleme
- Identifikation von wenigen „Action Items“

Weitere Meetings

- Schätzklausuren
 - ▶ Zweck: Aufwände größerer Features schätzen
 - ▶ Kann eingeplant werden
 - ▶ Teilnehmer: Team und Product Owner
 - ▶ Methoden: Schätzpoker, UC-Points etc.
- Teambesprechungen
 - ▶ Abstimmung, Vereinbarungen, Diskussionen, Design, Architektur etc.
 - ▶ Nach Daily Scrum vereinbaren
 - ▶ Jeder kann sie einberufen
- Anforderungsworkshops

Scrum - der Rahmen

Rollen

- Produkt-Owner
- ScrumMaster
- Team

Meetings

- Sprint-Planung
- Sprint-Review
- Sprint-Retrospektive
- Tägliches Scrum-Meeting

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Der Product Backlog



Product Backlog

- Eine Liste aller gewünschten funktionalen und nicht-funktionalen Anforderungen
- Idealerweise soll jeder Eintrag wertvoll für Benutzer des Produktes oder Kunden sein
- Vom Produkt-Owner priorisiert
- Vom Team geschätzt
- Höher priorisierte Einträge sind detaillierter dargestellt
- Nur eine Liste bei mehreren Teams
- Jeder kann Einträge beisteuern
- Sichtbar für alle

Product Backlog: Beispiel

Anforderung	SP	Sprint#
Parallelen Zugriff auf selben Kunden verbieten	8	2
Funktion „Neuer Kunde“ auch in Intranet-GUI	3	2
User-Login gegen zentrales LDAP	5	2
Performance der Report-Queries unter 10 sec drücken	8	2
...		
Kontaktdatenimport aus CRM-System	13	3
Kunden können ihre Kontaktadresse selbst ändern	13	4
...		
Mehr Endgeräte unterstützen	?	?
Skalierbarkeit des DB-Zugriffs erhöhen	?	?

Das Sprint Ziel

- Kurze Angabe dessen, worauf sich die Arbeiten während des Sprints fokussieren

Database Application

Die Applikation soll auch auf dem MS SQL Server laufen, zusätzlich zu Oracle.

Life Sciences

Implementiere die Features, die für geschlechtsspezifische Studien benötigt werden

