



Gordon Breuer
Senior IT Consultant
(Product Owner)



Oliver Joncker
Senior IT Consultant
(Scrum Master Coach)

Begrüßung & Vorgehen

- *Was machen wir heute und wie gehen wir vor?*

Zeit		Dauer ca.
10:00	Beginn der Veranstaltung	
	Organisatorisches und Theorie-Block	1h
	Sprint 1 (inklusive Theorie)	1,45h
	Sprint 2 (inklusive Theorie)	1h
13:45	Abschluss / Feedback	0,25h
14:00	Ende der Veranstaltung	

- Kurze Einführung in Scrum
- Wir erproben und erleben Scrum live & agil anhand unseres Projektes „Scrum Lego City“
 - Es gibt immer einen sehr kurzen Theorieblock zu einem konkreten Thema
 - direkt im Anschluss wird das Erlernte verinnerlicht durch Hands-on Übung
 - Im Laufe des heutigen Tages führen wir so ein vollständiges Scrum Projekt durch (mit allen Bestandteilen von Scrum)!
- Im Anschluss sollte noch genug Zeit sein, Fragen zu diskutieren und das Erlebte zu rekapitulieren.

Ziel: Scrum erlernen durch erleben!

Bildet Teams

Setting:

- Stellt Euch an einer vorgegebenen Linie hinsichtlich Eurer gefühlten Scrum-Erfahrung auf (Von wenig Scrum- bis zu viel Scrum-Erfahrung).
- Bildet Gruppen, am Besten zählt ihr jeweils bis zur Anzahl der vorgesehenen Gruppen durch (Z.B. 1..2..1..2.. ...).
- Alle 1er, 2er bilden ein Scrum Team.
- Definiert für den ersten Sprint den Scrum-Master (kann auch wechseln).
- Denkt euch einen Team-Namen aus (notiert diesen auf das task board).

Eine kurze Einführung in Scrum

Characteristika

- Leichtgewichtiger Management-Rahmen
- Einfach zu verstehen (!)
- Guter Startpunkt für Agiles Vorgehen
- Selbstorganisierende Teams
- Produktentwicklung in Serien von Sprints
- Priorisierte Anforderungen im Backlog
- Keine spezifische Entwicklungsmethode vorgeschrieben, stattdessen

Generative Regeln, um ein agiles Umfeld für die Auslieferung von Produkten zu schaffen



Gefahren

- Beteiligte sind nicht reif für Scrum
- Passt nicht in die Organisation
- Einführung ohne Erfahrung mit Scrum
- Scrum & Festpreisprojekte

Agile Manifesto (Agile Werte)



Quelle: www.agilemanifesto.org

Agile Prinzipien

- **Welcome changing requirements**, even late in development.
- **Working software** is the **primary measure** of progress.
- Agile processes promote **sustainable** development: The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.
- ...

