

# Softwaretechnik 1(A)

# Moderne Vorgehensmodelle

Autorin: Prof. Dr. Sabine Sachweh

## Themen dieser Veranstaltung

- **Agile Vorgehensmodelle**
- **Scrum**
- **ScrumTale**



# Moderne Vorgehensmodelle

## 01: Agile Vorgehensmodelle

## Der Frust

- der Einsatz „schwerfälliger“ Vorgehensmodelle, die sich selbst dienen und die Softwareentwicklung nicht hinreichend im Fokus haben, frustrierten unzählige Entwickler
- häufig schien ihre tägliche Arbeit Vorlage für den nächsten Dilbert-Cartoon zu sein



© Scott Adams, Inc./Dist. by UFS, Inc.

## Die Situation

- Mitte Februar 2001 trafen sich 17 (frustrierte) Softwaretechniker in einer Lodge in einem Skigebiet der Wasatch-Berge in Utah (USA), um Ski zu fahren, zu entspannen und sich auszutauschen ...
- doch es kam anders ...



## Der Aufstand

- diese kleine Minderheit schloss sich zusammen, um ein „Manifest“, eine „Streitschrift“, zu verfassen und den „Rest der Welt“ damit zu konfrontieren ...



## Agile Werte

- **Rückkopplung**
  - Kommunikation zwischen Teammitgliedern
  - Feedback vom Kunden durch frühe Softwareauslieferung
- **Mut**

Offenheit und Ehrlichkeit  
(besonders bei der Kommunikation von negativen Dingen)
- **Respekt**

Nur wer sich respektiert fühlt kann offen sprechen

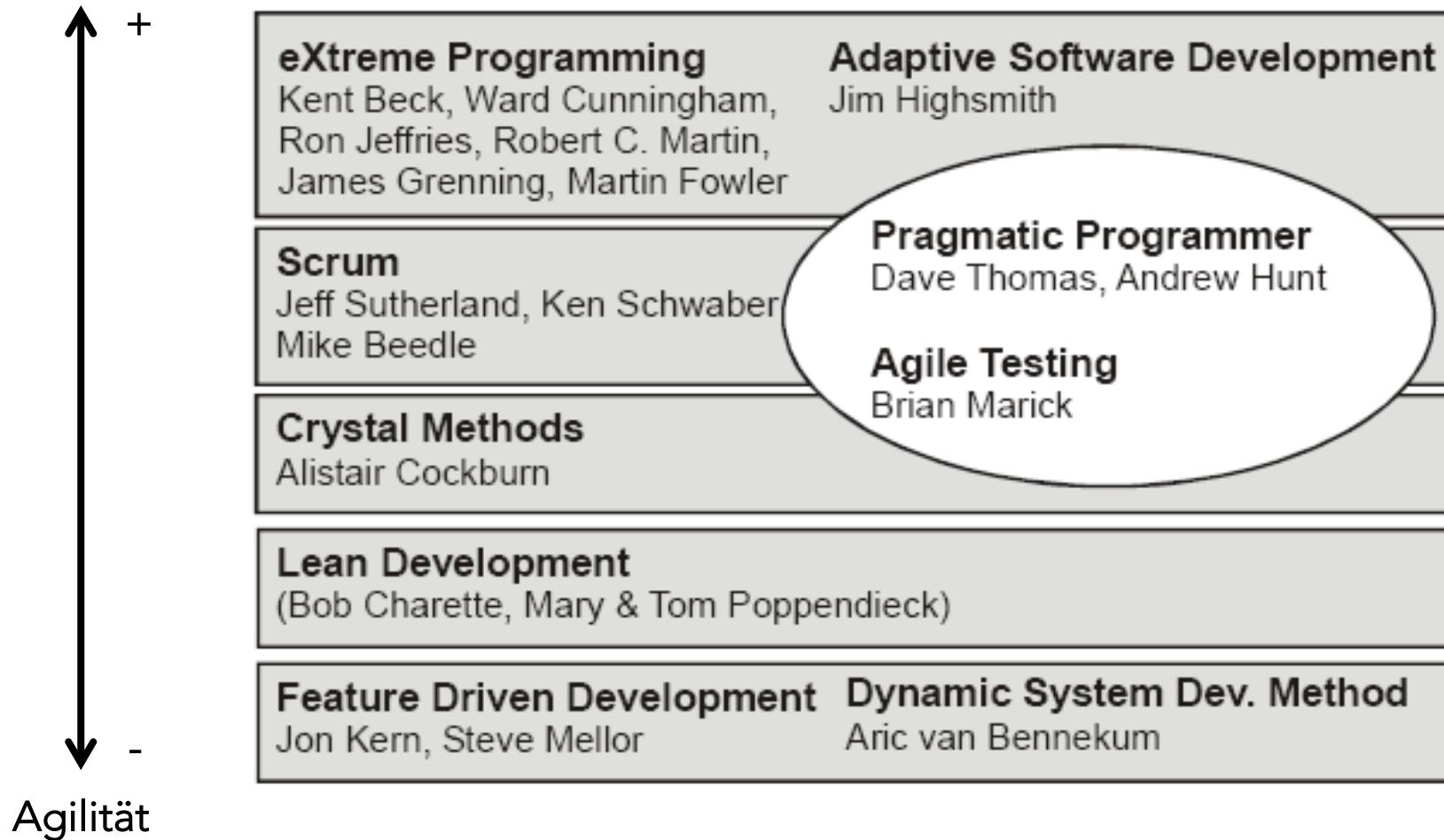
## Zwölf Prinzipien

1. Unsere höchste Priorität ist den Kunden durch frühzeitige und kontinuierliche Lieferung hochwertiger Software zufrieden zu stellen
2. Wir nehmen auch späte (bezogen auf den Prozess) Anforderungsänderungen gerne auf. Agile Prozesse nutzen den Wandel für den Wettbewerbsvorteil des Kunden.
3. Auslieferung funktionierender Software in Intervallen von ein paar Wochen bis zu ein paar Monate, mit der Präferenz für kürzere Zeiträume.
4. Domänenexperten und Entwickler müssen täglich während des gesamten Projekts zusammenarbeiten.
5. Baue Projekte um motivierte Individuen, gib ihnen die Umgebung und Unterstützung, die sie brauchen und vertraue darauf, dass sie die Arbeit erledigen.
6. Die effizienteste und effektivste Methode um Informationen an und innerhalb eines Entwicklerteams zu vermitteln ist das persönliche Gespräch.

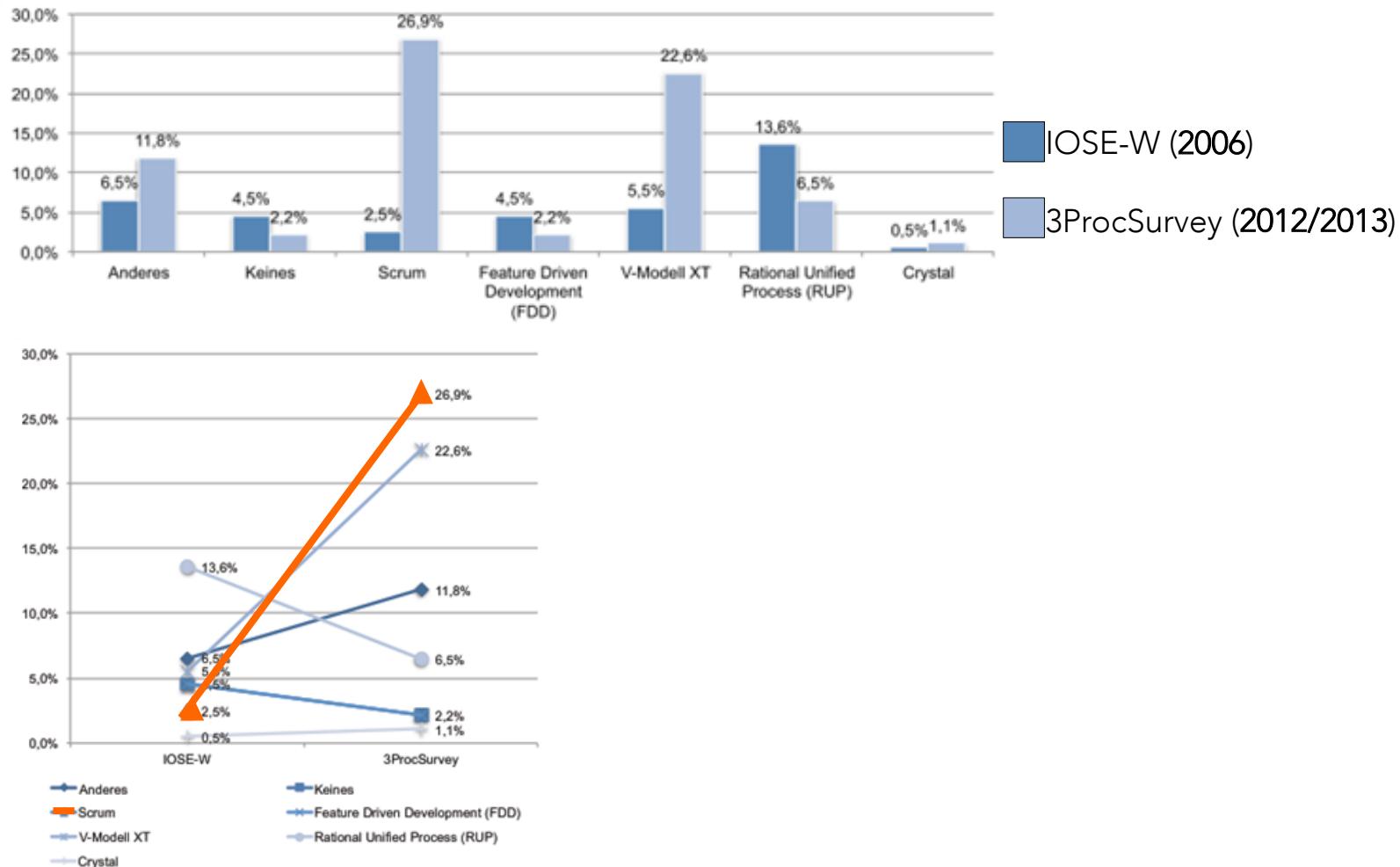
## Zwölf Prinzipien

7. Funktionierende Software ist das primäre Maß für den Fortschritt.
8. Agile Prozesse erfordern eine stabile Arbeitsumgebung. Geldgeber, Entwickler und Benutzer sollten in der Lage sein auf unbestimmte Zeit ein konstantes Engagement beizubehalten.
9. Kontinuierliche Aufmerksamkeit auf hervorragende technische Qualität und gutes Design erhöhen die Flexibilität.
10. Einfachheit - die Kunst, die Menge ungetaner Arbeit zu maximieren - ist essentiell.
11. Die besten Architekturen, Anforderungen und Designs werden von selbstorganisierenden Teams erschaffen.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann, stimmt sich ab und passt das Verhalten dementsprechend an.

## Typische Agile Vorgehensmodelle



## Relevanz im Zeitverlauf





# Moderne Vorgehensmodelle

## 02: Scrum

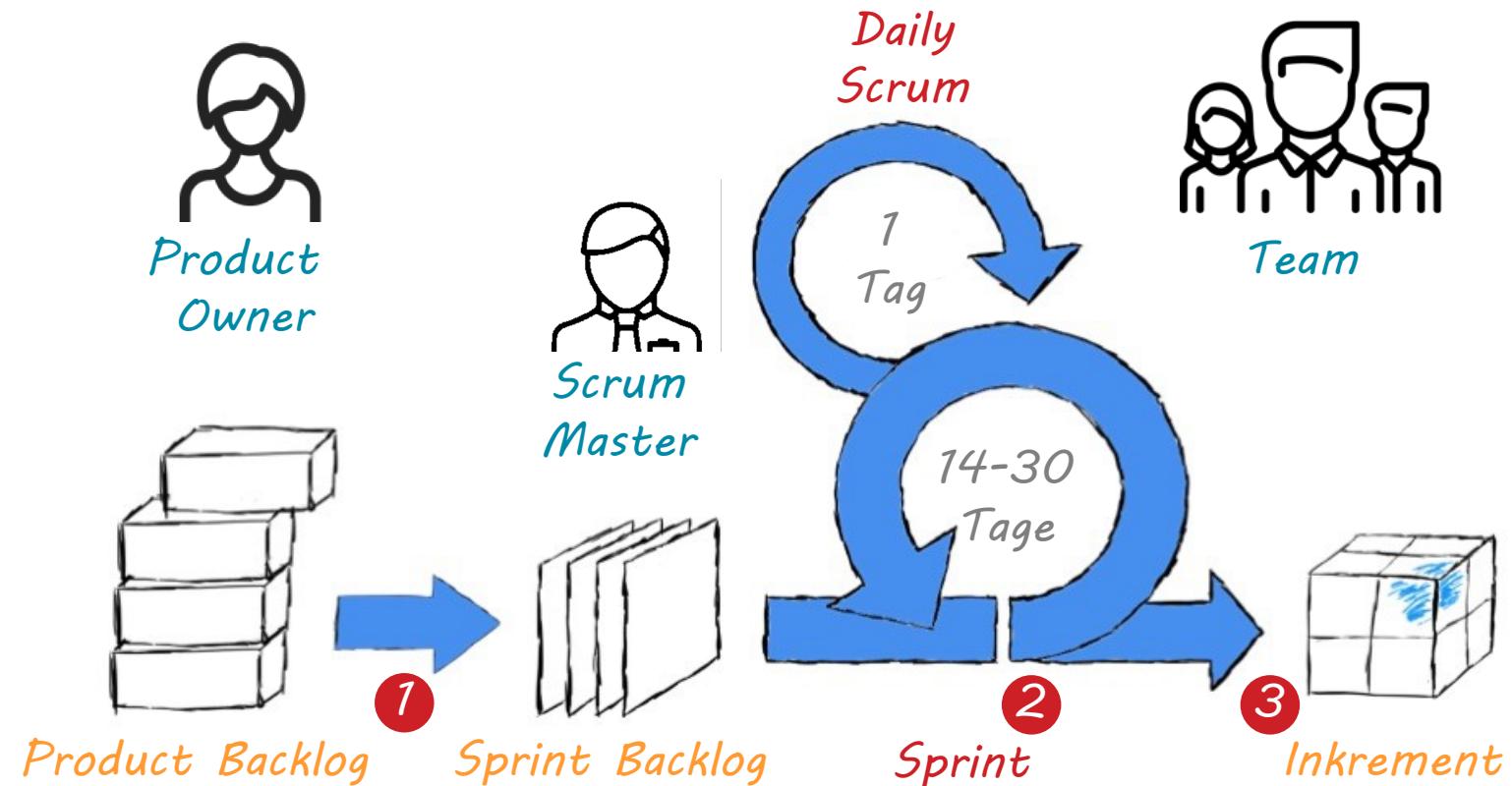
# Scrum

- **keine Entwicklungsmethode** im eigentlichen Sinne, sondern Managementrahmen für beliebige Projekte
- **einfach zu erlernen** → leichtgewichtig
- **Prozessablauf** grob festgelegt, aber bietet **genügend Spielraum** zur Reaktion auf ungeplante Veränderungen
- **wesentliche Elemente** (3-5-3)
  - Rollen
  - Artefakte
  - Events (Meetings und Zyklen)
- das **Team** sollte 3-9 Personen enthalten



## Scrum

- 1 Sprint Planning
- 2 Sprint Review
- 3 Retrospektive



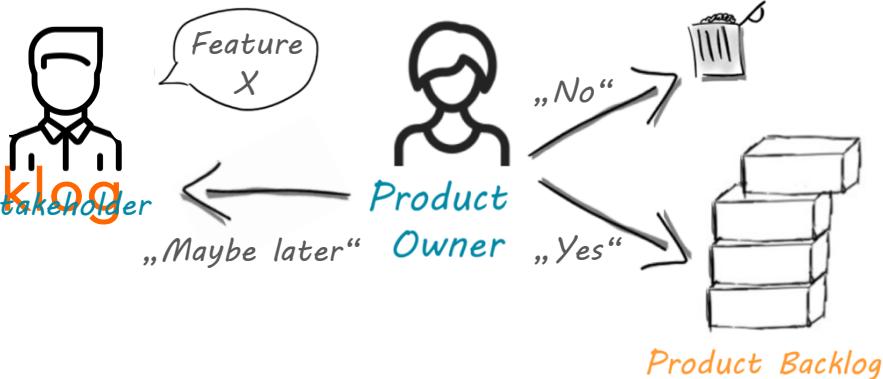
## Rollen – Product Owner



Product  
Owner

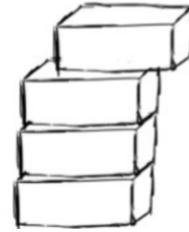
- verantwortlich den Wert des Produkts zu maximieren, der sich aus der Arbeit des Entwicklungsteams ergibt
- verantwortet das Product Backlog, dazu zählt:
  - eindeutige Beschreibung von Backlog-Items;
  - Priorisierung der Items, um die Ziele und die Vision am besten umzusetzen;
  - Arbeit des Entwicklungsteams bestmöglich einsetzen;
  - sicherstellen, dass das Product Backlog für alle sichtbar, transparent und klar ist und zeigt, woran das Scrum-Team als Nächstes arbeiten wird;
  - sicherstellen, dass das Entwicklungsteam die Items so weit wie erforderlich verstanden hat
- er kann Teile der o.g. Aufgaben an das Entwicklungsteam delegieren, dennoch bleibt er selbst rechenschaftspflichtig
- er ist eine Person, die ggf. auch die Wünsche eines Ausschusses vertreten kann, aber wer die Priorität eines Items im Product Backlog ändern möchte, muss sich an den Product Owner wenden
- die gesamte Organisation muss die Entscheidungen des Product Owners respektieren und niemand kann das Entwicklungsteam zu anderen Aufgaben zwingen

## Artifact - Product Backlog



- Wünsche der beteiligten Stakeholder fließen in das Product Backlog ein
- umfasst Liste von **Backlog Items**, also von Aufgaben, die zu erledigen sind, wie beispielsweise
  - eine Recherche, eine Evaluation oder ein Experiment (→ „Spike“) oder
  - eine Fehlerbeseitigung (→ „Defect“ bzw. „Bug“) oder natürlich
  - eine Anforderung beispielsweise formuliert als User Story (→ „Feature“)
- stellt eine **momentane Sicht** dar, die im Laufe des Projektes erweitert oder angepasst werden kann – Änderungen werden jedoch erst im nächsten Sprint berücksichtigt
- es werden nur bekannte und **keine eventuellen zukünftigen Anforderungen** aufgenommen

## Exkurs: User Story



- Beschreibt Anforderung so präzise und einfach wie möglich, sodass jeder Benutzer sie verstehen kann
- eine Story kann frei formuliert werden wie z.B.:

- **Anwendung starten:**

Die Anwendung startet, indem sie das zuletzt bearbeitete Dokument des Anwenders öffnet, damit der Anwender Zeit spart.