Online Banking

Unterstütze sofortige Transaktionsausführung statt nächtliche Batch-Verarbeitung

Sprint Backlog

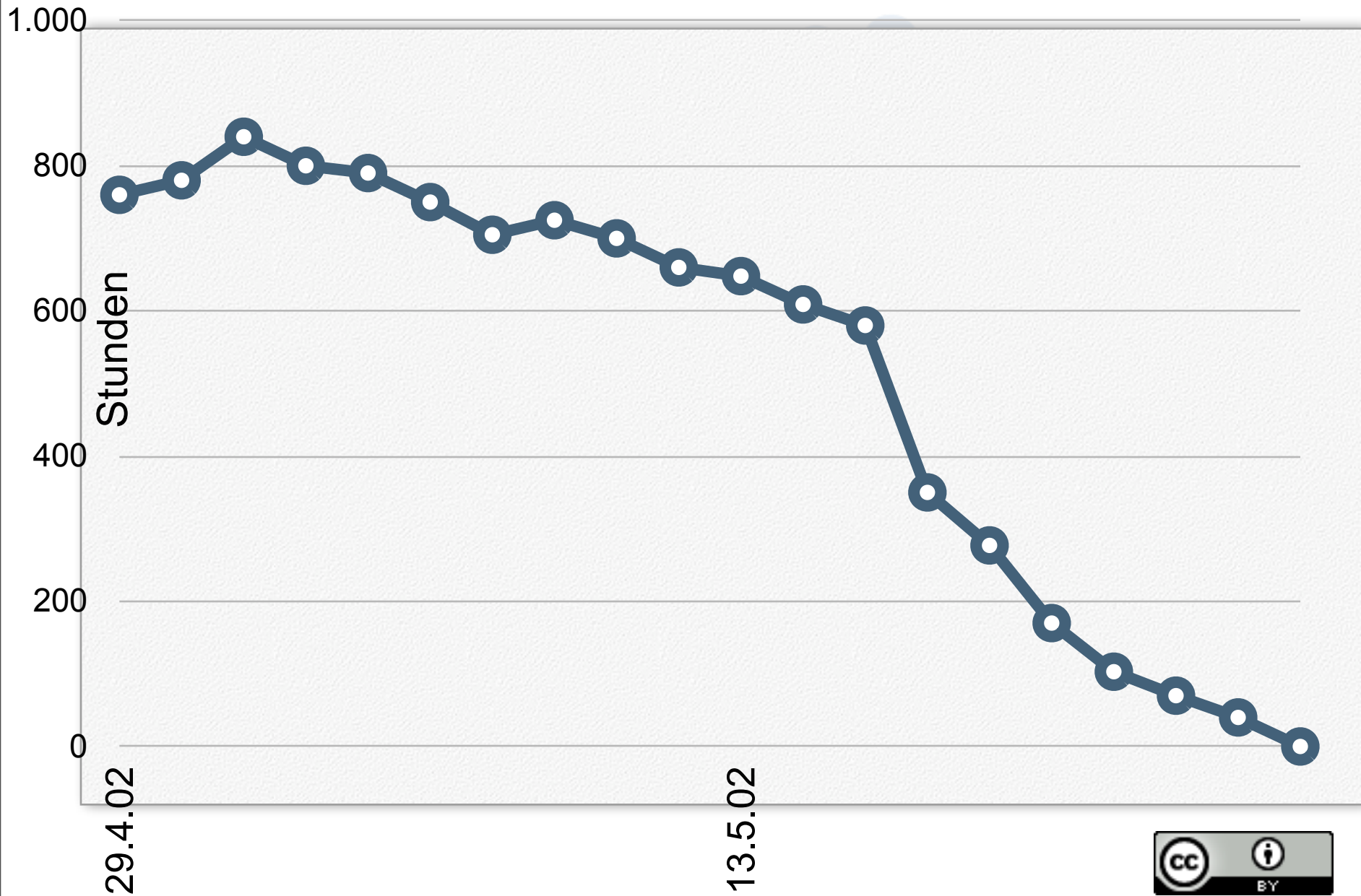
Liste aller im laufenden Sprint durchzuführenden Tasks und ihrer (Rest-)Schätzungen

Task	Status	Verant.	Restz. (h)
Schema für LDAP-Daten erweitern	i.A.	Marc	4
Test-Fixture User Login schreiben	fertig	Marc	0
GUI-Design User Login	i.A.	Luc	4
GUI-Design User Login umsetzen	offen		8
Schreibe Online-Help für Login	offen		2
...
Summe			122