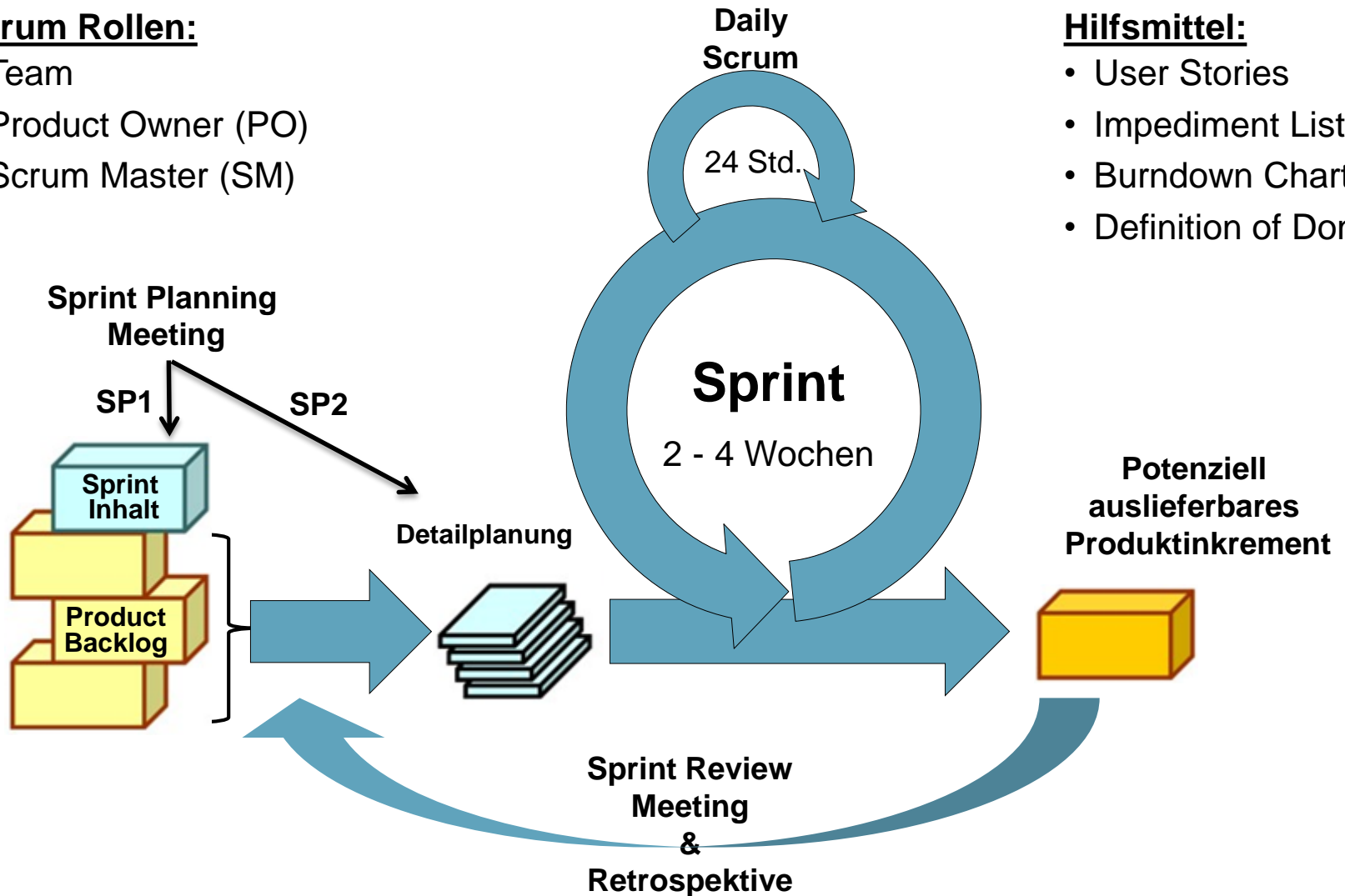
**Agile Methoden basieren auf einem gemeinsamen Wertesystem.
Diese Werte sind notwendig für das Funktionieren von Agilen Methoden.**

Scrum Rollen:

- Team
- Product Owner (PO)
- Scrum Master (SM)

Hilfsmittel:

- User Stories
- Impediment List
- Burndown Charts
- Definition of Done



Team

- Ca. 5-9 Personen (Vollzeit)
- Funktionsübergreifend (cross functional) QS, Programmierer, UI-Designer, etc.
- Mitgliedschaft kann sich nur zwischen den Sprints verändern
- Schätzt Anforderungen
- Steuert sich selbst
- Verantwortlich für zugesagte Lieferung



Product Owner

- Erfasst Bedürfnisse aller Stakeholder
- Bestimmt Auslieferungsdatum und Inhalt
- Ist verantwortlich für Produkt (Releaseplanung) und Projekterfolg (u.a. ROI)
- Definiert und priorisiert Features abhängig von deren Geschäftswert
- Akzeptiert oder weist Arbeitsergebnisse zurück
- Entscheidungsbefugnis (Produkt)



Scrum Master

- Ist verantwortlich für die Einhaltung der Scrum-Werte und -Techniken
- Ist Coach (Scrum-Techniken) und Moderator
- Hilft beim Beseitigen von Hindernissen
- Schützt das Team vor Störungen
- Unterstützt Lernprozess und Selbstorganisation des Teams
- Kurz: „servant of the team“
- Aber ...
 - Darf nicht inhaltlich arbeiten
 - Ist nicht Teil des (Umsetzungs-)Teams
 - Hat keine Weisungsbefugnis gegenüber dem Team



Product Backlog

- Enthält die zukünftigen Features (meist in Form von User Stories)
- Wird nur vom Product Owner verwaltet
- Features sind vom Product Owner nach Geschäftsnutzen priorisiert (Serialisierung)
- Granularität nimmt nach unten hin ab

Definition of done

Wann ist eine User Story „fertig“ ?

Beispiele:

- Feature vollständig implementiert
- Junit-Tests erstellt und erfolgreich getestet
- Integrationstests erfolgt
- Dokumentation im IT Konzept/ Benutzerhandbuch erfasst
- Release Notes angepasst

Sprint Backlog

- Wird vom Team verwaltet (Task Board)
- Besteht aus allen Tasks, die das Team im aktuellen Sprint umsetzt
- Tasks: User Stories werden vom Team in handliche Aufgaben heruntergebrochen (Tasks dauern < 2-8h)
- Tasks enthalten Aufwand

Impediment List

- Liste aller Hindernisse
- Wird vom Scrum Master verwaltet
- Wird genutzt, um Team-interne Hindernisse selbst zu beheben und Team-externe Hindernisse zu kommunizieren.