Aktor unklar!

- **Anwendung schließen:**

Wenn der Anwender die Anwendung beendet, erscheint eine Anfrage, ob das bearbeitete Dokument gespeichert werden soll, damit Änderungen nicht verloren gehen.

Was heißt hier beendet?

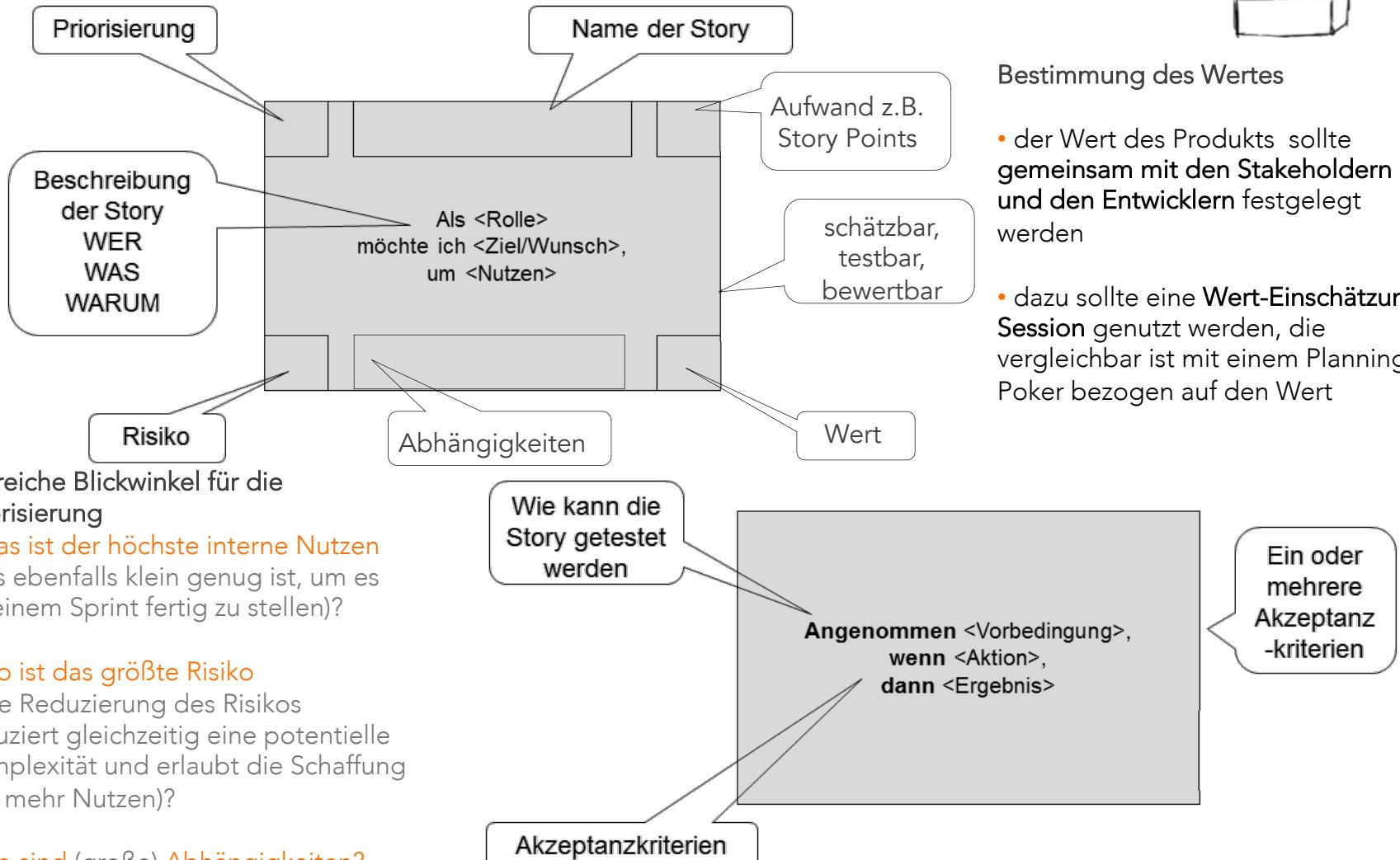
- hilfreich ist die Story zuallererst in einem Satz wie folgt zu beschreiben:

Als **<BENUTZER>** möchte ich **<FUNKTION>**, um damit **<NUTZEN>** zu erzielen.

- **Anwendung starten:**

Als Autor möchte ich nach dem Start der Anwendung mein zuletzt bearbeitetes Dokument sehen, um Zeit zu sparen.

## Exkurs: Story (Item) Card



## Exkurs: Items splitten

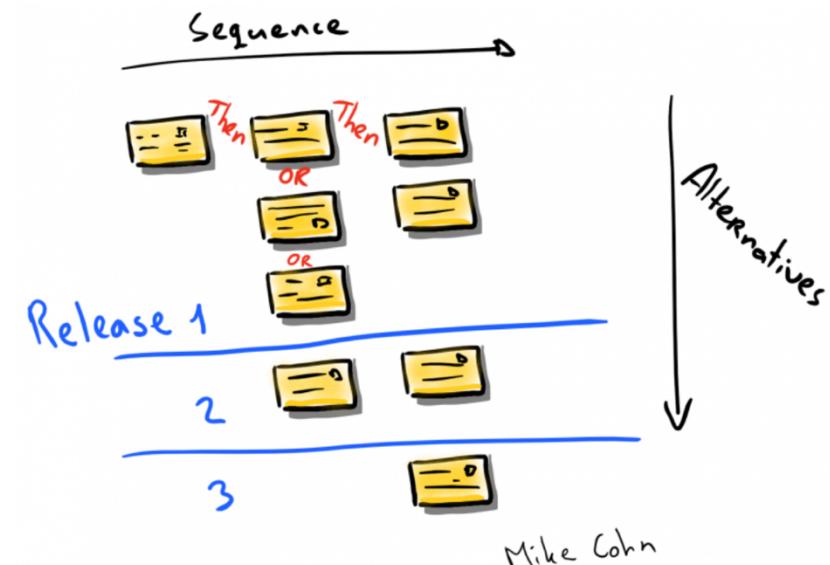


- Items dürfen nicht zu groß sein, d.h. zu große Items müssen gesplittet werden
- Splitten von Items mithilfe des SPIDR Ansatzes (nach Mike Cohn):
  - **Spikes** – (ein paar) Nachforschungen betreiben, um das Item besser zu verstehen
  - **Paths** – Items auf Basis des Pfades in der Applikation (Use Case).  
Ist ein „Login“ über verschiedene Kanäle möglich, dann kann man sich zunächst auf einen Kanal wie „Login über Xing“ beschränken
  - **Interfaces** – einbauen, um Möglichkeit späterer Erweiterung (→ Feinschliff) zu erlauben, während die ersten Iterationen sich auf das Wesentliche
  - **Data** – nur einen Teil der angedachten Daten oder Datenarten verwenden, wie beispielsweise ein Filter, der zunächst nur wenig Attribute nutzt, aber später erweitert werden kann
  - **Rules** – bestimmte Regeln (früh) ggf. in der ersten Iteration festlegen, wie zum Beispiel „Das System sollte innerhalb von 5 Millisekunden laden“  
→ Nichtfunktionale Anforderungen

## Exkurs: Story Map



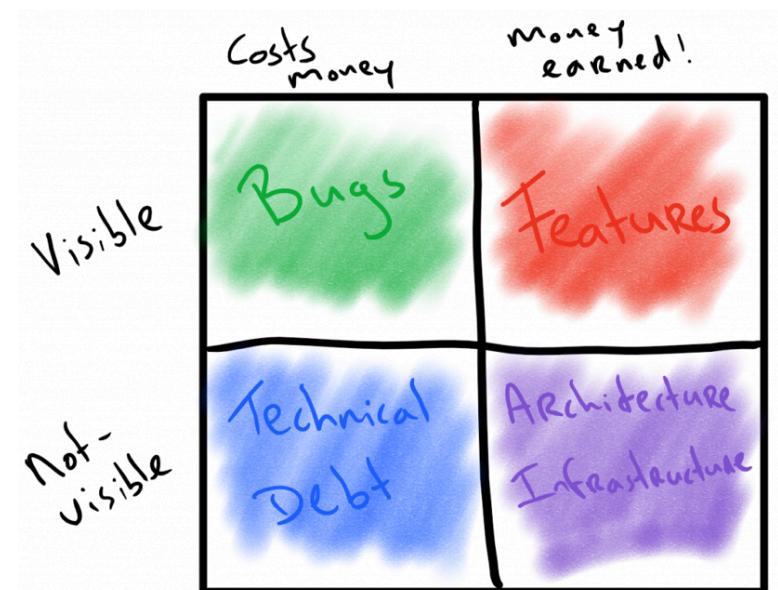
- im Product Backlog sind die **Aufgaben nach Priorität** geordnet, was es erschwert den Überblick darüber zu bekommen, wie das **Produkt** am Ende funktionieren wird
- die Story Map wird eingesetzt, um eine andere Sicht auf die Items bezogen auf die Funktionalität (→Use Case) zu visualisieren
- Beispiel:  
Für ein PC-Spiel würde man oben links starten mit dem „Laden des Spiels“, dann würde wahrscheinlich „Starte ein neues Spiel“ oder „Vorherige Highscores angucken“ kommen ...



## Exkurs: Backlog Quadrant (optional)



- Product Owner, sollten nicht nur über das Erstellen von Features nachdenken  
→ der Wert eines Produktes hängt von wesentlich mehr Faktoren ab
  - jedes Team macht auch Fehler und
  - in jedem jedem Projekt sammeln sich bewusst oder unbewusst technische Schulden an
  - Investitionen in die Zukunft wie in Infrastruktur oder Architektur der Plattform sind ebenfalls relevant
- beim Ausbalancieren hilft der Backlog Quadrant



## Rollen – Scrum Master



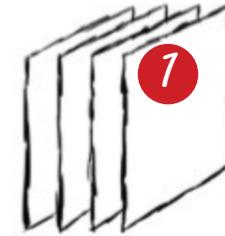
- hilft die Scrum-Theorie, -Praktiken, -Regeln und -Werte zu verstehen und umzusetzen → Scrum Service Provider
- unterstützt den Product Owner u.a. durch:
  - Sicherstellen, dass Ziele, Umfang und Domäne vom Team so gut wie möglich verstanden werden, ebenso wie die Wichtigkeit klarer und präziser Item-Beschreibungen;
  - Finden von Methoden für ein effektives Product Backlog Management;
  - Hilfe beim Verstehen und Erlernen von Flexibilität;
  - Ermöglichung von Scrum-Events nach Bedarf;
- unterstützt das Entwicklungsteam auf verschiedene Arten, darunter:
  - (falls erforderlich, Scrum-Schulung des Teams)
  - Coaching in Selbstorganisation und funktionsübergreifender Arbeitsweise;
  - Unterstützung bei der Entwicklung hochwertiger Produkte;
  - Beseitigung von Hindernissen für den Projekt-Fortschritt;
  - Ermöglichung von Scrum-Events nach Wunsch oder Bedarf;
- unterstützt die Organisation(seinheit) u.a. durch:
  - Führung und Coaching bei der Einführung von Scrum;
  - Unterstützung von Mitarbeitern und Stakeholdern beim den ersten Scrum-Projekten;
  - Veränderungen, die die Produktivität des/der Scrum-Teams steigern;
  - Austausch mit anderen Scrum-Mastern, um die Effektivität von Scrum in der Organisation zu erhöhen

## Events - Sprint Planning

- im Sprint Planning, das der Scrum Master organisiert, wird der nächste Sprint geplant
- die Dauer eines Sprint Plannings für einen 30-Tage-Sprint darf einen Arbeitstag nicht überschreiten
- Das Sprint Planning adressiert zwei Fragen:
  - 1. Was kann man mit diesem Sprint erreichen?
    - Der Input für dieses Meeting ist das **Product Backlog**, das **neueste Produktinkrement**, die **projizierte Kapazität** des Entwicklungsteams während des Sprints und die **vergangene Leistung** des Entwicklungsteams.
    - Der Product Owner **diskutiert das zu erreichende Ziel** und die damit verbundenen Backlog Items
    - Das Entwicklungsteam **bewertet den Aufwand** der Backlog Items  
→ z.B. **Planning Poker**
    - Die Menge der Items, die für den Sprint ausgewählt werden, werden ausschließlich durch das Entwicklungsteam bestimmt
    - Während der Sprint-Planung erstellt das Scrum-Team auch ein Sprint-Ziel. Das Sprint-Ziel ist ein Ziel, das innerhalb des Sprints erreicht werden soll, und es gibt dem Entwicklungsteam einen Hinweis darauf, warum das Inkrement erstellt wird.