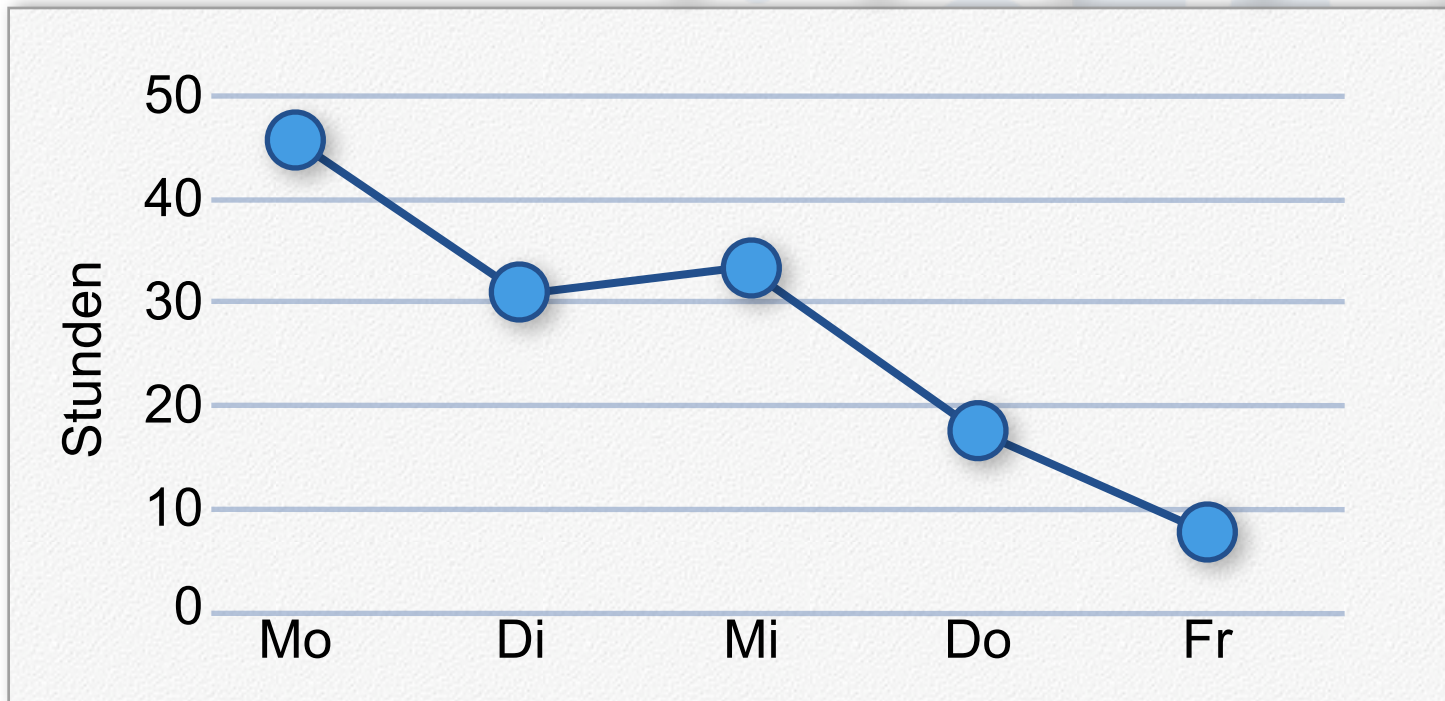
Management des Sprint Backlog

- Team-Mitglieder wählen Tasks aus, die Arbeit wird nie zugewiesen
- Der geschätzte Restaufwand wird täglich aktualisiert
 - ▶ Alternative 1: Nur fertige Tasks werden auf 0 gestellt
 - ▶ Alternative 2: Nur fertige Stories werden auf 0 gestellt
- Jedes Team-Mitglied kann Tasks hinzufügen, löschen oder ändern
- Überarbeitung des Sprint Backlogs sobald neue Informationen vorliegen
- An prominenter Stelle sichtbar machen
 - ▶ z.B. Whiteboard im Team-Raum

Das Sprint Burndown-Diagramm



Tasks	Mo	Di	Mi	Do	Fr
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Weitere Artefakte

- Produktvision
 - ▶ "Vertrag" zwischen Sponsor und Product Owner
 - ▶ Jeder Beteiligte sollte sie verstanden haben
 - ▶ Oberste Instanz bei Priorisierung
- Hindernisliste
 - ▶ Was hindert das Team am best möglichen Vorankommen?
 - ▶ Organisatorische und technische Hindernisse
 - ▶ Auch diese Liste muss priorisiert und abgearbeitet werden
- Release-Burndown

Typische Kinderkrankheiten

- Die ersten Planungen dauern laaaaaaaange
- Das Projekt hat noch kein Budget, das Personal ist vorläufig und die Ausrüstung unangemessen
- Das Team hat Schwierigkeiten, sich die Arbeit selbst einzuteilen
- Es existiert kein (auch nicht interner) Kunde
- Uneinigkeit bezüglich der einzusetzenden Technologie
- Abgeschlossene Arbeiten sind nicht wirklich **fertig** (done done)

Wer ist wofür verantwortlich? (1)

Aktivität	Verantwortlicher	Verantwortung
Durchsetzung der Vision	Product Owner	<ul style="list-style-type: none">▶ Erstellung, Unterhaltung und Kommunikation der Produktvision.▶ Gewährleistung der initialen und folgenden Budgetierung durch die Erstellung von Release Plan und initialem Backlog.
Maximierung des ROI	Product Owner	<ul style="list-style-type: none">▶ Überwachung des Projekts hinsichtlich der ROI-Ziele und der wirtschaftlichen Vision.▶ Aktualisierung und Priorisierung des Backlog, um sicherzustellen, dass die wertvollste Funktionalität zuerst fertig gestellt wird.▶ Verfeinerung des Product Backlog und Kontrolle von Ausgaben und Erfolgen.