Burndown-Charts

- Visualisierung bereits geleisteter und noch verbleibender Arbeit.
- Offenbart die „Velocity“
- Für jeden Sprint und ggf. Release

- **Das Product Backlog**
 - Enthält Backlog Items (User Stories, Epics) mit den zukünftigen Features
 - Jeder Eintrag ist mit einem Datum versehen (Eingang im Product Backlog)
 - Es gibt eine klare Sequenz, d.h. Backlog Items haben bzgl. ihrer Priorität eine eindeutig fortlaufende Nummer (Priorität wird vom Geschäftsnutzen abgeleitet).
 - Je „weiter weg“ die Umsetzung der Backlog Items ist, desto „größer“ ist ihre Beschreibung (und ungenauer die Schätzung, z.B. Epics 100).
 - Jedes Backlog Item enthält eine Schätzung für seine Größe, die in Story Points ausgedrückt wird (Die Schätzung ist jeweils Team-individuell).

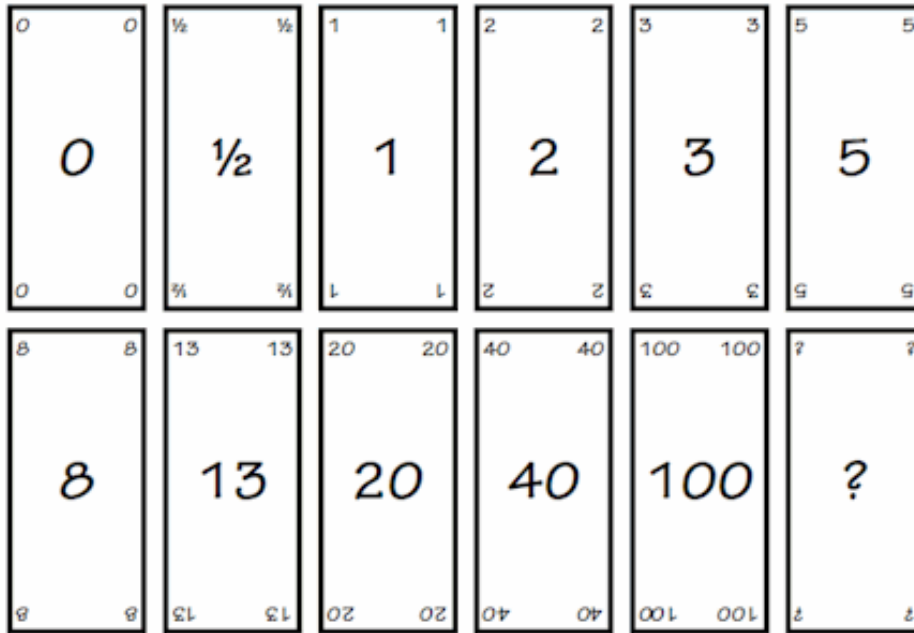
- **Die User Story**
 - Beschreibt eine Anforderung an ein (Software-)System
 - Die Anforderung besitzt einen konkreten und sichtbaren Mehrwert für den Kunden.
 - **Beispiel:**
 - „Als Kunde möchte ich mich am System anmelden.“
 - „Als Kunde möchte ich mir ein Kundenkonto anlegen.“
 - Muster: Als <Rolle> benötige ich <Feature> [um <Grund>].

„Es soll eine Lego Stadt aus Häusern, Autos, Flugzeugen, Straßen und weiteren Objekten gebaut werden.“

Setting:

- Die Teilnehmer erstellen in einem oder mehreren Teams die gewünschte Stadt (je nach Zahl der Teilnehmer)
- Pro Scrum-Team ist ein Scrum Master zu bestimmen (kann auch wechseln)
- Alle Scrum-Teams arbeiten gemeinsam an einem Product Backlog