## Events - Sprint Planning → Artefact: Sprint



■

- ...
- 2. Wie werden die Aufgaben erledigt?
  - das Ergebnis dieser Frage wird im **Sprint Backlog** festgehalten
  - das **Sprint Backlog** enthält die ausgewählten Items sowie den Plan, wie diese erstellt werden sollen plus mindestens ein Item zur Prozessverbesserung aus der Retrospektive
  - das Entwicklungsteam beginnt normalerweise mit dem Entwurf des Systems und der Arbeit, die erforderlich ist, um das Product Backlog in ein funktionierendes Produktinkrement umzuwandeln
  - bis zum Ende des Meetings sind die vom Entwicklungsteam für die ersten Tage des Sprints geplanten Arbeiten in Einheiten von einem Tag oder weniger zerlegt
  - das Entwicklungsteam organisiert sich selbst, um die übernommenen Aufgaben sowohl während der Sprint-Planung als auch ggf. während des gesamten Sprints zu managen
  - das Entwicklungsteam kann Personen einladen, technische oder fachliche Beratung zu leisten.
  - wenn sich dabei herausstellt, dass die Items nicht passen, so kann das Team mit dem Product Owner einen Kompromiss oder eine Änderung aushandeln

## Events - Sprint Planning

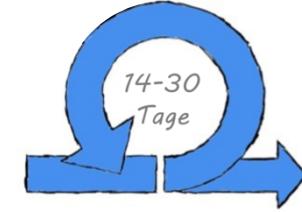
- am Ende der Sprint-Planung sollte das Entwicklungsteam dem Product Owner und Scrum Master erklären können, wie es beabsichtigt, als selbstorganisierendes Team zu arbeiten, um das Sprint-Ziel zu erreichen und das erwartete Ergebnis zu erstellen
- **Sprint Ziel**
  - ist eine Zielvorgabe für den Sprint
  - es gibt dem Entwicklungsteam Flexibilität hinsichtlich der im Sprint umzusetzenden Funktionen
  - es ist ein Element, das die Aufgaben im Team verbindet und einen größeren Zusammenhang (Sinn) herstellt
- innerhalb eines Sprints werden diese Rahmenbedingungen nicht modifiziert, die **Verwaltung des Spring Backlogs** ist im Sprint ein Arbeitsinstrument und liegt in den Händen des Entwicklungsteams

## Rollen – Entwicklungsteam



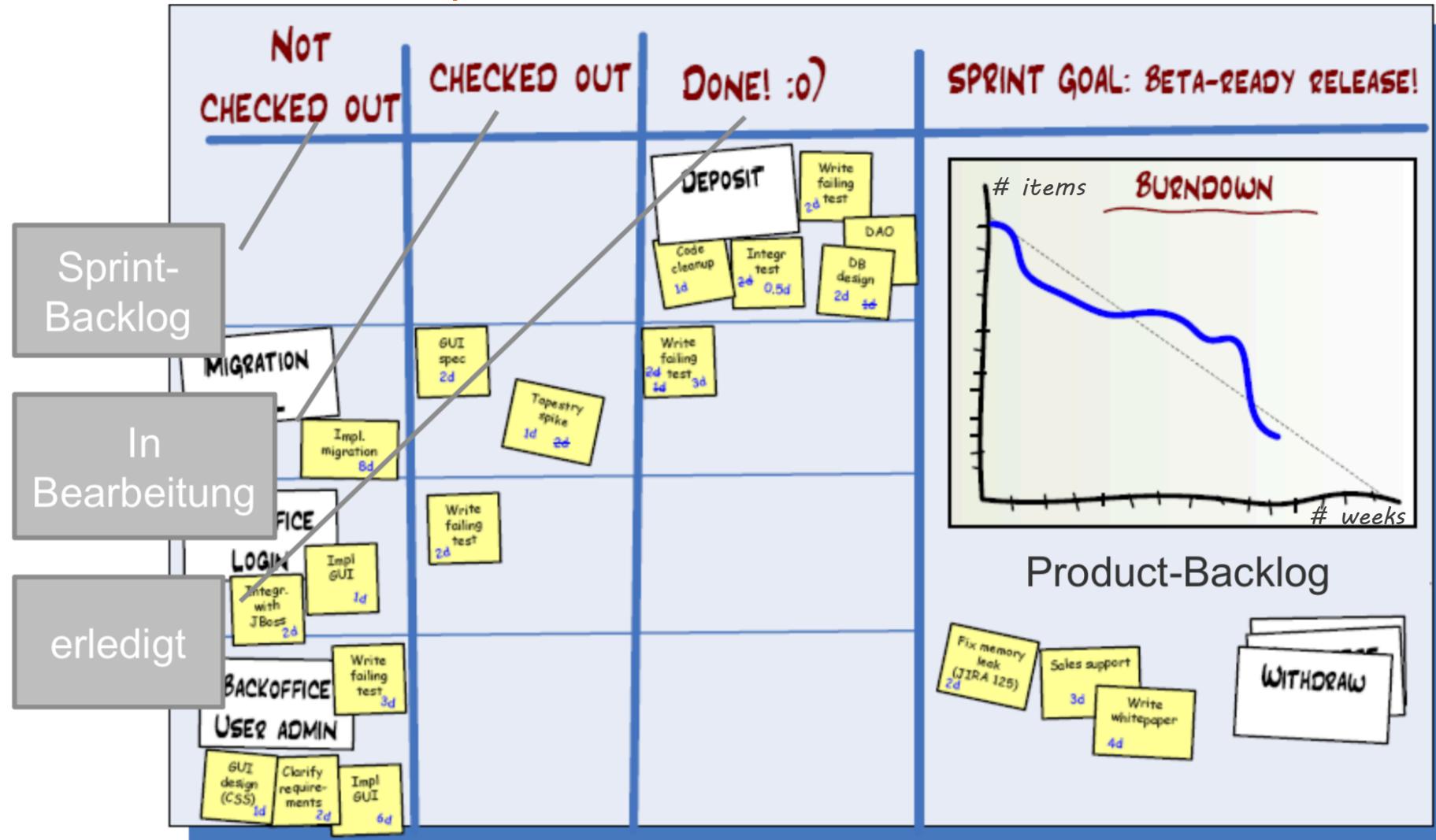
- das Team besteht aus Fachleuten, die gemeinsam bis zum Ende eines jeden Sprints ein abnehmbares („Done“) Inkrement erstellen können, ohne auf externe Ressourcen zugreifen zu müssen
- das Team organisiert sich selbst
- Entwicklungsteams haben die folgenden Eigenschaften:
  - Keiner (nicht einmal der Scrum Master) teilt dem Entwicklungsteam mit, wie sie arbeiten müssen.
  - Entwicklungsteams arbeiten funktionsübergreifend und verfügen über alle Fähigkeiten, die zur Erstellung eines Produktinkrements erforderlich sind.
  - Es gibt keine Subteams, unabhängig von den zu erfüllenden Aufgaben wie z. B. Testen, Architektur, Betriebsabläufe oder Geschäftsanalysen.
  - Auch wenn einzelnen Mitglieder möglicherweise über spezielle Fähigkeiten verfügen, so obliegt die Rechenschaftspflicht jedoch beim gesamten Entwicklungsteam.

## Events - Sprint

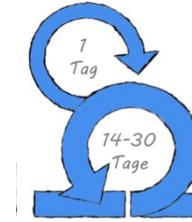


- haben eine feste Länge (2-4 Wochen)
- während des Sprints:
  - arbeitet das Entwicklerteam das Sprint-Backlog eigenverantwortlich ab
  - ohne Einflüsse und Druck von außen
  - es werden keine Items durch Product Owner oder Scrum Master ergänzt
  - der Umfang kann zwischen dem Product Owner und dem Entwicklungsteam geklärt und neu ausgehandelt werden, falls sich die Annahmen wesentlich verändern
  - Rückfragen beim Product Owner sind erlaubt
- Sprints ermöglichen Vorhersagbarkeit, indem sie mindestens jeden Kalendermonat die Überprüfung und Anpassung des Fortschritts sicherstellen.
- ⇒Sprints begrenzen das Risiko auch auf einen Kalendermonat.

## Events – Sprint (Kanban)



## Events – Daily Scrum



- dient dazu Fortschritt auf dem Weg zum Sprint-Ziel festzustellen
- findet (täglich) zur gleichen Zeit am gleichen Ort statt  
(max. 15 Minuten)
- wird vom **Scrum Master** organisiert, doch das Entwicklerteam ist für die Ausgestaltung verantwortlich
- Was wird besprochen?
  - Was habe ich seit dem letzten Daily Scrum getan? → gestern
  - Was werde ich bis zum nächsten Daily Scrum tun? → heute
  - Was behindert mich? → Probleme
- ausführlichere Diskussionen schließen sich häufig direkt an
- Daily Scrums verbessern die **Kommunikation**, reduzieren andere Besprechungen, **identifizieren Probleme**, markieren und fördern **schnelle Entscheidungen** und **verbessern den Wissensstand** des Entwicklungsteams.

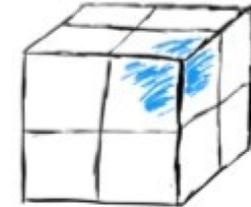
## Events – Sprint Review

- wird am **Sprint-Ende** von **Scrum Master** organisiert (max. 4 Stunden)
- zu den **Teilnehmern** zählen das Scrum-Team und wichtige Stakeholder, die vom Product Owner eingeladen wurden
- es ist eine **informelle Besprechung**, keine Prüfung, die Feedback hervorrufen und die Zusammenarbeit fördern soll  
⇒ Ergebnis: überarbeitetes Sprint Backlog + Input für die nächste Iteration
- die Aufgaben sind wie folgt verteilt:
  - **Product Owner**: erklärt, welche Product Backlog-Elemente "erledigt" und welche nicht "erledigt" wurden. Nicht erledigte Items wandern zurück ins Product-Backlog.
  - **Entwicklungsteam** beschreibt: was während des Sprints gut gelaufen ist, auf welche Probleme es gestoßen ist und wie diese Probleme gelöst wurden.
  - **Entwicklungsteam** demonstriert: die Arbeitsergebnisse (laufende Software – kein PowerPoint) und beantwortet Fragen dazu.
  - **Product Owner** erörtert das Product Backlog in seiner aktuellen Form und leitet daraus voraussichtlichen Ziel- und Liefertermine ab (falls gewünscht).
  - **Alle**: bearbeiten, was als Nächstes zu tun ist, sodass das review einen wertvollen Input für die nachfolgende Sprint-Planung liefert.
  - **Überprüfung**, ob sich Rahmenbedingungen verändert haben, z.B. die Marktsituation oder die Verwendung des Produktes
  - **Überprüfung** des Zeitplans, des Budgets, der potenziellen Funktionen und des Marktes für die nächsten Versionen

## Events – Sprint Retrospektive

- wird nach der Sprint-Überprüfung und vor der nächsten Sprint-Planung von **Scrum Master** organisiert (max. 3 Stunden)
- obwohl Verbesserungen jederzeit implementiert werden können, bietet die Sprint-Retrospektive eine formelle Gelegenheit, sich darauf zu konzentrieren
- Der Zweck der Sprint Retrospektive ist:
  - **Rückblick**: wie der letzte Sprint in Bezug auf Personen, Beziehungen, Prozesse und Tools verlaufen ist.
  - **Analyse**: Identifizieren und priorisieren der wichtigsten Punkte, die gut gelaufen sind, und potenzieller Verbesserungen.  
**(Was war gut? - Was war schlecht?)**
  - Erstellung eines **Plans** zur Implementierung von **Verbesserungen** der Arbeitsweise mit dem Ziel die Produktqualität durch verbesserte Arbeitsabläufe zu erhöhen oder die Definition von "Done".  
→ **Auswahl der Verbesserung**, die im nächsten Sprint umgesetzt werden sollen.

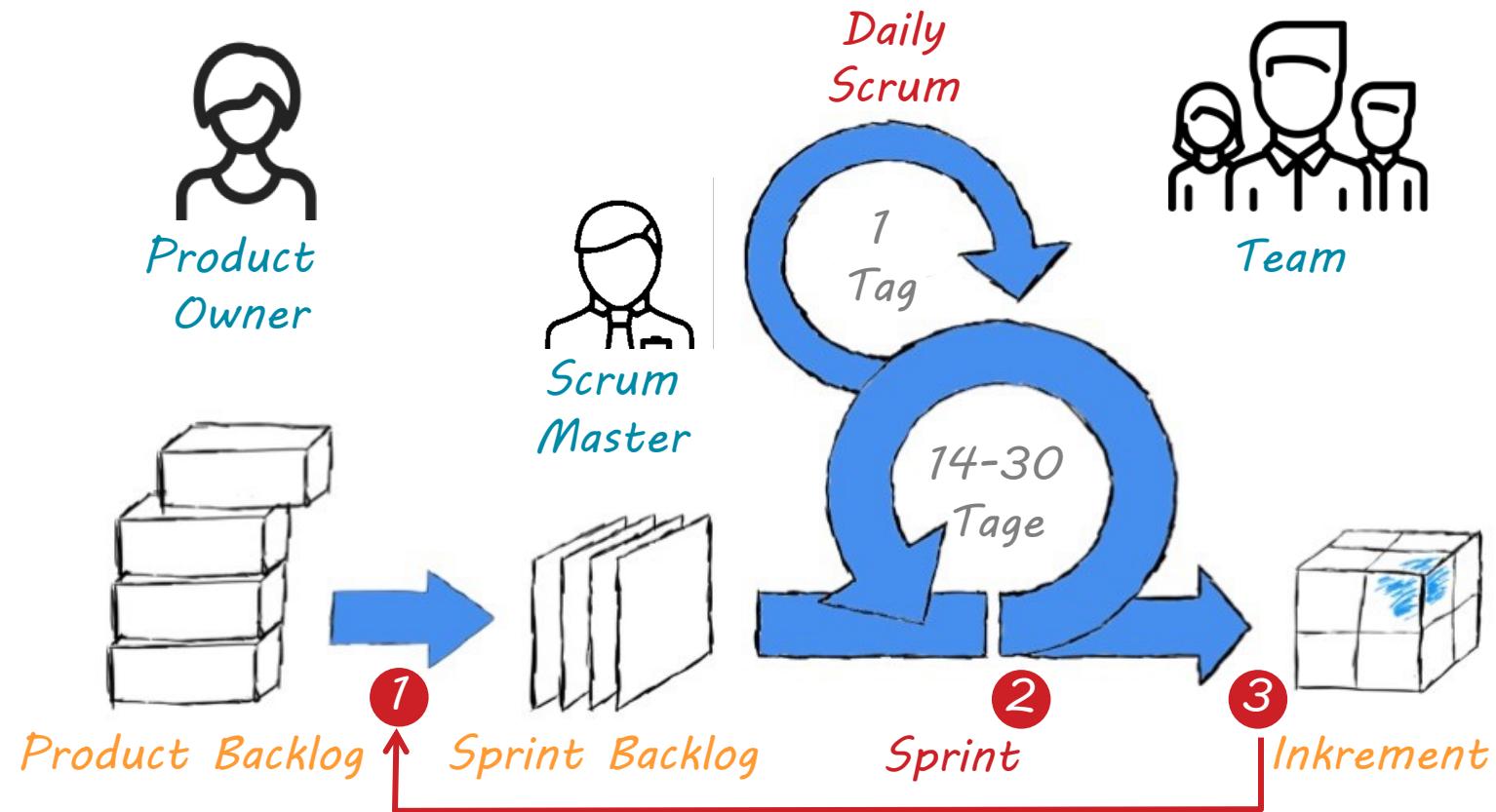
## Artefact – Inkrement



- ein Inkrement beschreibt eine Reihe überprüfbarer, erledigter Arbeiten (Items)
- umfasst die Summe aller während eines Sprints abgeschlossenen Items ("Done") und die Ergebnisse aller vorherigen Sprints
- das Inkrement muss sich in brauchbarem Zustand befinden, unabhängig davon, ob der Product Owner beschließt, es freizugeben
- **Definition of Done (DoD)**
  - gemeinsames Verständnis des Teams davon, was es bedeutet, wenn eine Arbeit abgeschlossen ist, um Transparenz zu gewährleisten
  - sie hilft auch bei der Bestimmung der Arbeit, die in einem Sprint erledigt werden kann
  - Ermöglicht, dass ein Product Owner ein Inkrement möglicherweise sofort freigeben kann → muss auch Integrationstest umfassen

## Scrum

- 1 Sprint Planning
- 2 Sprint Review
- 3 Retrospektive





# Moderne Vorgehensmodelle

## 03: Scrum Tale



ScrumTale  
SIMULATION GAME

The logo for ScrumTale features the word "ScrumTale" in a large, blue, serif font. To the left of the "S", there is a blue line-art icon that forms a stylized infinity symbol or a loop, with a small blue arrow pointing upwards and to the right, indicating a cycle or iteration.