Wer ist wofür verantwortlich? (2)

Aktivität	Verantwortlicher	Verantwortung
Durchführung der Iteration	Team	<ul style="list-style-type: none">▶ Übersetzung / Konkretisierung / Erweiterung der Einträge im Product Backlog in Einträge für das Sprint Backlog.▶ Implementierung
Durchsetzung des Prozesses	Scrum Master	<ul style="list-style-type: none">▶ Aufsetzen eines erfolgreichen Teams durch Sicherstellung, dass Projekt- und Organisationskultur für die ROI-Ziele optimiert werden.▶ Durchsetzung der Scrum Praktiken.▶ Abschirmung des Teams von Störungen von Außen. Entfernen von Hindernissen.
Release	Product Owner	<ul style="list-style-type: none">▶ Entscheidung, wann offizielle Releases erzeugt werden.▶ Verschieben geplanter Releasezeitpunkte.

Scrum erleben: Das Projekt

Fortgeschrittenes

Was bedeutet fertig - „done done“?

- Code eingecheckt
 - Alles kompiliert fehlerfrei
 - Alle automatischen Unit Tests laufen
 - Alle automatischen Akzeptanztests laufen
- Performance-Tests wurden durchgeführt
 - ▶ und die Bottlenecks beseitigt
 - Manuelle Tests wurden durchgeführt
 - ▶ und alle aufgetauchten Probleme wurden beseitigt
 - System wurde in Testumgebung deployt
 - ▶ und auch hier laufen alle Tests

Scrum um technische Praktiken ergänzen

Gefahr: Anhäufung großer technischer Schulden, die uns immer mehr bremsen

- Testgetriebene Entwicklung
 - ▶ Test, Code, Refactor
 - ▶ Evolutionäres Design
- Agile Akzeptanztests
- Pair Programming
- Collective Ownership
- Continuous Integration
- Coding Standards

Planung mit User Stories (1)

- Ziel: Wir arbeiten immer an den wichtigsten Dingen und schaffen möglichst schnell einen Geschäftswert
- (User) Stories
 - ▶ beschreiben das *Was*, nicht das *Wie*
 - ▶ haben immer einen Geschäftsnutzen
 - ▶ stellen keine detaillierten Anforderungen dar
 - ▶ können in einem 1/2 Sprint umgesetzt werden
- Stories werden vom Product Owner vor dem Planungsmeeting erstellt

Planung mit User Stories (2)

- Große Stories (Epics) müssen vor der Umsetzung kleingeschnitten werden
- Formulierung: *Als ... möchte ich ... damit ...*
"Als **diensthabender Wachmann** möchte ich **gleichzeitig mehrere Kameras auf einem Bildschirm betrachten können**, damit **mir kein wichtiges Ereignis entgeht.**"
- Verwaltung der Stories vorzugsweise auf Karteikarten
- Gruppierung von Stories zu einem Feature