Lets play „Planning Poker“



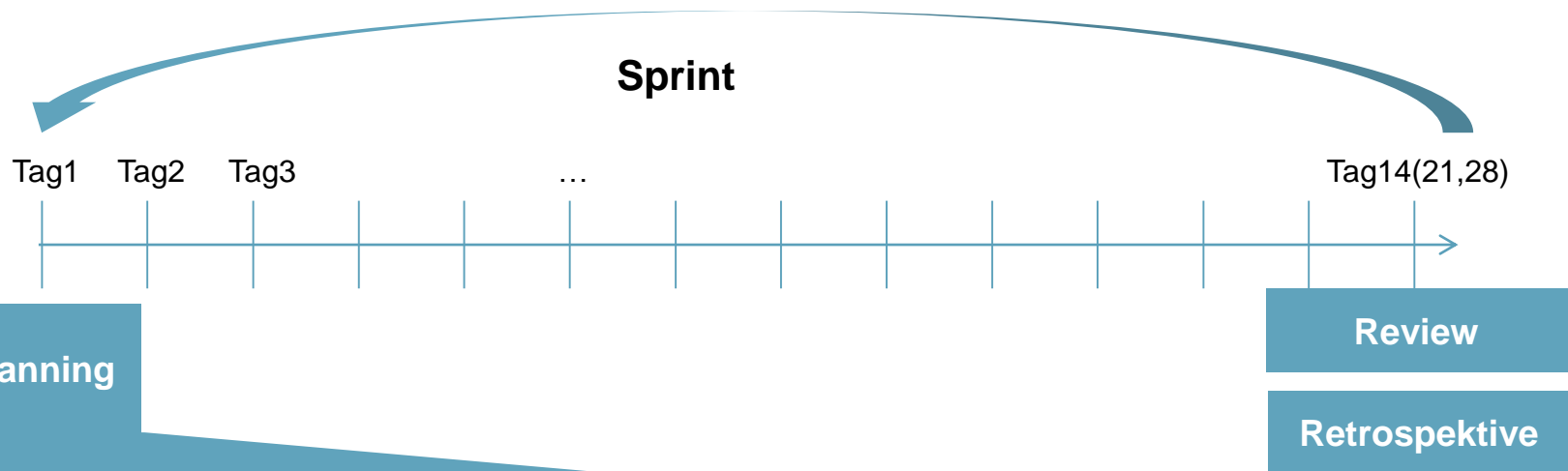
Was ist Planning Poker?

- Expertenschätzung mit dediziertem Schätzmeeting
- Nutzung von identischen Schätzkarten pro Schätzer (angelehnt an Fibonacci Zahlen)
- Schätzung in Story Points

Wie funktioniert Planning Poker?

- Schätzmeeting mit Scrum Master als Moderator
- Schätzer sind in der Regel das Team
- Für jedes zu schätzende Feature werden folgende Schritte durchgeführt:
 - Der Product Owner stellt eine User Story vor
 - Das Team diskutiert die User Story mit dem Product Owner
 - Der Moderator fordert die Schätzer auf, eine Karte auszuwählen und verdeckt auf den Tisch zu legen
 - Alle Karten werden gleichzeitig umgedreht
 - Je ein Ausreißer nach oben und unten werden diskutiert
 - Schätzer einigen sich auf eine gemeinsame Schätzung

***„Planning Meeting“
erklären und dann je Team
durchführen.***



Das „Sprint Planning“ findet zu Beginn jedes Sprints statt

Teil 1 – (Präsentation und Inhalt)

- In diesem Treffen erklärt der Product Owner dem Team alle einzelnen Anforderungen (Backlog-Einträge), die für den nächsten Sprint benötigt werden.
- Product Owner und Scrum-Team einigen sich auf das Sprint-Ziel (Commitment)
- Teilnehmer: Team/PO/SM , Dauer : 1 Stunde pro Sprintwoche

Teil 2 - (Design)

- Dieses Treffen organisiert das Scrum-Team eigenverantwortlich
- Arbeit (User Stories) wird in Aufgaben (Tasks) zerlegt (möglichst jede Aufgabe < 1BT)
- Aus der Summe der Aufgaben entsteht das „Sprint Backlog“
- Teilnehmer: Team/PO/SM , Dauer : 1 Stunde pro Sprintwoche

User Story

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

Tasks

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

8 hours

Max

Code the middle tier

(Task als Karte/Haftnotiz)



Sprint Backlog - Taskboard

User Story	To Do	In Progress	Done
User Story 1		Task 2	Task 1
User Story 2	Task 4	Task 3	
User Story 3	Task 5		
	Task 6	Task 7	
	Task 8	Task 9	
	Task 10		