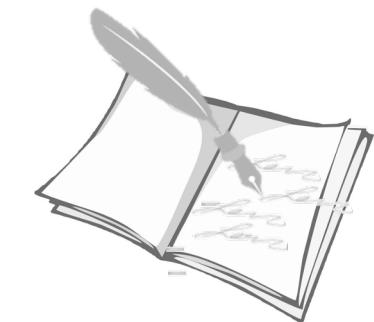
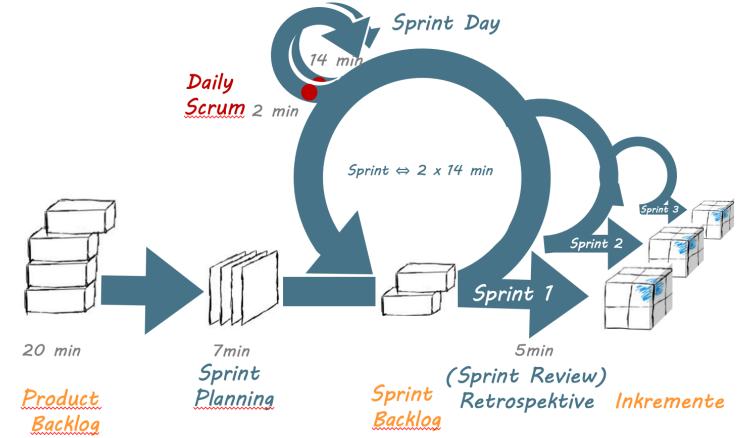
## Ziel des Workshops

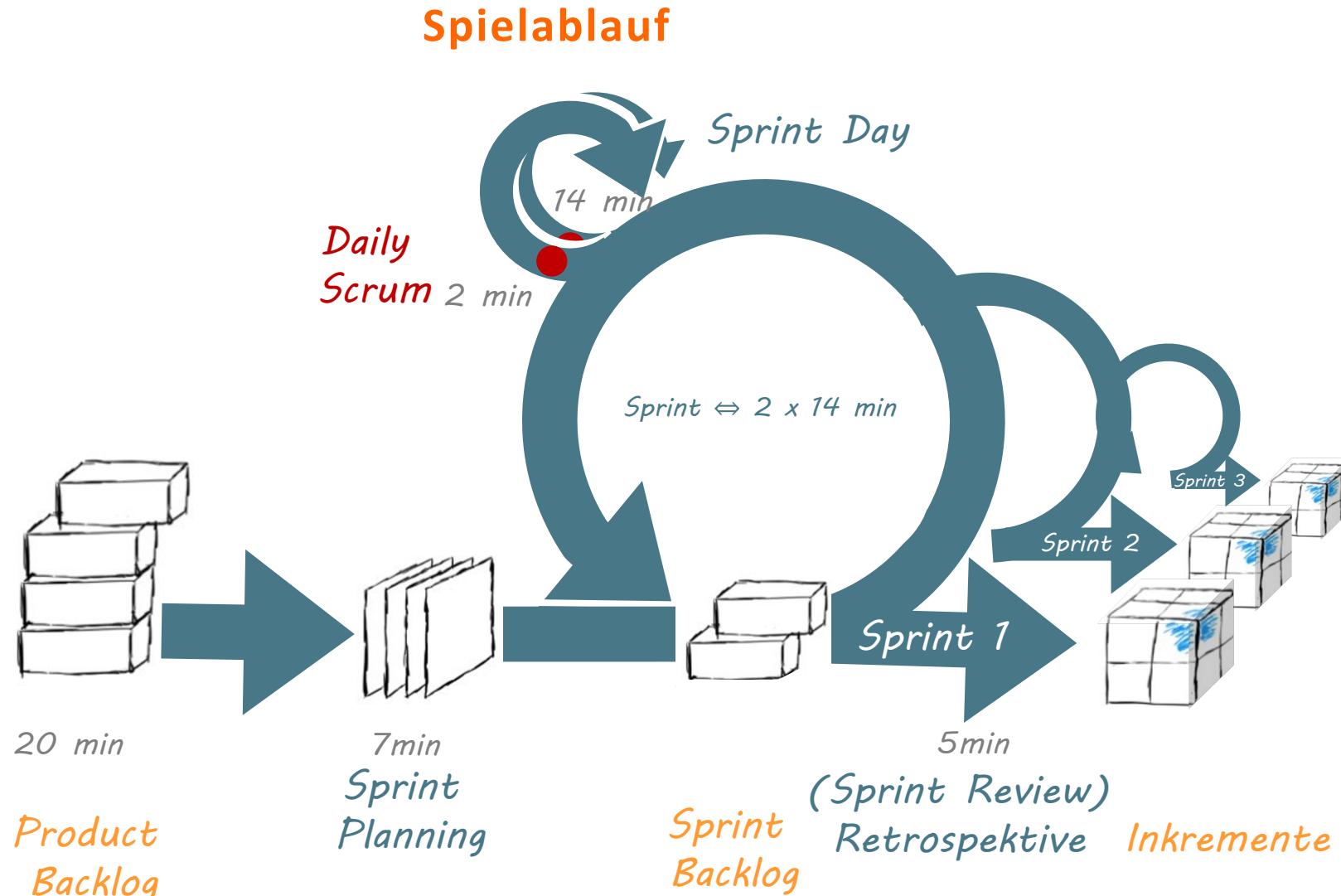
### ■ Primäres Ziel:

Ziel ist es, zu erfahren, wie es **sich anfühlt, Scrum einzusetzen** und gemeinsam ein Produkt in einem iterativem Ansatz bereitzustellen.

### ■ Sekundäres Ziel (Beispiel):

Das Ziel dieses **Simulationsspiels** ist es, mit Scrum die beste **Kriminalgeschichte** aller Zeiten zu schreiben ;-)





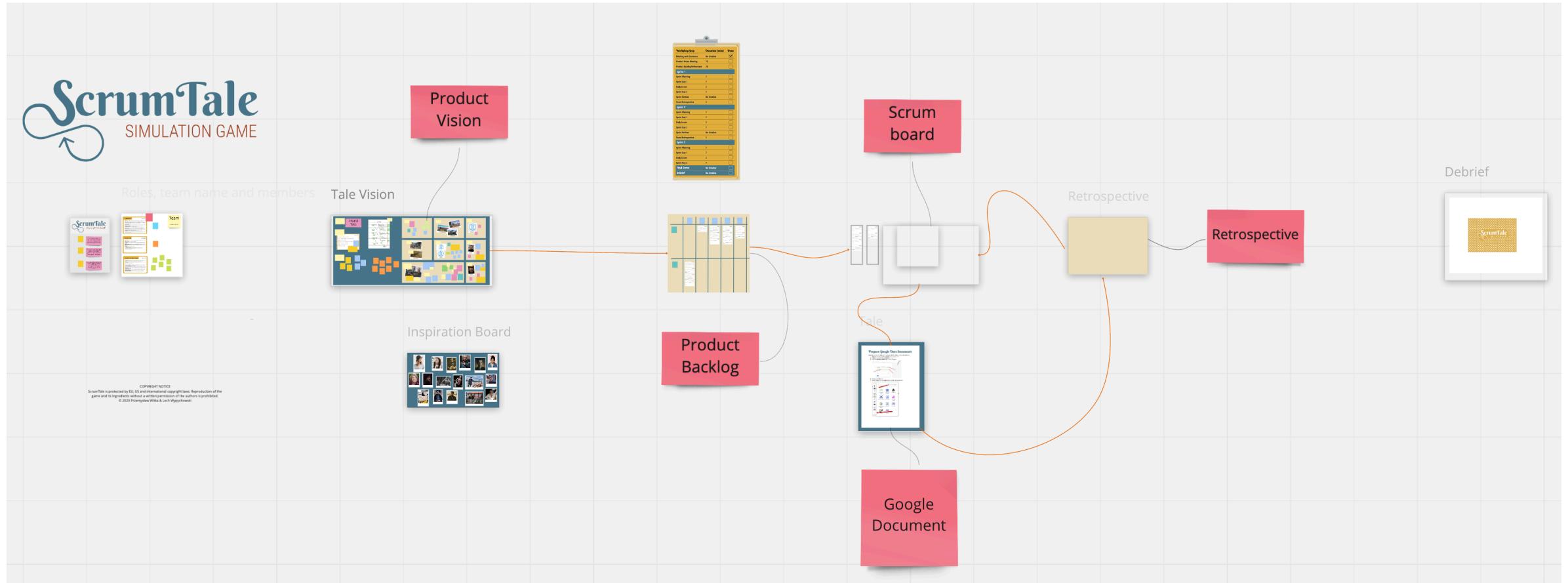
Workshop Step	Duration (min)	Done
Meeting with Customer	No timebox	<input checked="" type="checkbox"/>
Product Vision Meeting	15	<input type="checkbox"/>
Product Backlog Refinement	20	<input type="checkbox"/>
<b>Sprint 1</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Sprint Review	No timebox	<input type="checkbox"/>
Team Retrospective	5	<input type="checkbox"/>
<b>Sprint 2</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Sprint Review	No timebox	<input type="checkbox"/>
Team Retrospective	5	<input type="checkbox"/>
<b>Sprint 3</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Final Demo	No timebox	<input type="checkbox"/>
Debrief	No timebox	<input type="checkbox"/>

## Geschichten und Teams

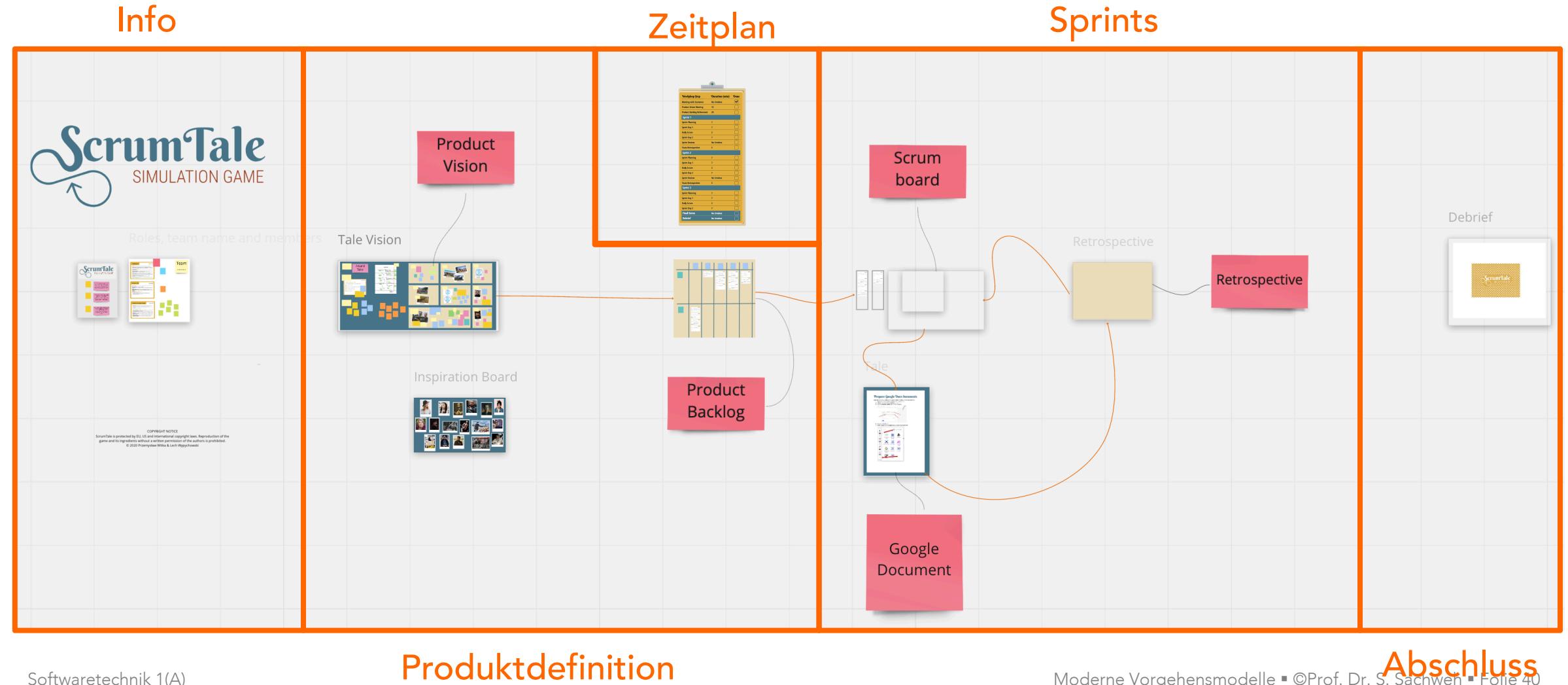
- Das Ziel ist es, **eine Detektiv-Geschichte** zu schreiben.
- Sie können sich die Geschichte wie eine einzelne unabhängige Sherlock Holmes-Geschichte vorstellen.
- Jede Geschichte wird von einem einzelnen Team unter Verwendung eines **Miro-Boards** und eines separaten **Google-Dokuments** geschrieben.



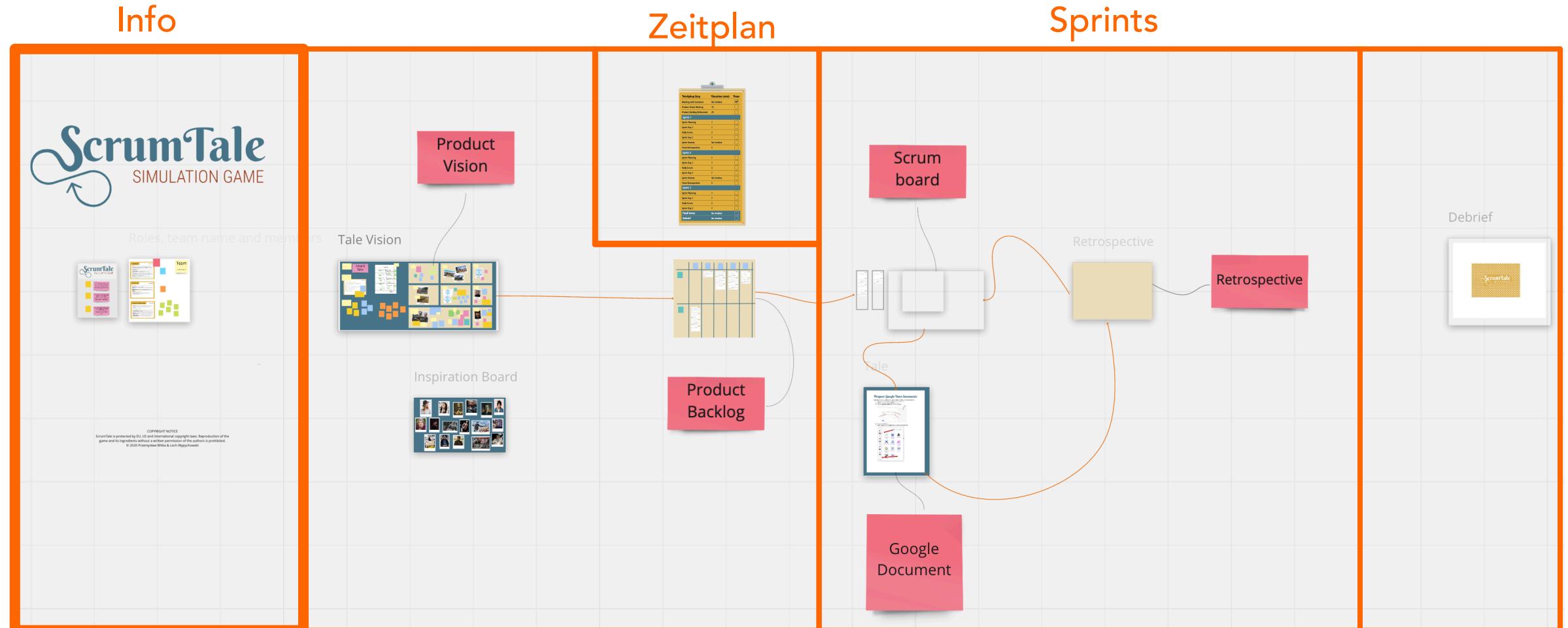
# Wie ist das Miro-Board aufgebaut?



# Wie ist das Miro-Board aufgebaut?



# Wie ist das Miro-Board aufgebaut?



Intro

Info

# ScrumTale

SIMULATION GAME

Disclaimer

ScrumTale is not a game per se. Teams will not compete with each other. It's designed for experiential learning purposes.

Workshop Goal

The goal of this simulation game is to write the greatest crime story ever using Scrum Framework

Product

Your product will be a Tale. The Tale is like an independent Sherlock Holmes story, written by your team, using a Google document for parallel writing.

Roles, team name and members

Tutors are Product Owners in the ScrumTale

John

Team  
.....

## Product Owner (PO)

**Tale Vision:** You are responsible for translating the initial vision into a single Tale and making sure that the Tale is consistent with the user needs. You will work closely with the other Product Owners and the other POs on the product backlog of the single Tale. You are responsible for setting priorities and clarifying user stories. Be open to the Development Team's suggestions. Use blank Story Cards to define user stories. Focus on MVR and quality. No all user stories need to be implemented.

**Product Backlog:** The Product Owner is responsible for a user story backlog of the product backlog of the single Tale. You are responsible for setting priorities and clarifying user stories. Be open to the Development Team's suggestions. Use blank Story Cards to define user stories. Focus on MVR and quality. No all user stories need to be implemented.

**Product Vision meeting:** Work with CPO and the Customer on the back and Tale definition. "Back Vision" and "Tale Vision" Bp charts will help you in this process.

**Backlog Refinement meeting:** You are responsible for the refined Product Backlog.

**During the sprint:** You are responsible for the user stories in the product backlog. You are responsible for the sprint planning, sprint review and the user stories in the product backlog.

**Sprint Review meeting:** Developers Team present product increments. You are responsible for verifying delivered stories against Acceptance Criteria. If the Developers Team does not accept Story Cards in "Sprint" column, after rejected user stories go back to the Product Backlog. If the Developers Team accepts Story Cards, pay attention to product consistency, quality and business value. Make sure that all deliverables and gaps are reflected in the product backlog.