Scrum Tools

<http://danube.com/scrumworks>

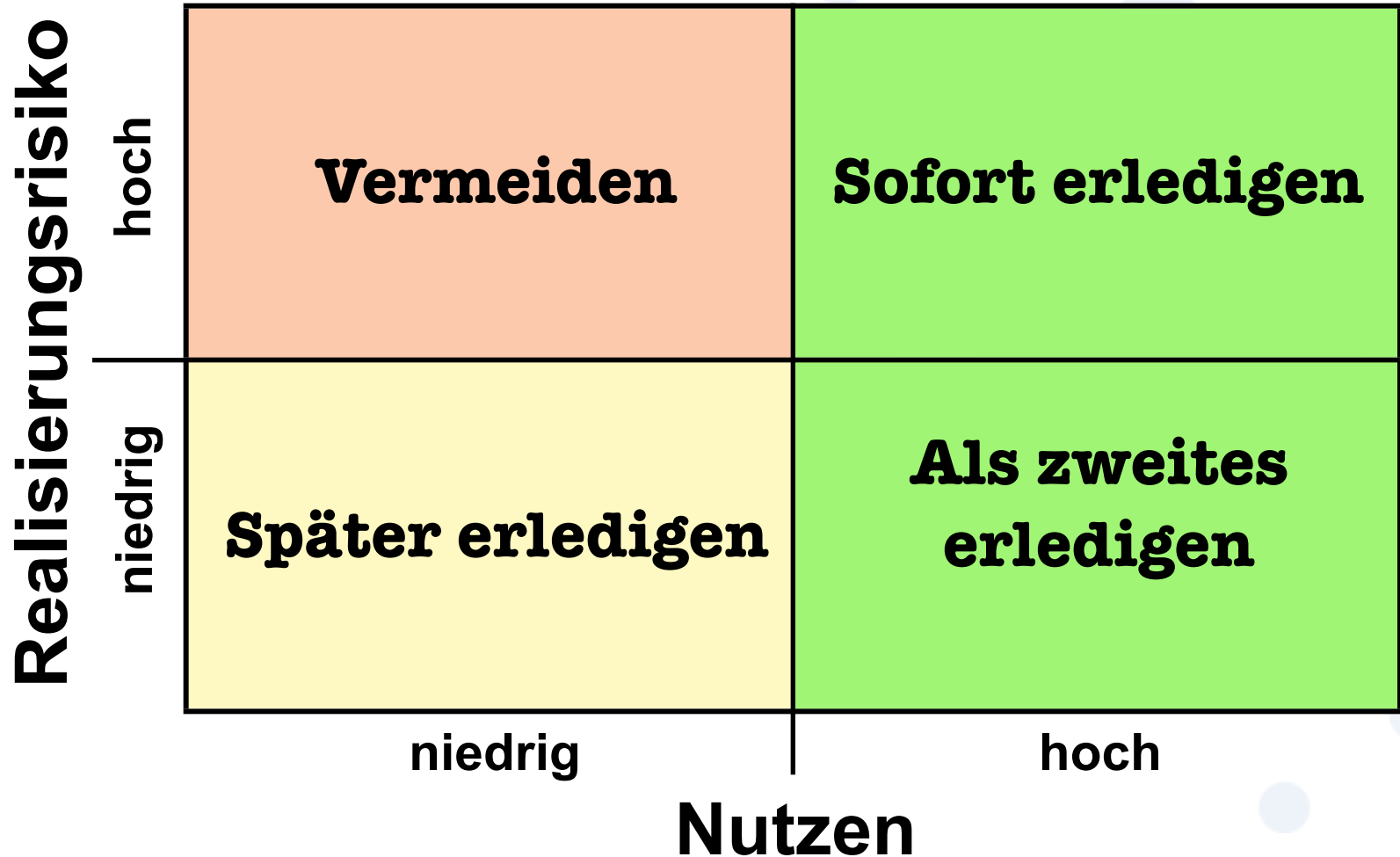
<http://www.versionone.net/>

<http://www.scrumd.com/>

Boris Gloger's Scrum-Tool-Liste:

[http://scrum4you.wordpress.com/2009/03/01/
scrum-tools-list/](http://scrum4you.wordpress.com/2009/03/01/scrum-tools-list/)

Priorisierung



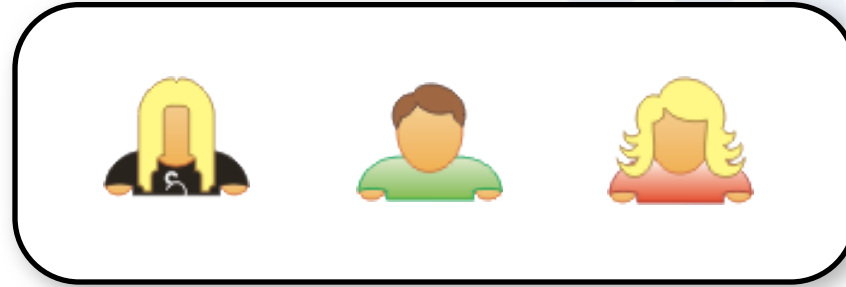
Schätztechniken

- Schätzen in abstrakten, vergleichenden „Story-Punkten“
 - ▶ 1, 2, 3, 5, 8, zu groß
- Schätzpoker, an dem alle im Team teilnehmen
- Grundlage der Zusage für den nächsten Sprint sind die erreichten Punkte des letzten Sprints
 - ➔ automatische Eichung
- Anfängliche Schätzung:
 - ▶ *Ideale Entwicklungszeit*
 - ▶ *Erstellen einer normierten Vergleichsstory*