selected Product backlog

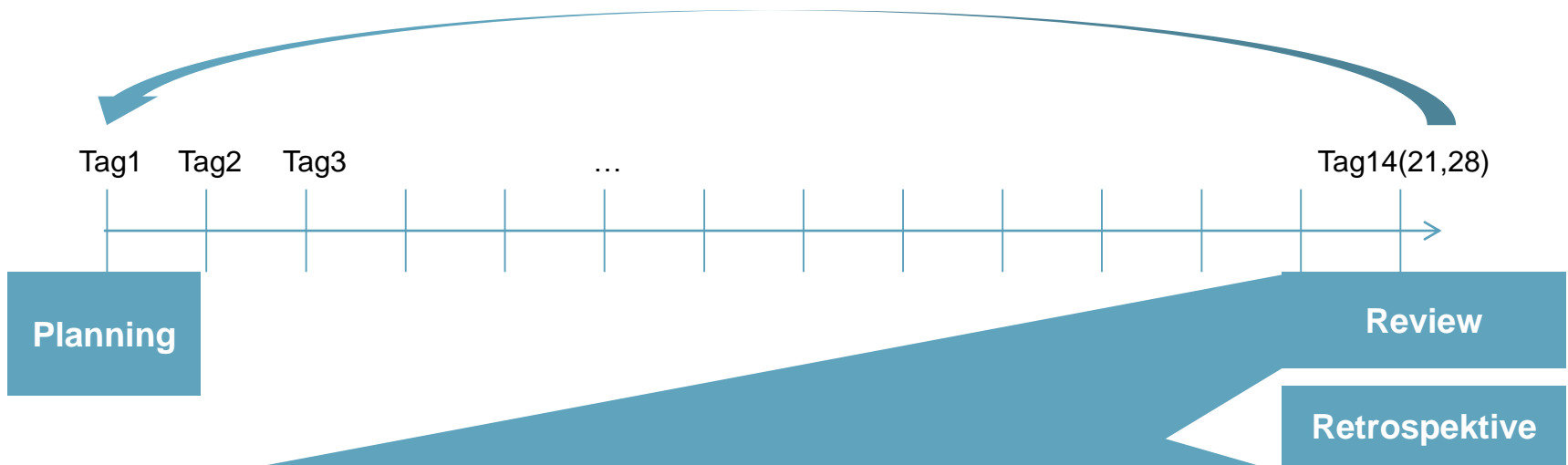
Tasks to do

Burndown chart

Completed tasks

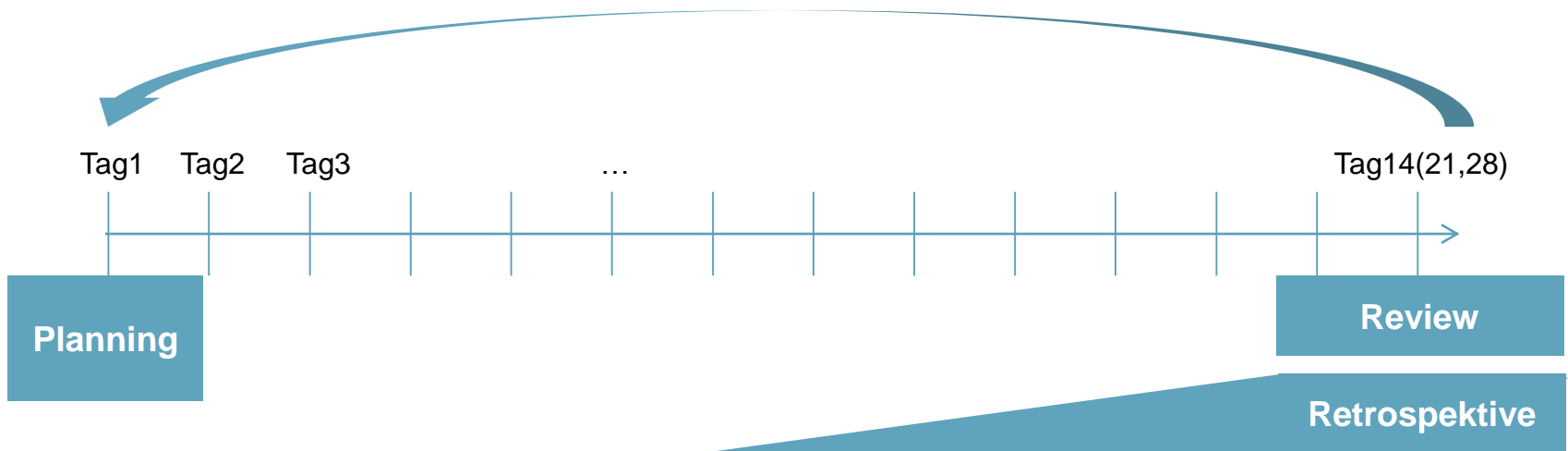
***Lets build our
„Lego City“***

Sprint Review & Retrospektive



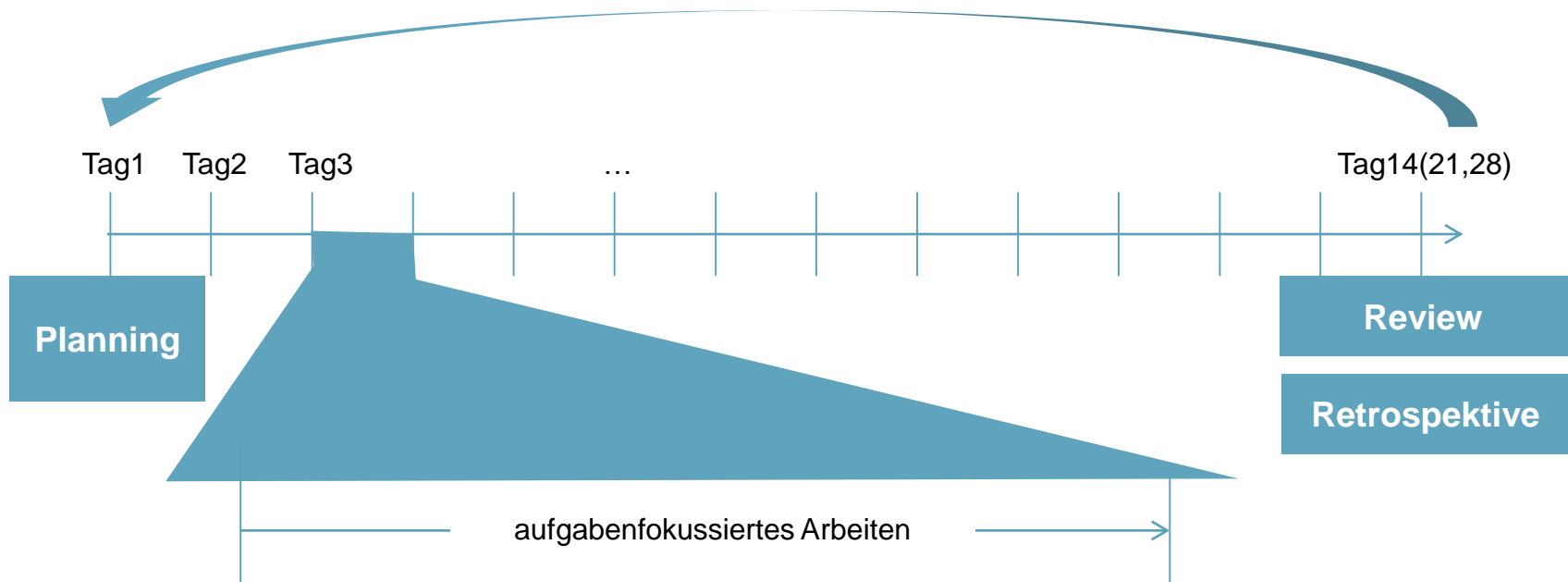
Im „**Sprint Review**“ wird das Sprint Ergebnis einem informellen Review durch das Team und den Product Owner unterzogen.

- Das Ergebnis wird vom Team vorgeführt (=laufende Software, keine Folien)
- Der Product Owner prüft, ob das Ergebnis seinen Anforderungen entspricht und entlastet dann das Team.
- Eventuelle Änderungen führen zu Anpassungen an Product Backlog.
- Teilnehmer: Jeder (frühes Feedback), Dauer: ca. 45 Minuten pro Sprintwoche



In der „**Sprint Retrospektive**“ wird der zurückliegende Sprint betrachtet

- Folgende Fragen werden von allen beantwortet:
 - „Was war gut?“ (best practice)
 - „Was könnte verbessert werden?“ (Verbesserungspotential)
- Jedes Verbesserungspotential wird protokolliert und aktiv verfolgt.
- Scrum Master treibt die Umsetzung der Verbesserungspotentiale aktiv voran (direkt & indirekt).
- Teilnehmer: Team und SM (Moderator) , weitere auf Einladung
- Dauer: 45 Minuten pro Sprintwoche



Im „Daily Scrum“ berichtet sich das Team

- Selbststeuerung für das Team
- Teilnehmer: Team, SM (Moderator), Zuhören darf jeder
- Jeden Tag zur gleichen Zeit und Ort (i.d.R. vor Task Board)
- Zeitlich begrenzt auf 15 min, Pünktlicher Start