**Sprint Retrospective:** Share your observations regarding the impact of the development process of the product.

In "One Product Owner" version of the game, i.e. "One PO" and, you will act as both PO and CPO. During "Teaching Refinement" and "Sprint Planning" meetings, you will be responsible for the user stories and answer the Developers Team to make decisions about product implementation details so that you can focus on the vision at high level requirements.

## Scrum Master (SM)

**SM in ScrumTale:** Support Product Owners in their roles. Help them to keep backlog healthy. Make sure they prioritize the backlog and set clear goals for the Development Team.

**Focus:** Keep everybody focused - on the business value, sprint goals, etc.

**Communication:** You are responsible for engaged in product development, use your time to observe dynamics of the team. Ensure there is efficient collaboration within the Scrum Team and across both teams. Work closely with the other Scrum Master to improve communication.

**Improvement and adaptation:** Encourage the Scrum Team to spot obstacles and experiments. Make sure impediments are eliminated with creative solutions.

**Motivation:** Motivation is a key for the team. Turn on your radar for disgruntled people and work with them to find out the reasons of motivation.

**Facilitation:** Use your facilitation skills to make meetings effective.

**Your ScrumTale facilitation responsibilities:**

- At the beginning of each sprint you will put the Story Cards from the stack and read it out loud to the team.
- At the end of each sprint, rank it as complete on the game board.
- Help the team to track time during events.

Olivia

## Development Team Member (Developer)

**Developer in ScrumTale:** Scrum team organizes itself to find the best ways to deliver a great product. You are responsible for "how" you build the product. You are ultimately responsible for product quality and continuous improvement process.

**Communication:** You are responsible not only for communication within your team, but also relationships with other teams.

**Engineering practices:** You are responsible for the quality, as a good engineer practices MVR and DMR. Experiment with techniques like code review or pair programming.

**Developer role in ScrumTale meetings:**

**Product Vision meeting:** Each team can design team members for this meeting. They will contribute to the product definition.

**Backlog Refinement meeting:** You share responsibility with PO to clarify product features and priorities. You may propose changes to the backlog.

**Sprint Planning meeting:** Developers Team comment the scope of the Sprint. Miss oriented Story Cards on the "Sprint Backlog". You can propose new user stories if needed. You should also plan living defects found in a previous sprint. Use red post-its for defects.

**During the sprint:** Work hard, have fun and collaborate effectively. Work towards a single user story. Make sure the entire back is coherent and meets Definition of Done.

**Sprint Review meeting:** The Scrum Team representatives read out the whole Tale. Be ready as to answer with all questions, not just comments, etc. Make sure you capture all the defects and gaps created during the meeting. Use red post-its for defects and blank Story Cards.

**Sprint Retrospective meeting:** The Scrum Team is responsible to propose an improvement plan for the next sprint.

## Rollenverteilung

- Es gibt folgende **Rollen** zu besetzen:
  - ein „Product Owner“ → **Tutor**
  - ein „Scrum Master“ → **aus der Praktikumsgruppe**
  - Entwicklungsteams → **Praktikumsgruppe ohne „Scrum Master“**
- Rollenkarten sind auf dem **Miro-Board** hinterlegt; schreiben Sie die Zuordnung der Namen zu den Rollen für ihr Team auf Post-its neben die entsprechenden Rollenkarte.
- Die **Tutoren haben eine Doppelrolle**:
  - Tutoren unterstützen sie als „normaler Tutor“ beim „Spielen“
  - Tutoren springen an den Stellen, wo ein Product Owner benötigt wird, als Product Owner ein.

## Rollen – Product Owner (PO)

- Business Value:** Das Ziel des PO ist es, den Geschäftswert der Lösung für den Kunden zu maximieren. Sie arbeiten mit dem Kunden eng zusammen, um seine Bedürfnisse möglichst genau zu erfassen.
- Product Backlog:** Der PO ist für das Product Backlog (der Geschichte) verantwortlich, Prioritäten zu setzen und User Stories (Tickets) zu klären. Der PO ist offen für die Vorschläge des Entwicklungsteams und verwendet leere Karten, um fehlende User Stories zu definieren. Der Fokus liegt auf dem „Minimum Viable Product (MVP)“ und der Qualität, d.h. Es müssen nicht alle User Stories implementiert werden.

## Rollen – Scrum Master (SM)

**Aufgabe:** Der Scrum Master unterstützt den Product Owner in seiner Rolle und das Team organisatorisch.

**Fokus:** Der SM hält den Fokus des Teams auf die Aufgabe gerichtet und dem Team den Rücken für die inhaltliche Arbeit frei.

**Kommunikation:** Er stellt sicher, dass innerhalb des Scrum-Teams eine effektive Zusammenarbeit umgesetzt wird.

**Inspektion und Adaptation:** Er ermutigt sein Scrum-Team, Blokaden zu erkennen und neue Strategien auszuprobieren, um diese zu beseitigen.

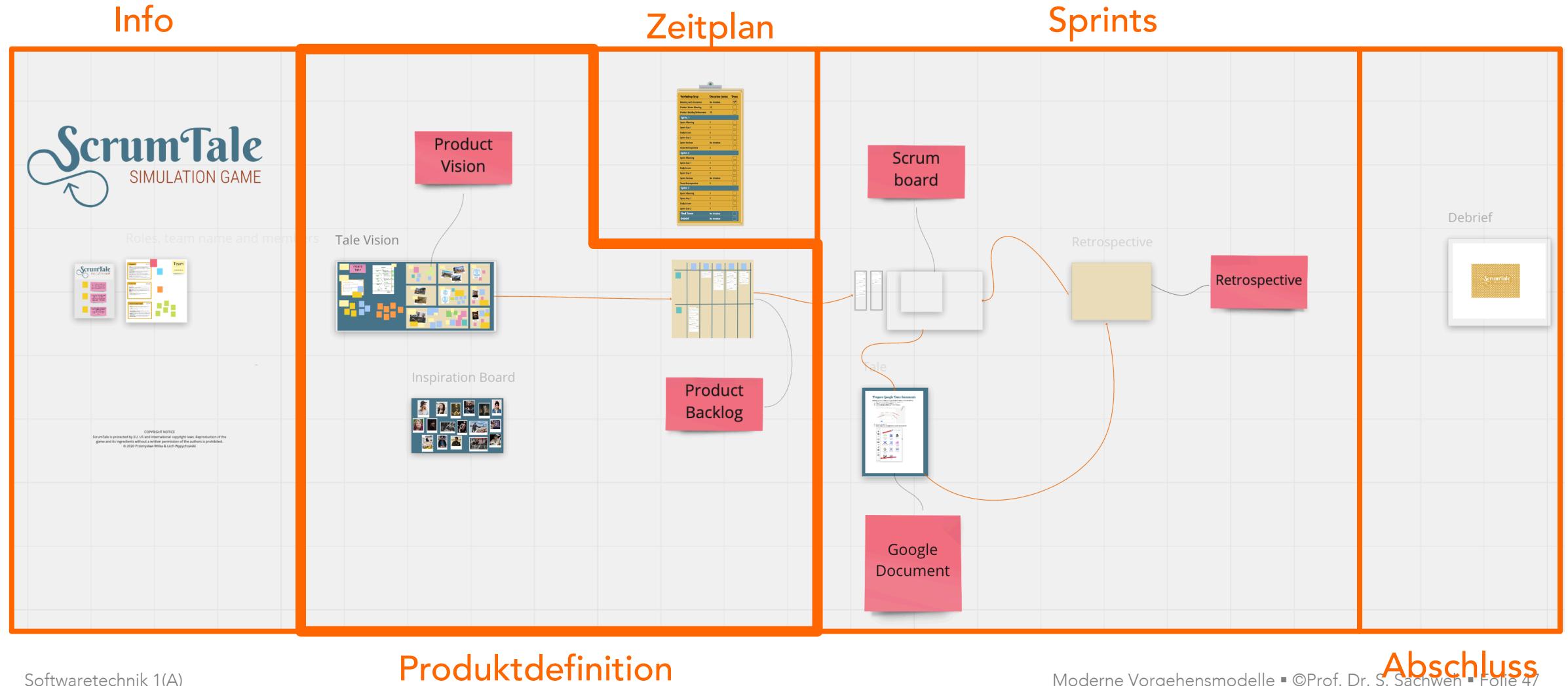
**Motivation:** Motivation ist der Antrieb des Teams. Der Scrum Master achtet auf Leute, die nicht mit Engagement bei der Sache sind.

**Moderation:** Er nutzt seine Moderationskompetenz, um Meetings effektiv umzusetzen.

## Rollen – Development Team

- Aufgabe:** Entwickler konzentrieren sich auf das Produkt und den Geschäftswert, dabei beachten Sie die Vorstellungen und Priorisierungen des Kunden.
- Selbst-organisation:** Sie sind dafür verantwortlich, „wie“ sie das Produkt erstellen. Sie sind letztendlich für die Produktqualität und kontinuierliche Verbesserungen verantwortlich.
- Kommunikation:** Sie tragen durch gute Kommunikation zu einem guten Klima und guter Produktivität im Team bei.
- Fachkenntnisse:** Sie sind für den Inhalt und die Qualität verantwortlich, daher verwenden sie gute fachliche Methoden und Ansätze.

# Wie ist das Miro-Board aufgebaut?



# Produktdefinition – Product Vision

## Definition of Done (DoD)

Each user story introduced in the product increment must fulfill these criteria:

- 1 The Book must be coherent.
- 2 Chronology must be kept along all Plots.
- 3 No grammar and punctuation mistakes.
- 4 Font: Arial, 11pt, black color.

## Abnahmekriterien (engl. Definition of „Done“)

- Die Geschichte muss konsistent sein.
- Die Leser müssen den Eindruck haben, dass die gesamte Geschichte von einer einzelnen Person geschrieben wurde.
- Jede Geschichte muss nach jedem Sprint ein potenziell druckbares Produktinkrement sein.
- Die Geschichte muss folgende Abnahmekriterien erfüllen:
  1. Die Geschichte muss konsistent sein.
  2. Die Chronologie muss in allen Handlungen bewahrt werden.
  3. Keine Grammatik- und Interpunktionsfehler.
  4. Schriftart: Arial 11pt; Schriftfarbe: schwarz.



**Definition of Done (DoD)**

Each user story introduced in the product increment must fulfill these criteria:

- 1 The Book must be coherent.
- 2 Chronology must be kept along all Plots.
- 3 No grammar and punctuation mistakes.
- 4 Font: Arial, 11pt, black color.

Sie können [] Klammern verwenden, um die unfertigen Teile im Text zu markieren. Sie können diese Teile während des Sprint-Reviews überspringen.

## Detektivin

### The Detective



Olivia  
Holmes

Age: 37 y.o

„I can instantly tell when somebody is lying - I can read people like an open book“

Quote

### About

Loves  
her  
work

Always  
wanted to  
become a  
detective

### Life

Moved out  
to Munich  
for  
education

Raised  
by her  
mother

Difficulty to  
build long-  
lasting  
relationships

Don't  
have  
friends

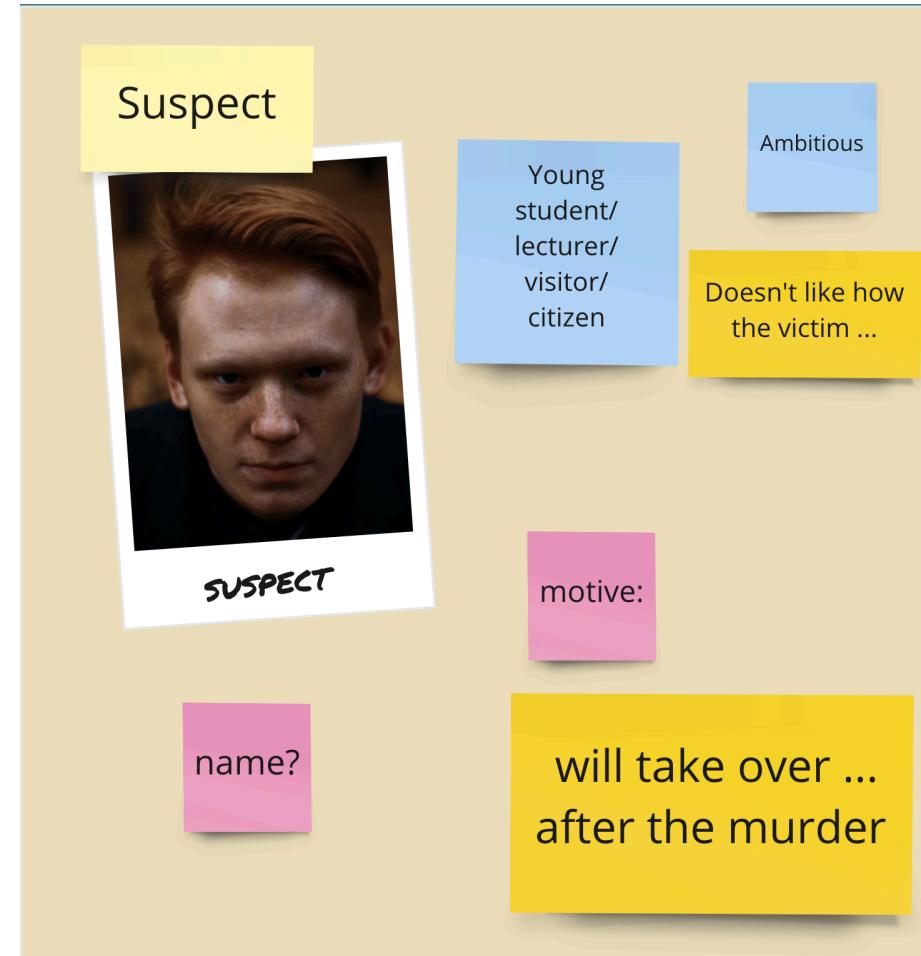
### Work

Shows  
workaholism  
symptoms

Treats men  
instrumentally.