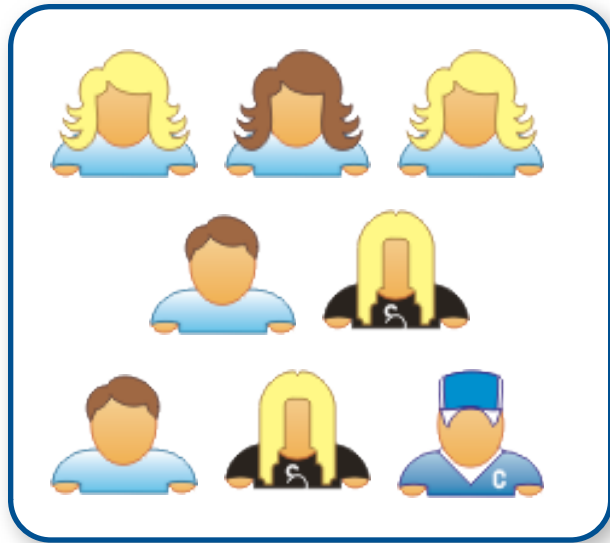
Skalierbarkeit

- Typische Teams bestehen aus 7 ± 2 Personen
 - ▶ Teams von Teams ermöglichen Skalierbarkeit
- Faktoren des Skalierens
 - ▶ Typ der Anwendung
 - ▶ Teamgröße
 - ▶ Teamverteilung (örtlich)
 - ▶ Projektdauer
- Scrum ist mehrmals für 500-Personenprojekte verwendet worden

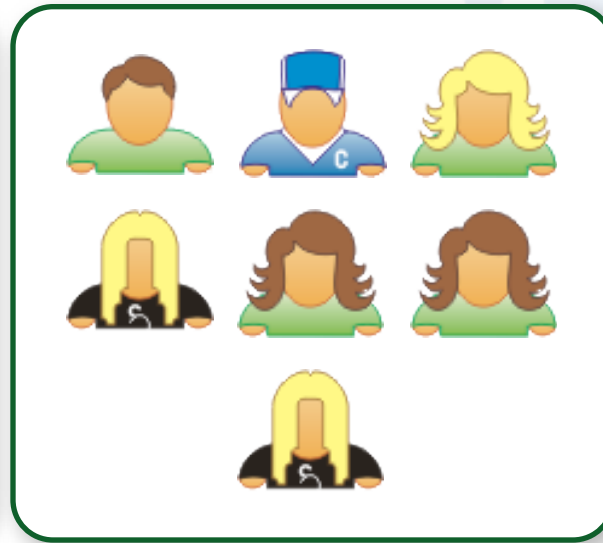
Skalieren mit "Scrum of Scrums"



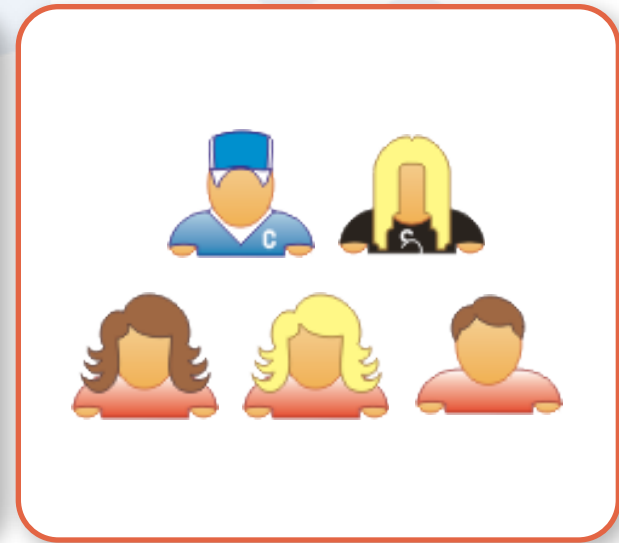
9:45 - 10:00



9:00 - 9:15



9:15 - 9:30



9:30 - 9:45

Vorzeitiges Ende: Abnormal Termination

- Ein Sprint kann vorzeitig abgebrochen werden
- Gründe:
 - ▶ Team sieht keine Möglichkeit das Ziel zu erreichen
 - ▶ Management negiert den Wert des aktuellen Ziels
- Bei vorzeitiger Beendigung besteht der nächste Schritt aus einem neuen Planungsmeeting. Dort wird über die Abbruchgründe gesprochen