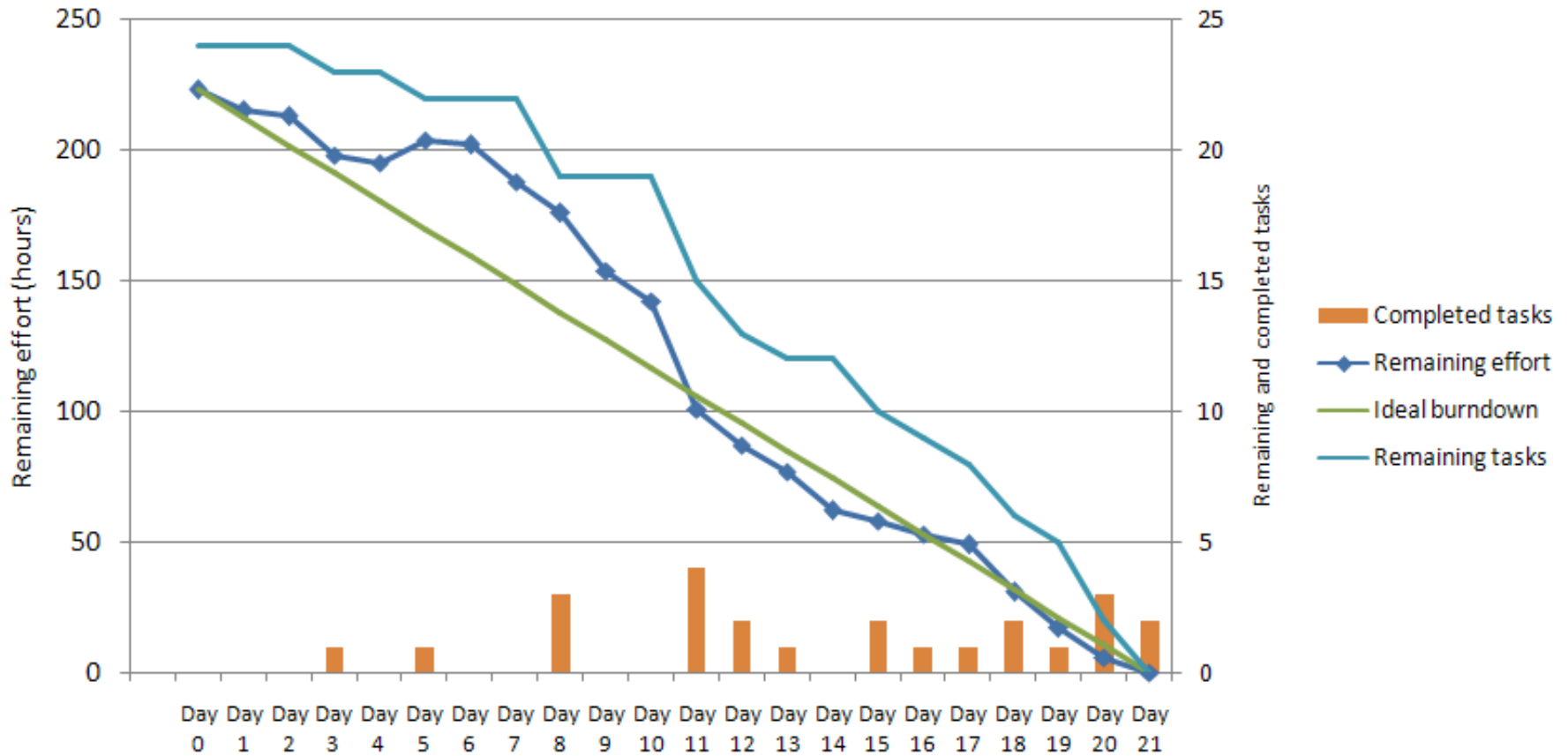
Jedes Teammitglied beantwortet drei Fragen:

- Was habe ich seit gestern erreicht?
- Was plane ich heute zu machen?
- Gibt es Hindernisse, welche verhindern, dass ich mein Ziel erreiche?

Daily Build

Nicht vorgeschrieben in Scrum
Dennoch sehr sinnvoll und hilfreich
Abschluss einer Aufgabe auf Tagesebene

Sample Burndown Chart



Übersicht der Meetings in einem Sprint

	Tag1	Tag2	Tag3		Tag13	Tag14 (21, 28)
08:00						
	Sprint Planning 1	Daily	Daily		Daily	Daily
10:00						
	Sprint Planning 2					
12:00						
14:00						Sprint Review
16:00						Sprint Retrospektive
18:00						

Vielen Dank für Ihre Aufmerksamkeit

msg systems ag (Travel & Logistics)

Peter Kretz (Bereichsleiter West)
Max-Planck-Straße 40 1
50354 Hürth/Köln
Peter.Kretz@msg-systems.com
Tel: 0151/42652425