### Privately

## Zeuge und Verdächtiger



## Stadt, Tatort und Tatwaffe

Town



CITY OF DORTMUND



FH DORTMUND

The Crime  
Scene

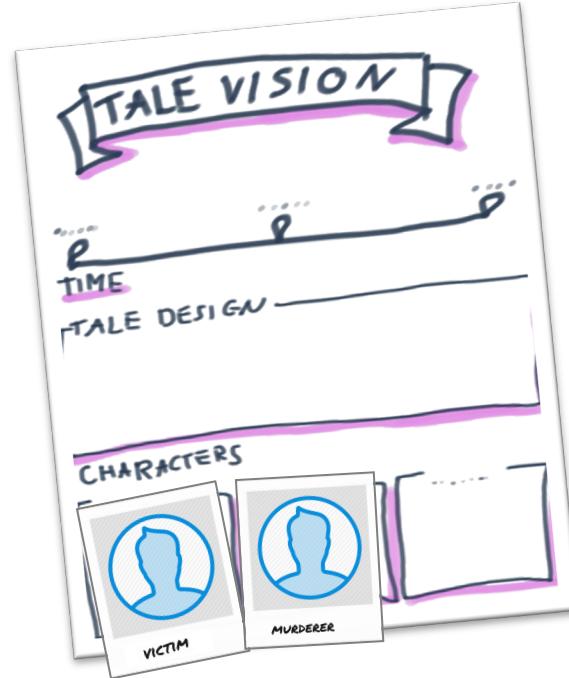


Murder  
weapon



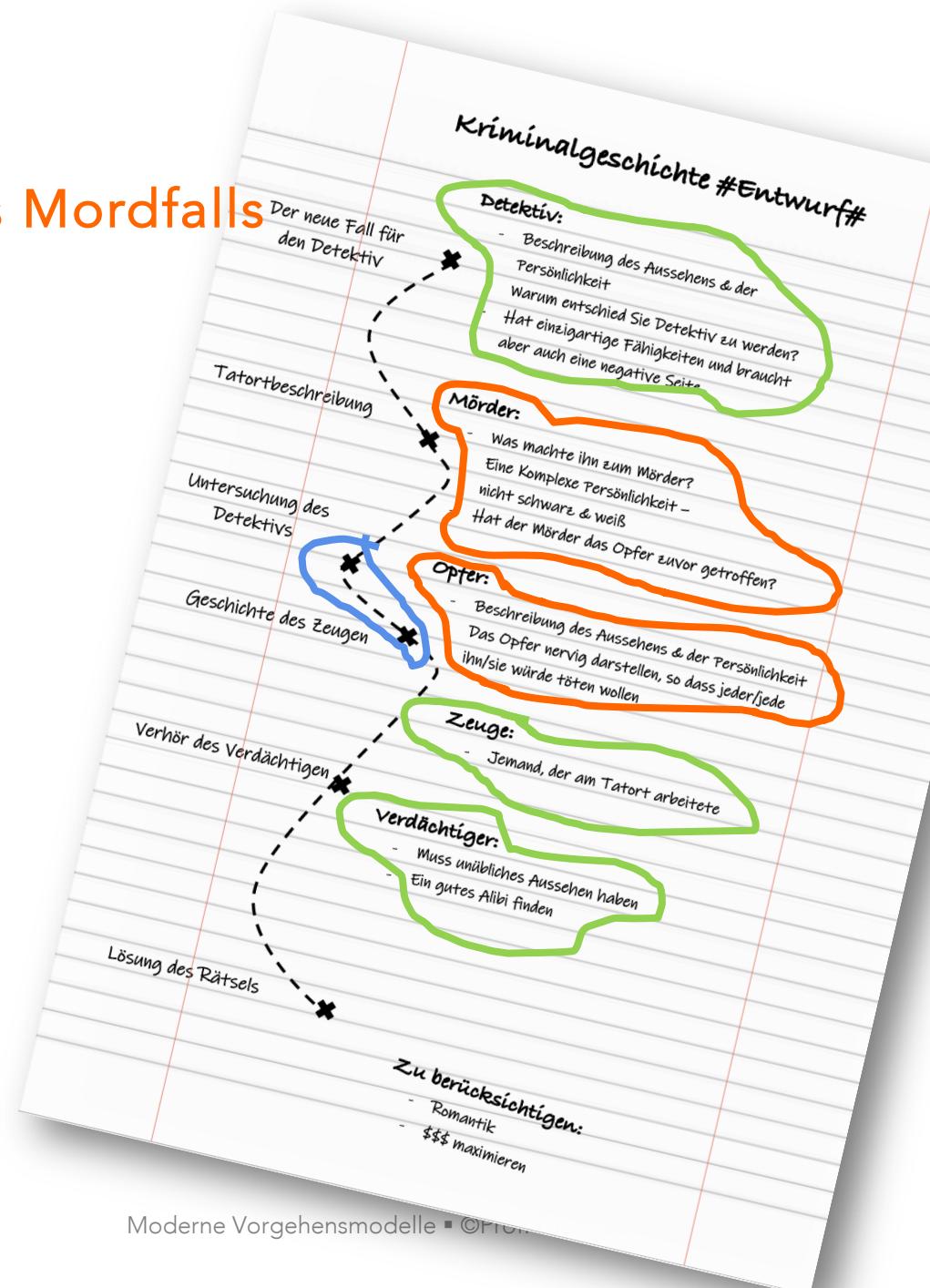
## Mordfall vervollständigen

- Was fehlt ...
  - das Opfer
  - der Mörder
  - das Motiv
  - der Ablauf der Tat
- Hinweis: Am Ende spielt die Schwester des Opfers eine entscheidende Rolle beim Lösen des Falls.
- Titel der Geschichte und in welchem Stil sie geschrieben sein soll.

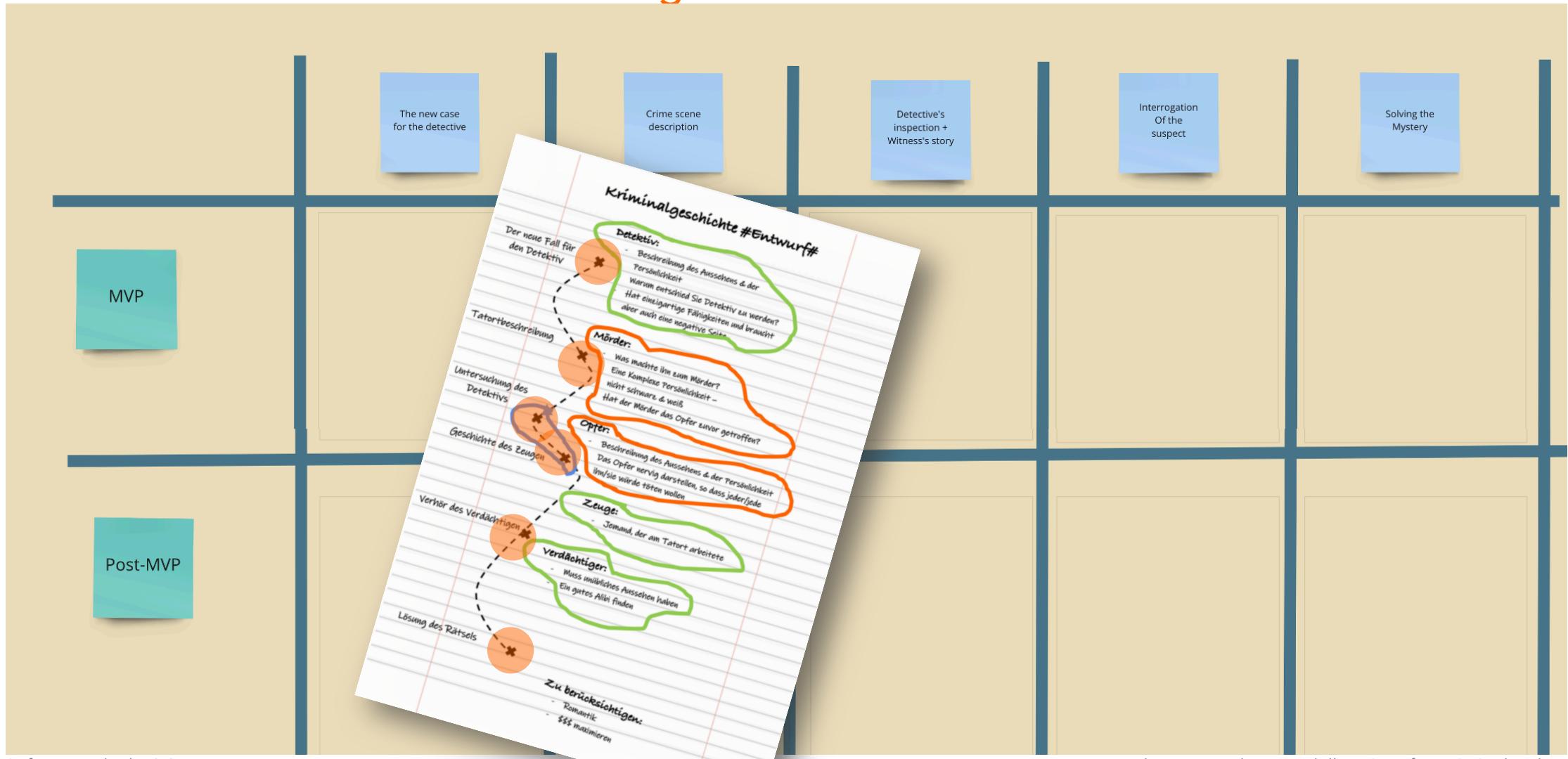


## Die Geschichte – Aufklärung des Mordfalls

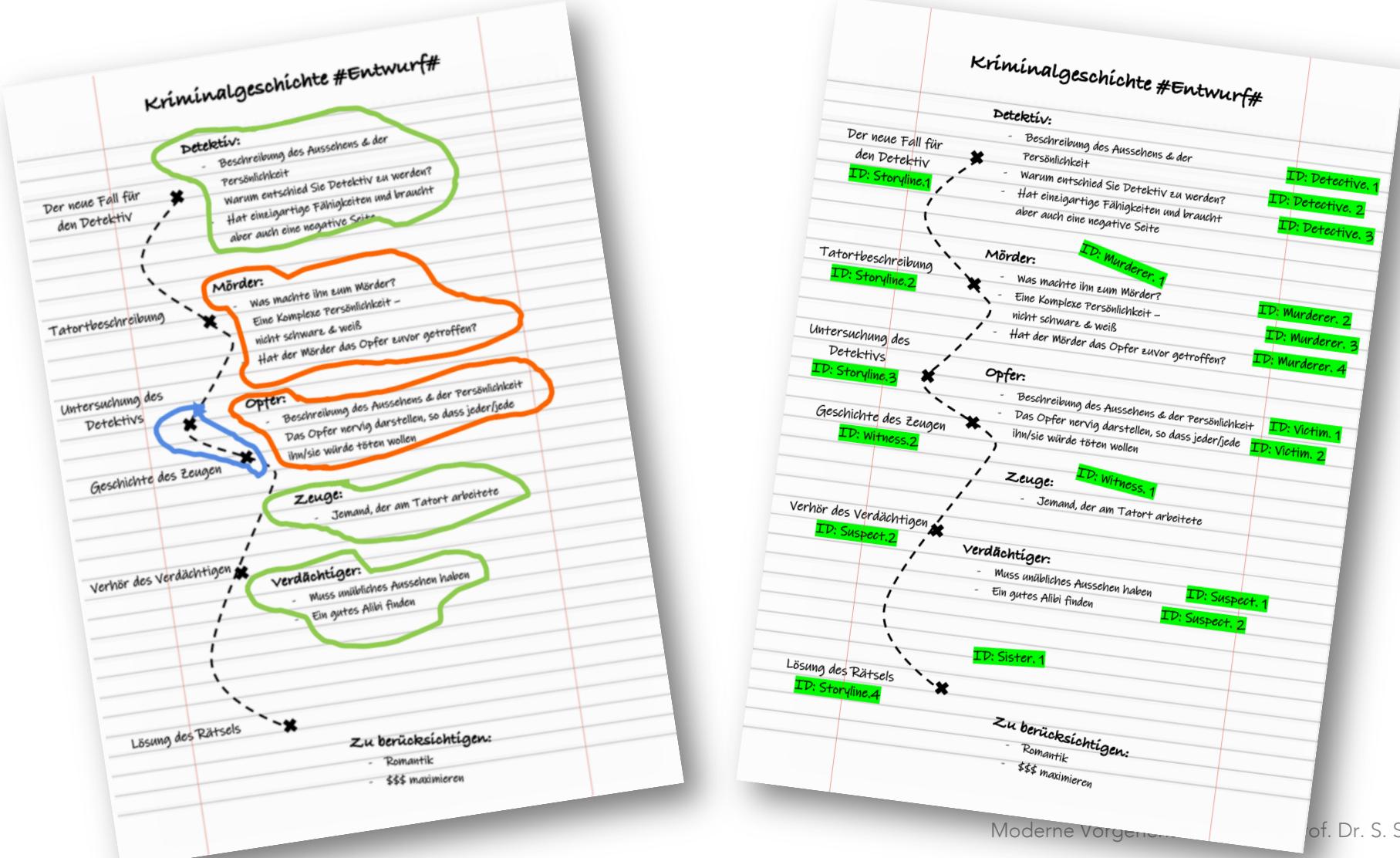
- Ein Glücksfall: ein Autor hat den **Entwurf** einer passenden Geschichte verloren
- Diese Geschichte enthält fünf relevante Szenen:
  - der neue Fall des Detektivs
  - die Tatortbeschreibung
  - die Untersuchung des Detektivs und die Geschichte des Zeugen
  - das Verhör des Verdächtigen
  - die Lösung des Rätsels



# Product Backlog



# Entwurf der Geschichte und User Stories



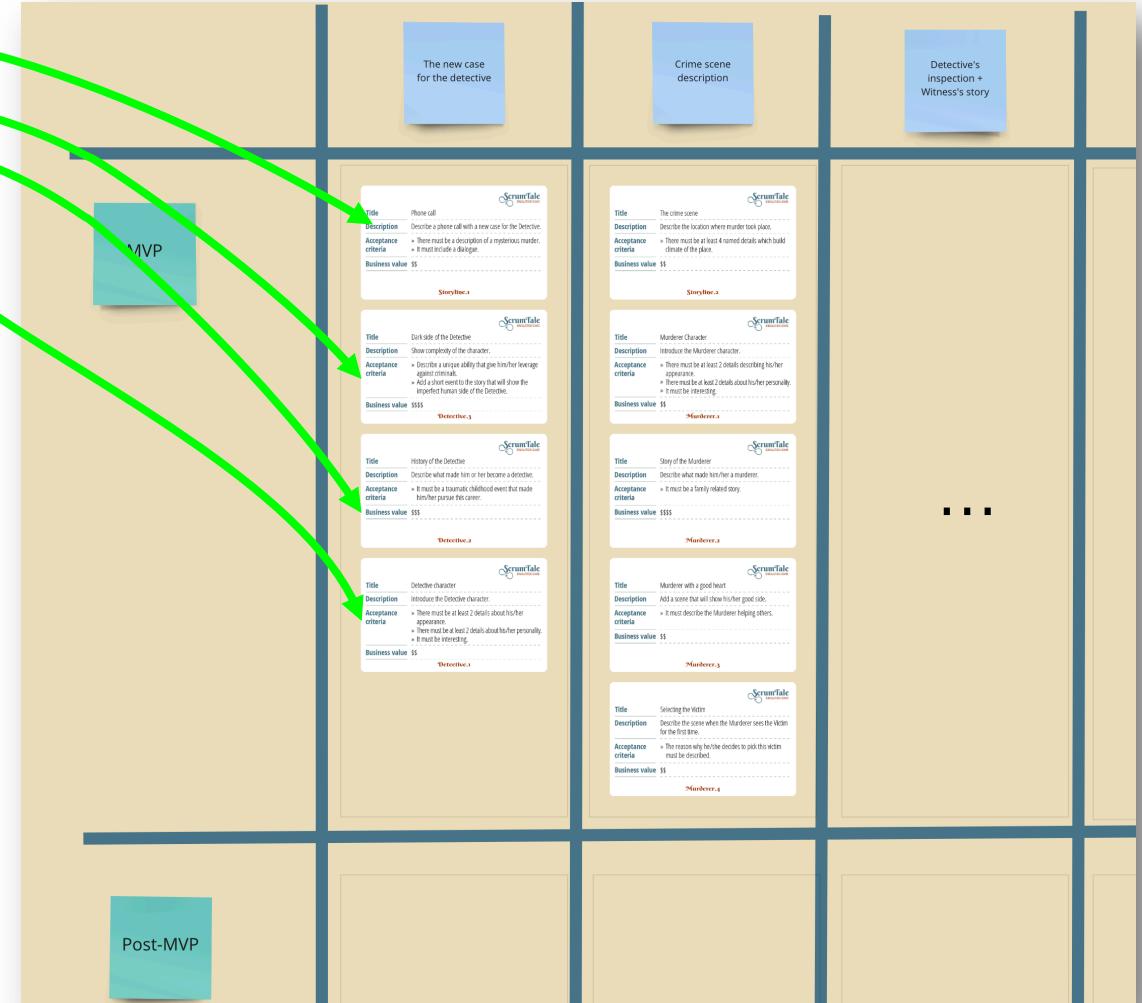
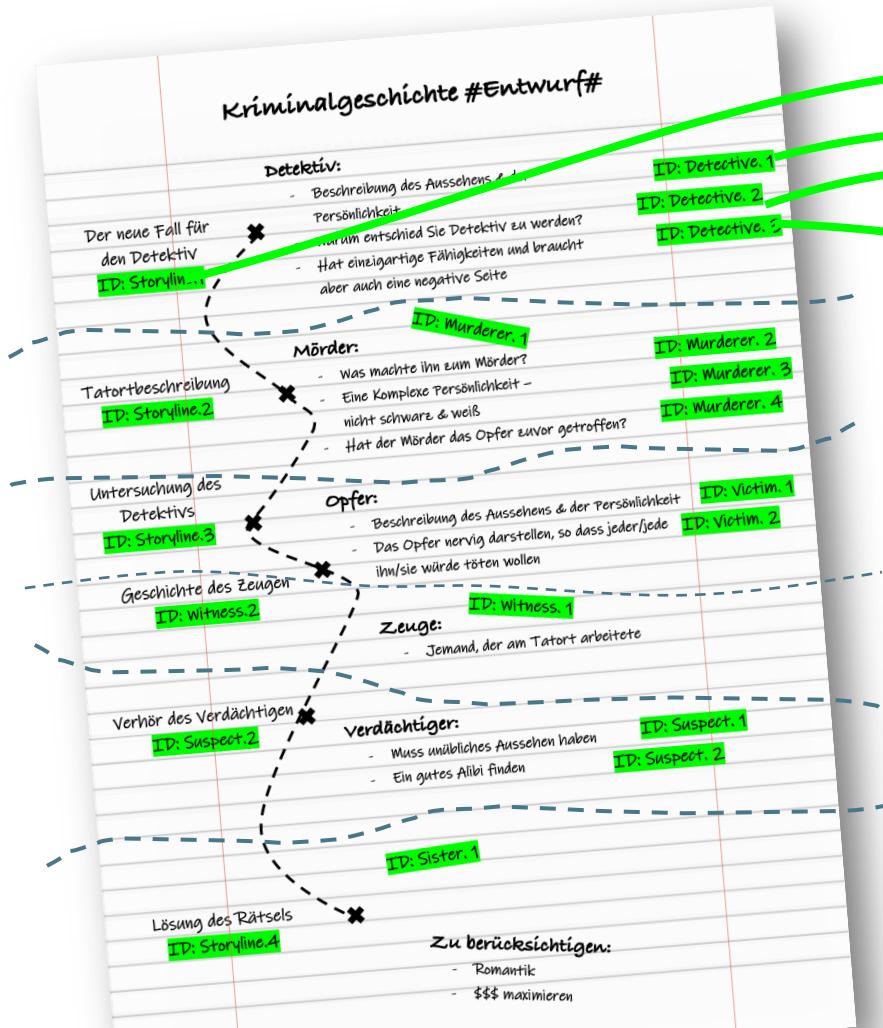
# Hilfsmittel (User Stories) ⇔ 19 Tickets