Projektbeginn: Voraussetzungen

- Geschulter Product Owner vorhanden
- Geschultes Team vorhanden
- Projektvision ist klar und allen bekannt
- Wichtige Anforderungen sind verstanden und grob spezifiziert
- Risiken wurden gesammelt und bewertet
- Machbarkeit wurde untersucht und positiv beurteilt
- Anfängliche Architektur ist skizziert
- Entwicklungs-, Test- und Build-Umgebungen stehen bereit

Projektbeginn mit Exploration

- Mögliche Ziele:
 - ▶ Klärung der Produktvision
 - ▶ Architekturdurchstich
 - ▶ Einübung von Techniken
 - ▶ Aufbau der Infrastruktur
 - ▶ Bewertung von Risiken
 - ▶ Basis für Schätzungen
 - ▶ Erstellen eines fachlichen Kernmodells
- Durchführung
 - ▶ 2 - 4 Kurzsprints (à 1 Woche)
 - ▶ Am Ende steht eine ausführliche Retrospektive

Agile Schlüsselerlebnisse (1)

- Der Product Owner erkennt, dass es kein Problem ist, nicht alle Anforderungen definiert zu haben.
- Die Entwickler erkennen, dass es kein Problem ist, dass nicht alle Anforderungen dokumentiert sind.
- Auf Grund der regelmäßigen Ergebnisse erkennt der Product Owner die Wichtigkeit seiner Beteiligung und gewinnt einen Glauben an den Projekterfolg.

Agile Schlüsselerlebnisse (2)

- Entwickler erkennen, dass Ihnen geholfen wird.
- Die Projektleiter erkennen, dass sie dem Team nicht sagen müssen, was zu tun ist.
- Innerhalb der Teams lösen sich die Rollen auf. Jeder wird zum Entwickler.

Scrum Master Zertifizierung



- „Offizielles“ CSM-Zertifikat
 - ▶ Seit März 2009 mit Prüfung
- Zusätzliches Material (z.B. Excel-Sheet)
- Aufnahme in CSM-Verzeichnis
<http://www.controlchaos.com/certification/list.php>

Für weitere Informationen

- www.scrumalliance.org
- www.it-agile.de/scrum
- scrumdevelopment@yahoogroups.com
- deutschescrum@yahoogroups.com

Agile Scrum-Leseliste

Agile Software Development von Alistair Cockburn

Agiles Projektmanagement mit Scrum von Ken Schwaber

Scrum - Agiles Projektmanagement erfolgreich einsetzen
von Roman Pichler

Scrum. Produkte zuverlässig und schnell entwickeln von Boris Gloger

Agile Retrospectives von Esther Derby und Diana Larsen

Agile Software Development with Scrum
von Ken Schwaber und Mike Beedle

The Enterprise and Scrum von Ken Schwaber

User Stories Applied for Agile Software Development von Mike Cohn

The Art of Agile Development von James Shore und Shane Warden

Extreme Programming Explained von Kent Beck



Copyrights

Teile dieser Präsentation entnommen aus (der deutschen Version von) “An Introduction to Scrum” von Mike Cohn, übersetzt von Simon Roberts und Birgit Panzram

- Mike Cohn
<http://www.mountangoatsoftware.com/presentation/30--an-overview-of-scrum>
- Simon Roberts
<http://scrumcenter.org/node/2>
- Diese Adaption: Johannes Link
<http://johanneslink.net>
- Ergänzte Inhalte teilweise mit freundlicher Genehmigung der it-agile GmbH (<http://it-agile.de>)



Lizenz



- Sie dürfen:
 - ▶ das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
 - ▶ Bearbeitungen des Werkes anfertigen
- zu den folgenden Bedingungen:
 - ▶ Namensnennung - Sie müssen den Namen des Autors/ Rechteinhabers in der von ihm festgelegten Weise nennen (wodurch aber nicht der Eindruck entstehen darf, Sie oder die Nutzung des Werkes durch Sie würden entlohnt).
- Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.
- Weiteres hierzu unter <http://creativecommons.org/licenses/by/3.0/>