Gordon Breuer (Senior IT Consultant)
Mergenthalerallee 73-75
65760 Eschborn
Gordon.Breuer@msg-systems.com
Tel: 0160/90915143

www.msg-systems.com

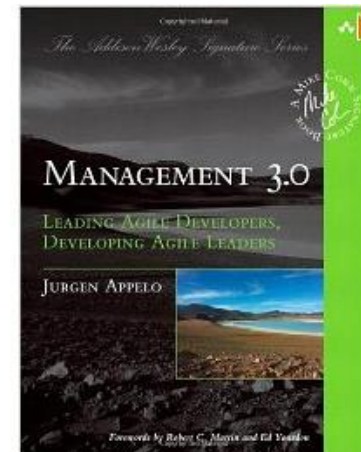
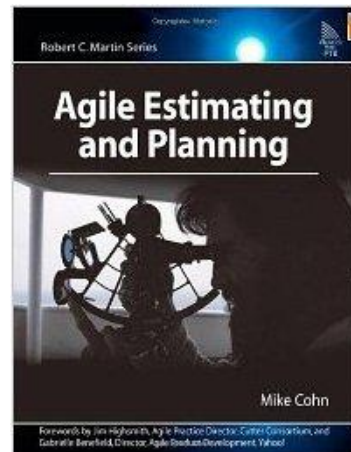
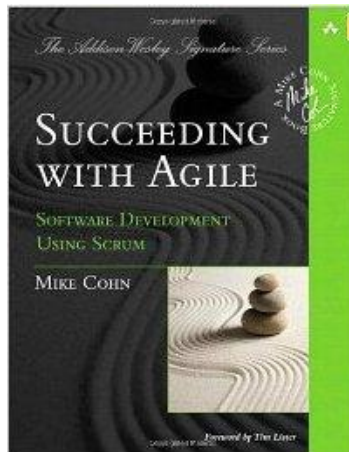


.consulting .solutions .partnership



Anhang

- Succeeding with Agile: Software Development Using Scrum; Mike Cohn;
<http://www.mountaingoatsoftware.com/books/7-succeeding-with-agile-software-development-using-scrum>
- Agile Estimating and Planning; Mike Cohn;
<http://www.mountaingoatsoftware.com/books/1-agile-estimating-and-planning>
 - Brief Summary: <http://www.niwotridge.com/BookReviews/AgileEstimatingandPlanning.pdf>
- The Software Project Manager's Bridge to Agility (Agile Software Development), Michele Sliger, Stacia Broderick;
<http://www.sligerconsulting.com/resources/books/>
- Management 3.0: Leading Agile Developers, Developing Agile Leaders, Jurgen Appelo;



Aspekt	Scrum Master	Product Owner	Team
Risikomanagement	✓ (Prozess)	✓ (Projekt/Release)	✓ (Technisch)
Zeit	✗	✓ (Projekt/Release)	✓ (Sprint)
Umfang	✗	✓ (Projekt/Release)	✓ (Sprint)
Kosten	✗	✓ (Projekt/Release)	✓ (Sprint)
Qualität	✓ (Prozess)	✓ (Umfang)	✓ (Testing)
Kommunikation	✓ (Prozess)	✓ (Projekt/Release)	✓ (Sprint)
Sourcing*	✓ (Anforderung)	(Anforderung)	✗

Die PL-Rolle ist aufgeteilt auf Product Owner und Team.

*Das Sourcing/Staffing erfolgt in enger Abstimmung mit dem Linenmanagement.

- PKW (Familie) in welchem mindestens 5 Personen Platz finden und ausreichend Laderaum zur Verfügung steht
- Flugzeug mit weißen Tragflächen und vier Turbinen
- PKW (Single)
- Einfamilienhaus mit einem Stockwerk und drei Bäumen im Garten
- Kranwagen mit ausziehbarem Arm und einer Seilwinde zum Heben und Umsetzen von Gegenständen
- Schwimmbad mit Sprungturm
- Kirche mit (Glocken-)Turm
- LKW mit 4 Achsen
- LKW-Garage in welcher der LKW (4-Achser) Platz findet, um von oben wettergeschützt untergestellt werden zu können.
- Einfamilienhaus mit Garten (einstöckig, PKW Stellplatz, Dach mit roten Ziegeln, Wände mit max. zwei Farben
- Mehrfamilienhaus mit zwei Stockwerken und einem Ladengeschäft im Erdgeschoss

- Hubschrauber mit vierarmigen Hauptrotor und einem Heckrotor
- Straßennetz, welches die Gebäude miteinander verbindet und Ampeln an den Kreuzungen
 - min. zwei Kreuzungen gefordert
- Einfamilienhaus mit zwei Stockwerken und Stellplatz
- PKW (Familie) in welchem mindestens 7 Personen Platz finden
- LKW Zugmaschine inklusive Anhänger mit je zwei Achsen

- Hauswände und Dächer sind „luftdicht“
- Elemente der Lego City sind stabil genug → zerfallen nicht bei leichtem Fingerdruck
- Die einzelnen Stadtelemente sind nicht zu bunt
- Die Elemente der Stadt sind in ein Gesamtstadtbild integriert