Storyline	Detective	Murderer	Victim	Witness	Suspect	Sister	
<p><b>Storyline.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Phone call</p> <p><b>Description</b> Describe a phone call with a new case for the Detective.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be a description of a mysterious murder.</li> <li>It must include a dialogue.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Detective.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Detective character</p> <p><b>Description</b> Introduce the Detective character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 details about his/her appearance.</li> <li>There must be at least 2 details about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Murderer.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Murderer character</p> <p><b>Description</b> Introduce the Murderer character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 details about his/her appearance.</li> <li>There must be at least 2 details about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Victim.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Victim character</p> <p><b>Description</b> Introduce the Victim (PO to provide details).</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 attributes describing the character.</li> <li>There must be at least 2 details about his/her personality.</li> <li>White this part from observer (3rd person) perspective.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Witness.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The Witness character</p> <p><b>Description</b> Introduce the Witness.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 details about his/her appearance.</li> <li>Job of the Witness must be related to the murder scene.</li> </ul> </p> <p><b>Business value</b> \$</p>	<p><b>Suspect.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The Suspect character</p> <p><b>Description</b> Introduce the Suspect.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be explained why he/she is suspected of committing the crime.</li> <li>There must be at least 2 details about his/her appearance.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Sister.1</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Victim's Sister</p> <p><b>Description</b> The Detective pays a visit to the Victim's Sister house.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be a description of the location.</li> <li>There must be at least 2 details about her appearance.</li> <li>The Sister tells a story about annoying habit of the Victim.</li> </ul> </p> <p><b>Business value</b> \$</p>	
<p><b>Storyline.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The crime scene</p> <p><b>Description</b> Describe the location where murder took place.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 4 named details which build climate of the place.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Detective.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> History of the Detective</p> <p><b>Description</b> Describe what made him or her become a detective.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be a traumatic childhood event that made him/her pursue this career.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Murderer.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Story of the Murderer</p> <p><b>Description</b> Describe what made him/her a murderer.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be a family-related story.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Victim.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Victim's habit</p> <p><b>Description</b> Describe an irritating habit of the Victim.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be super annoying.</li> </ul> </p> <p><b>Business value</b> \$</p>	<p><b>Witness.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The story of the Witness</p> <p><b>Description</b> Add a dialogue between the Detective and eyewitness.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>The Witness must describe what happened.</li> <li>There must be a weather description.</li> <li>The Witness cannot directly point to the Murderer.</li> <li>The story needs to bring new light to the case.</li> </ul> </p> <p><b>Business value</b> \$\$</p>	<p><b>Suspect.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The Interrogation</p> <p><b>Description</b> Add the Interrogation scene.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must include the Detective and the Suspect.</li> <li>It must include a dialogue.</li> <li>Interrogation room must be described.</li> <li>Detective's evil side must be shown.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Sister.2</b></p> <p><b>ScrumTale</b></p>	
<p><b>Storyline.3</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The Inspection</p> <p><b>Description</b> The Detective visits the crime scene.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>The body must be found in a bizarre arrangement.</li> <li>It must be a detail that caught Detective's attention.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Detective.3</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Dark side of the Detective</p> <p><b>Description</b> Show complexity of the character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>Describe a unique ability that give him/her leverage against criminals.</li> <li>A short event to the story that will show the imperfect human side of the Detective.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>	<p><b>Murderer.3</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Murderer with a good heart</p> <p><b>Description</b> Add a scene that will show his/her good side.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must describe the Murderer helping others.</li> </ul> </p> <p><b>Business value</b> \$</p>			<p><b>Suspect.3</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> The Alibi</p> <p><b>Description</b> Add the Suspects monologue to describe his/her alibi for the time the murder was committed.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be written as the Suspects monologue.</li> <li>Other people need to be involved in the story so that the alibi can be verified.</li> </ul> </p> <p><b>Business value</b> \$</p>		
<p><b>Storyline.4</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Solving the mystery</p> <p><b>Description</b> Detective solves the mystery.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be surprising.</li> <li>Describe how the trail from the murder scene helped to find the Murderer.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p>		<p><b>Murderer.4</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Selecting the Victim</p> <p><b>Description</b> Describe the scene when the Murderer sees the Victim for the first time.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>The reason why he/she decides to pick this victim must be described.</li> </ul> </p> <p><b>Business value</b> \$</p>					

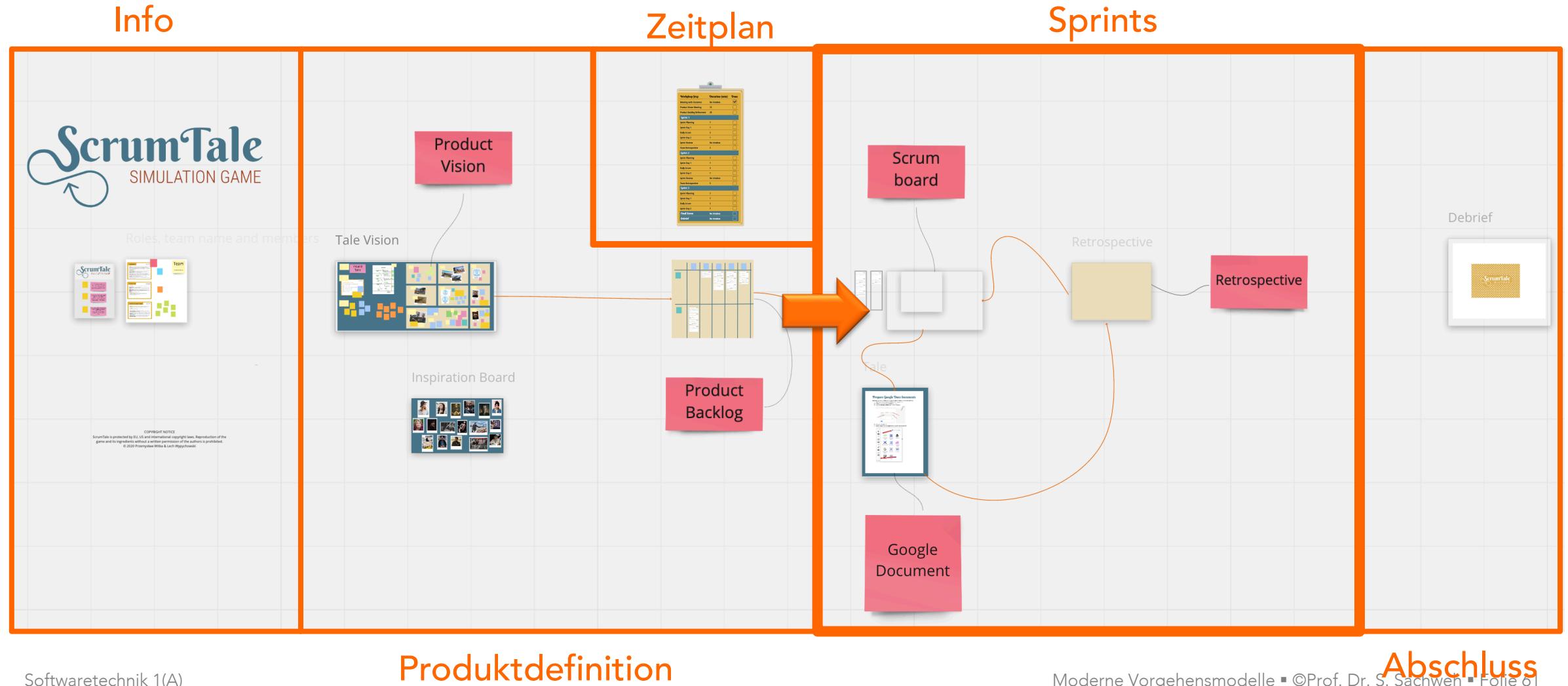
# User Story Card

Storyline	Detective	Murderer	Victim	Witness	Suspect	Sister
<p><b>ScrumTale</b></p> <p><b>Title</b> Phone call</p> <p><b>Description</b> Describe a phone call with a new case for the Detective.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be a description of a mysterious murder.</li> <li>It must include a dialogue.</li> </ul> </p> <p><b>Business value</b> \$\$</p> <p><b>Storyline.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> Detective character</p> <p><b>Description</b> Introduce the Detective character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 details about his/her appearance.</li> <li>There must be at least 2 traits about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p> <p><b>Detective.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> Murderer character</p> <p><b>Description</b> Introduce the Murderer character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 traits about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p> <p><b>Murderer.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> Victim character</p> <p><b>Description</b> Introduce the character of the Victim (to provide details).</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 traits about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p> <p><b>Victim.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> The Witness character</p> <p><b>Description</b> Introduce the Witness.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>There must be at least 2 traits about his/her personality.</li> <li>It must be interesting.</li> </ul> </p> <p><b>Business value</b> \$\$</p> <p><b>Witness.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> The Suspect character</p> <p><b>Description</b> Introduce the Suspect.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>He explains why he/she is suspected of the crime.</li> <li>There must be at least 2 details about his/her appearance.</li> </ul> </p> <p><b>Business value</b> \$</p> <p><b>Suspect.1</b></p>	<p><b>ScrumTale</b></p> <p><b>Title</b> Victim's Sister</p> <p><b>Description</b> The Detective pays a visit to the Victim's Sister house.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>The Detective pays a visit to the Victim's Sister house.</li> <li>There must be at least 2 details about her appearance.</li> <li>The Sister tells a story about annoying habit of the Victim.</li> </ul> </p> <p><b>Business value</b> \$</p> <p><b>Sister.1</b></p>
<p><b>Storyline.2</b></p>	<p><b>Storyline.2</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> History of the Detective</p> <p><b>Description</b> Describe what made him or her become a detective.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>It must be a traumatic childhood event that made him/her pursue this career.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p> <p><b>Detective.2</b></p>	<p><b>Storyline.3</b></p>	<p><b>Storyline.3</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Dark side of the Detective</p> <p><b>Description</b> Show complexity of the character.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>Describe a unique ability that give him/her leverage against criminals.</li> <li>Add a short event to the story that will show the imperfect human side of the Detective.</li> </ul> </p> <p><b>Business value</b> \$\$\$</p> <p><b>Detective.3</b></p>	<p><b>Storyline.4</b></p>	<p><b>Storyline.4</b></p> <p><b>ScrumTale</b></p> <p><b>Title</b> Selecting the Victim</p> <p><b>Description</b> Describe the scene when the Murderer sees the Victim for the first time.</p> <p><b>Acceptance criteria</b> <ul style="list-style-type: none"> <li>The reason why he/she decides to pick this victim must be described.</li> </ul> </p> <p><b>Business value</b> \$</p> <p><b>Murderer.4</b></p>	<p><b>Storyline.5</b></p>
<p><b>ScrumTale</b></p> <p><b>Titel</b> Hintergrund der Detektivin</p> <p><b>Beschreibung</b> Was hat dazu geführt, daß sie Detektivin wurde</p> <p><b>Akzeptanz-kriterien</b> » Es muss eine traumatische Kindheitserfahrung sein, die sie dazu brachte, diese Karriere einzuschlagen.«</p> <p><b>Geschäftswert</b> \$\$\$</p> <p><b>Detective.2</b></p>						

# Von der „Product Vision“ zum „Product Backlog“



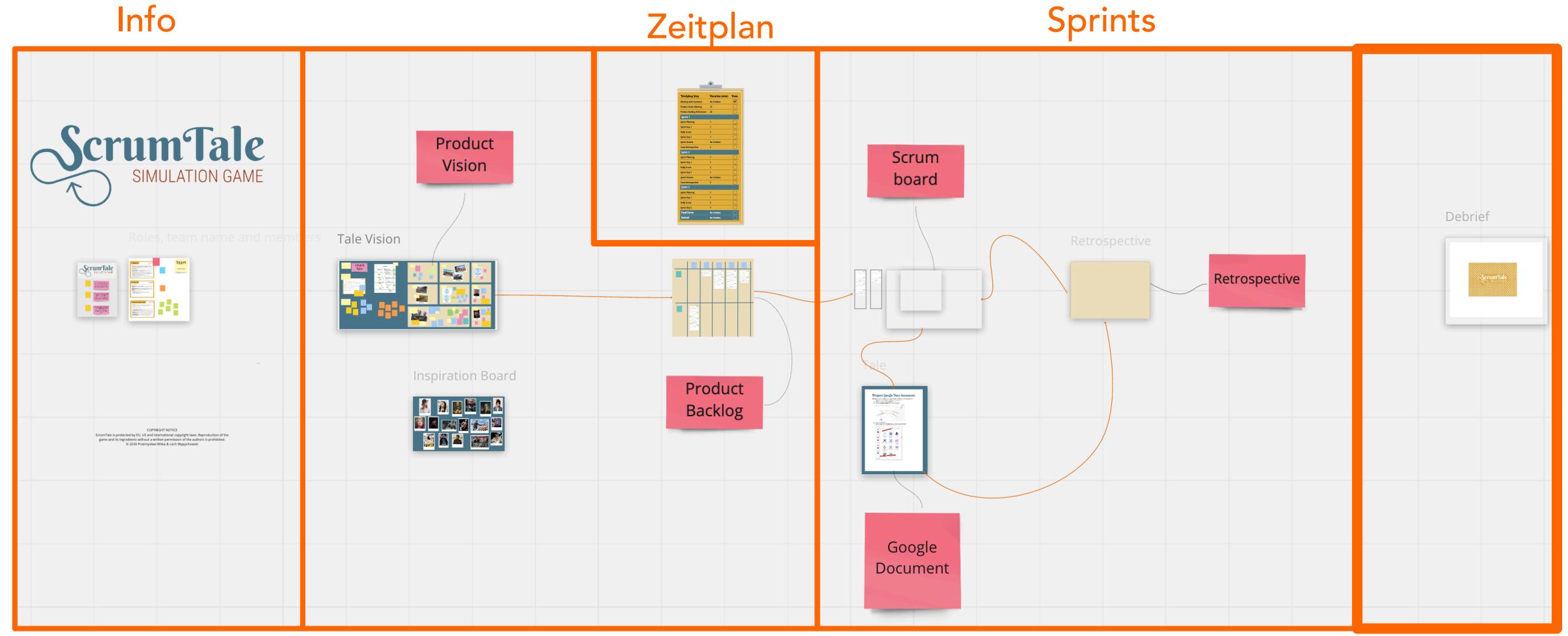
# Wie ist das Miro-Board aufgebaut?



# Von der „Product Vision“ zum „Product Backlog“



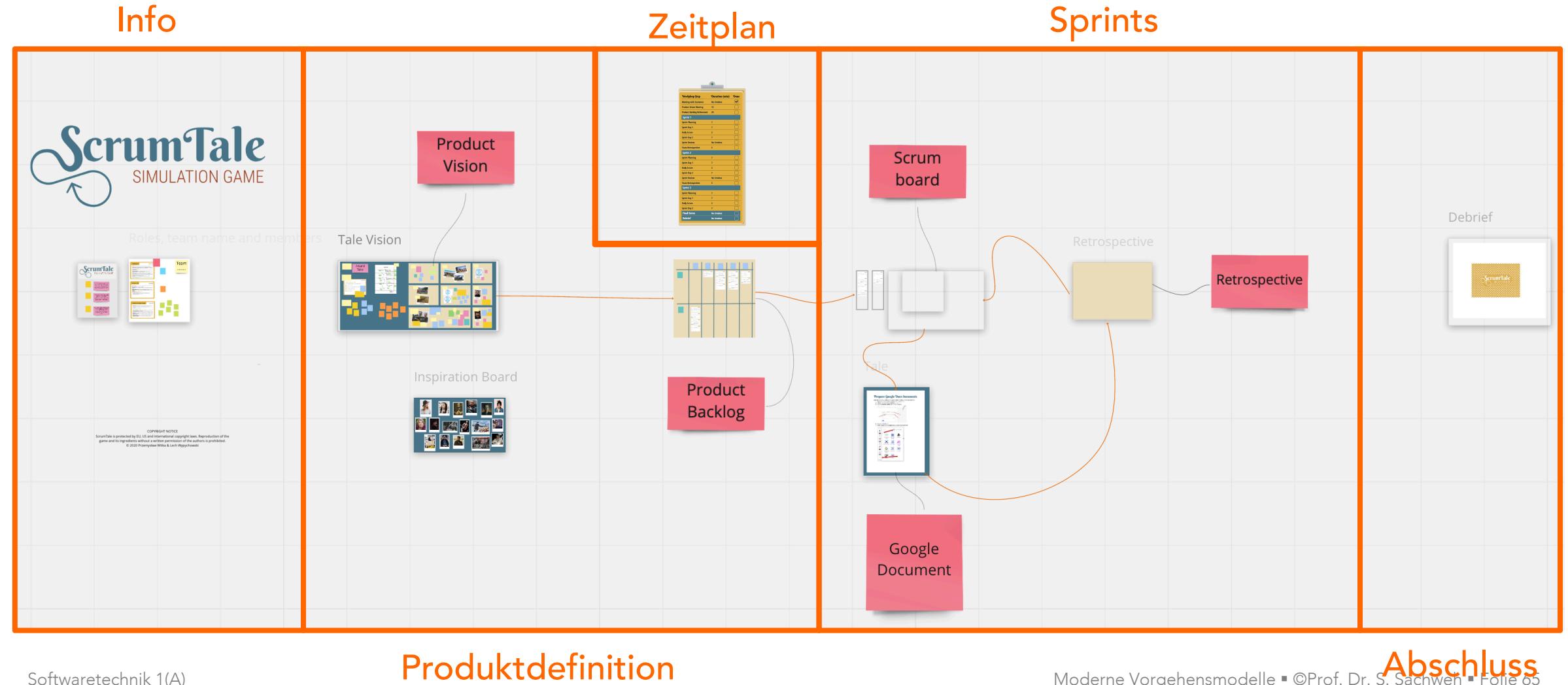
# Wie ist das Miro-Board aufgebaut?

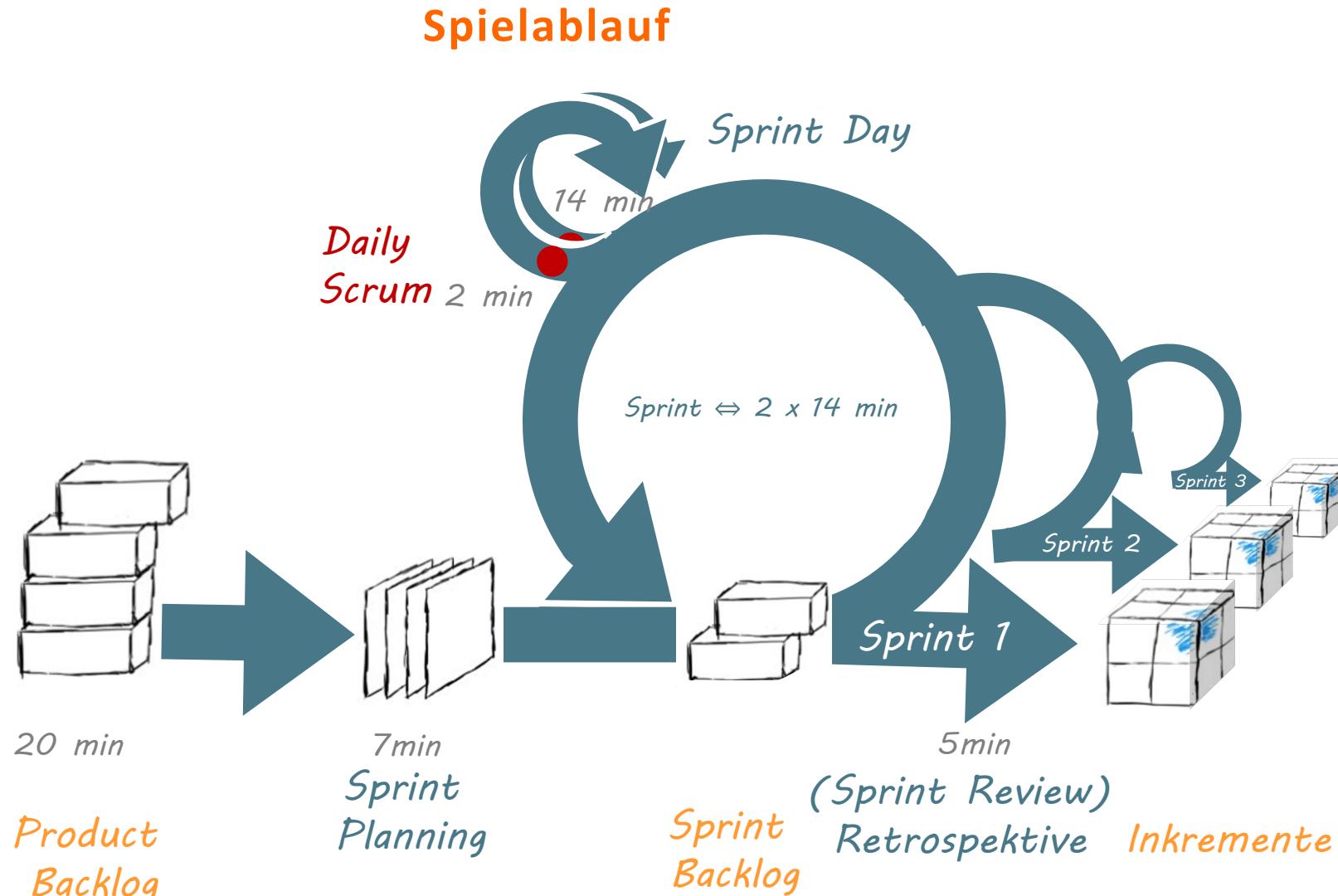


## Nachbesprechung (engl. Debriefing)

- Wurden die Kundenanforderungen erfüllt?
- War das Produkt am Ende jedes Sprints „potentiell versandfähig“?
- Wie wichtig waren die Sprints für den letztendlichen Erfolg?
- Wie hat sich diese Organisationsstruktur auf das Produkt und die Kommunikation ausgewirkt?
- Hatte das Scrum-Team alle Kompetenzen und Informationen, die für eine effiziente Arbeit erforderlich waren?
- Wie hat sich die Produktqualität im Laufe des Prozesses verändert?
- Wie beeinflusst Scrum die Kommunikation und Zusammenarbeit?
- Wie helfen Iterationen und Demonstrationen, das gewünschte Endprodukt zu liefern?
- Wie funktioniert die Priorisierung in Scrum?
- Haben die Teams das geliefert, was sie während der Sprintplanung zugesagt haben?
- Wie war die Motivation im Team.
- Welchen Einfluss hatten die Sprint Retrospektiven auf die Entwicklung?
- Gab es einen Bedarf an Design und Dokumentation?
- Gab es unmotivierte Teammitglieder? Was haben Sie dagegen getan?

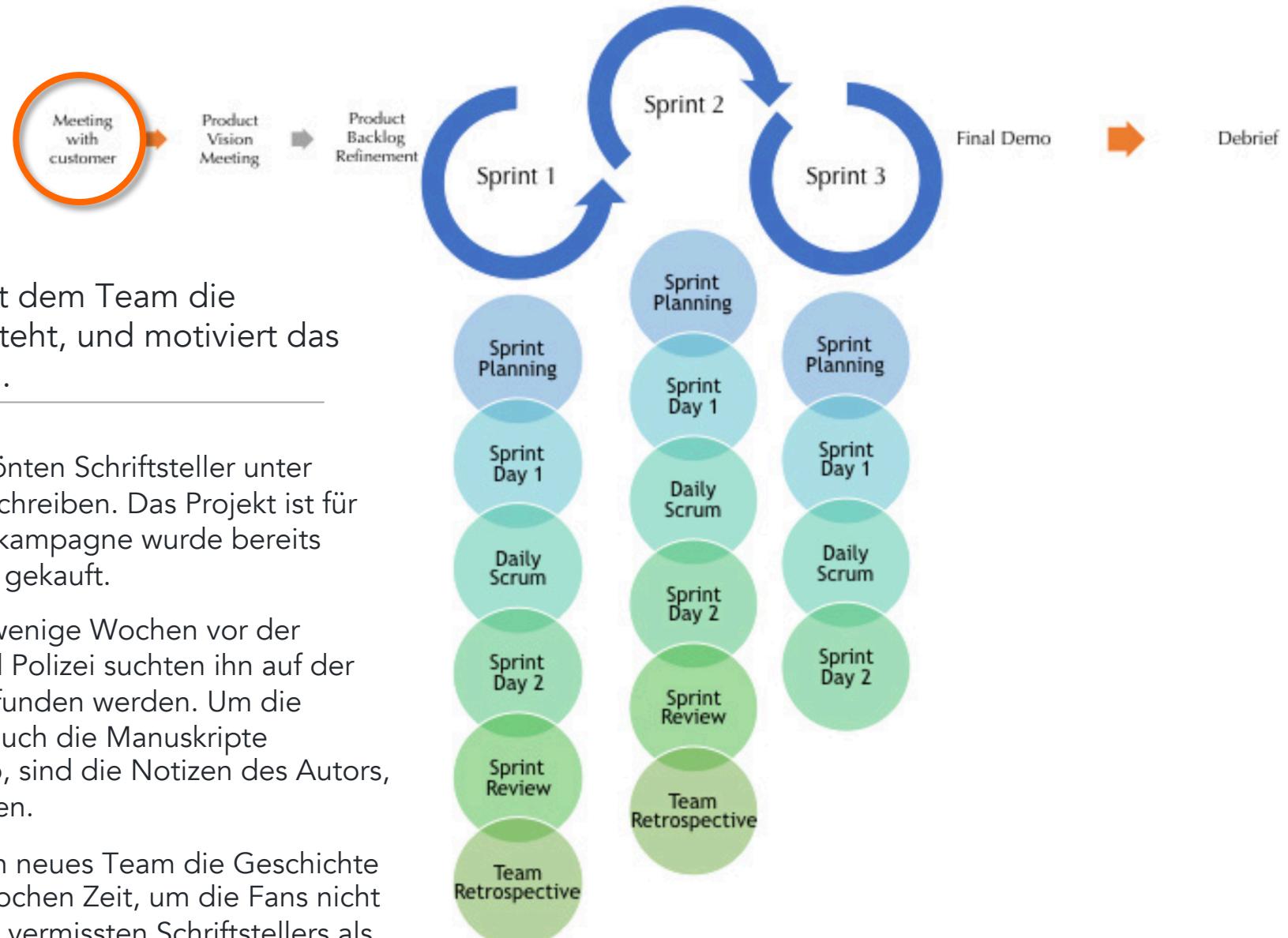
# Wie ist das Miro-Board aufgebaut?





Workshop Step	Duration (min)	Done
Meeting with Customer	No timebox	<input checked="" type="checkbox"/>
Product Vision Meeting	15	<input type="checkbox"/>
Product Backlog Refinement	20	<input type="checkbox"/>
<b>Sprint 1</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Sprint Review	No timebox	<input type="checkbox"/>
Team Retrospective	5	<input type="checkbox"/>
<b>Sprint 2</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Sprint Review	No timebox	<input type="checkbox"/>
Team Retrospective	5	<input type="checkbox"/>
<b>Sprint 3</b>		
Sprint Planning	7	<input type="checkbox"/>
Sprint Day 1	14	<input type="checkbox"/>
Daily Scrum	2	<input type="checkbox"/>
Sprint Day 2	14	<input type="checkbox"/>
Final Demo	No timebox	<input type="checkbox"/>
Debrief	No timebox	<input type="checkbox"/>

## Events



### Meeting with the customer

Der Kunde (CEO des Verlags) erläutert dem Team die Herausforderung, vor der der Verlag steht, und motiviert das Team sich für den Auftrag einzusetzen.

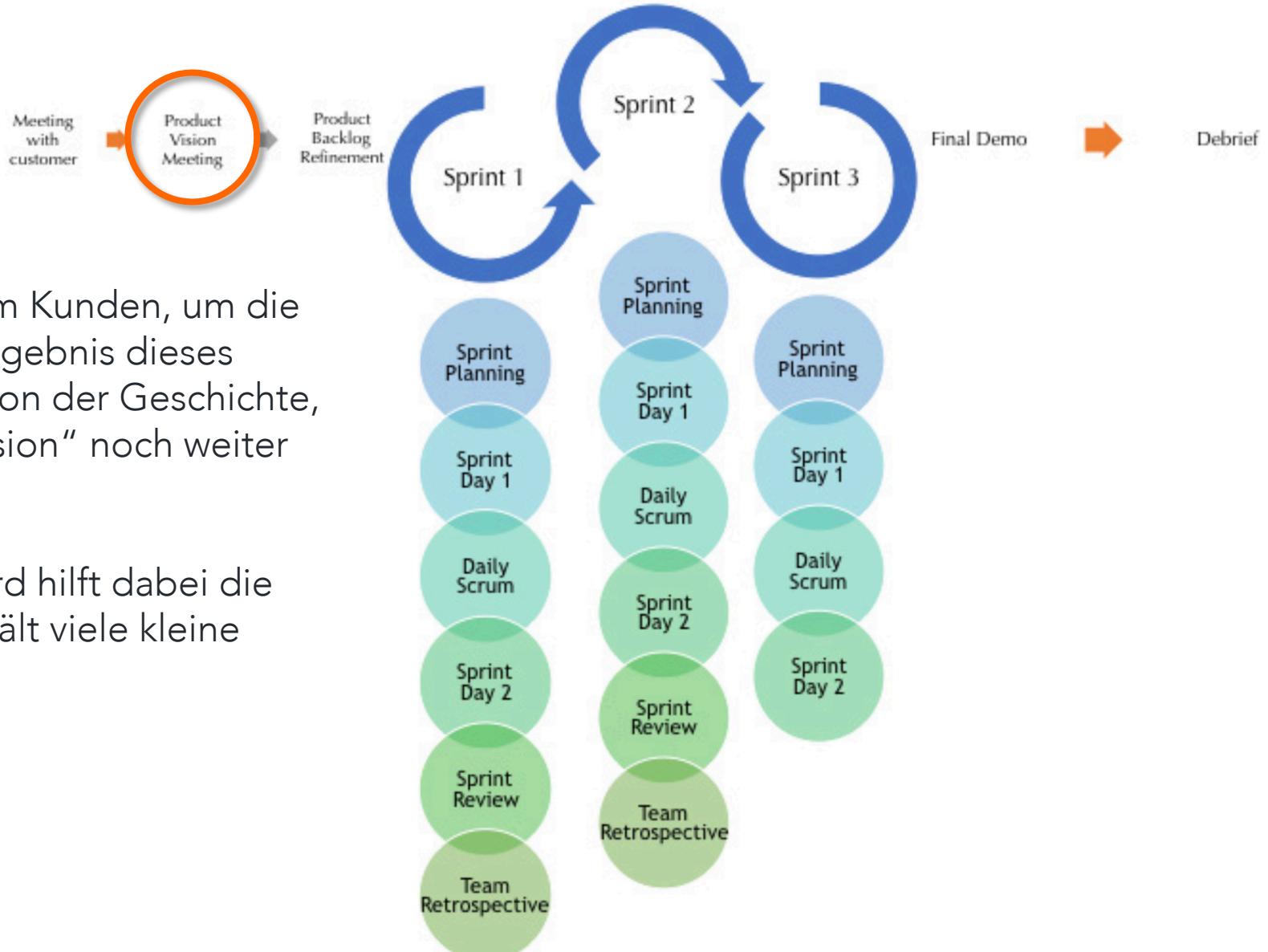
### Die Story

Der Verlag hat einen berühmten, preisgekrönten Schriftsteller unter Vertrag, um eine neue Krimigeschichte zu schreiben. Das Projekt ist für den Verlag sehr wichtig und die Marketingkampagne wurde bereits gestartet. Ein großes Filmstudio hat Rechte gekauft.

Alles verlief reibungslos, bis der Autor nur wenige Wochen vor der Buchpremiere vermisst wurde. Freunde und Polizei suchten ihn auf der ganzen Welt, aber er konnte nirgendwo gefunden werden. Um die Situation noch schlimmer zu machen, sind auch die Manuskripte verschwunden. Das einzige, was übrig blieb, sind die Notizen des Autors, die das Grundgerüst des Buches beschreiben.

Um einen Skandal zu vermeiden soll nun ein neues Team die Geschichte übernehmen. Das Team hat nur ein paar Wochen Zeit, um die Fans nicht zu enttäuschen und hat nur die Notizen des vermissten Schriftstellers als Ausgangsbasis.

## Events

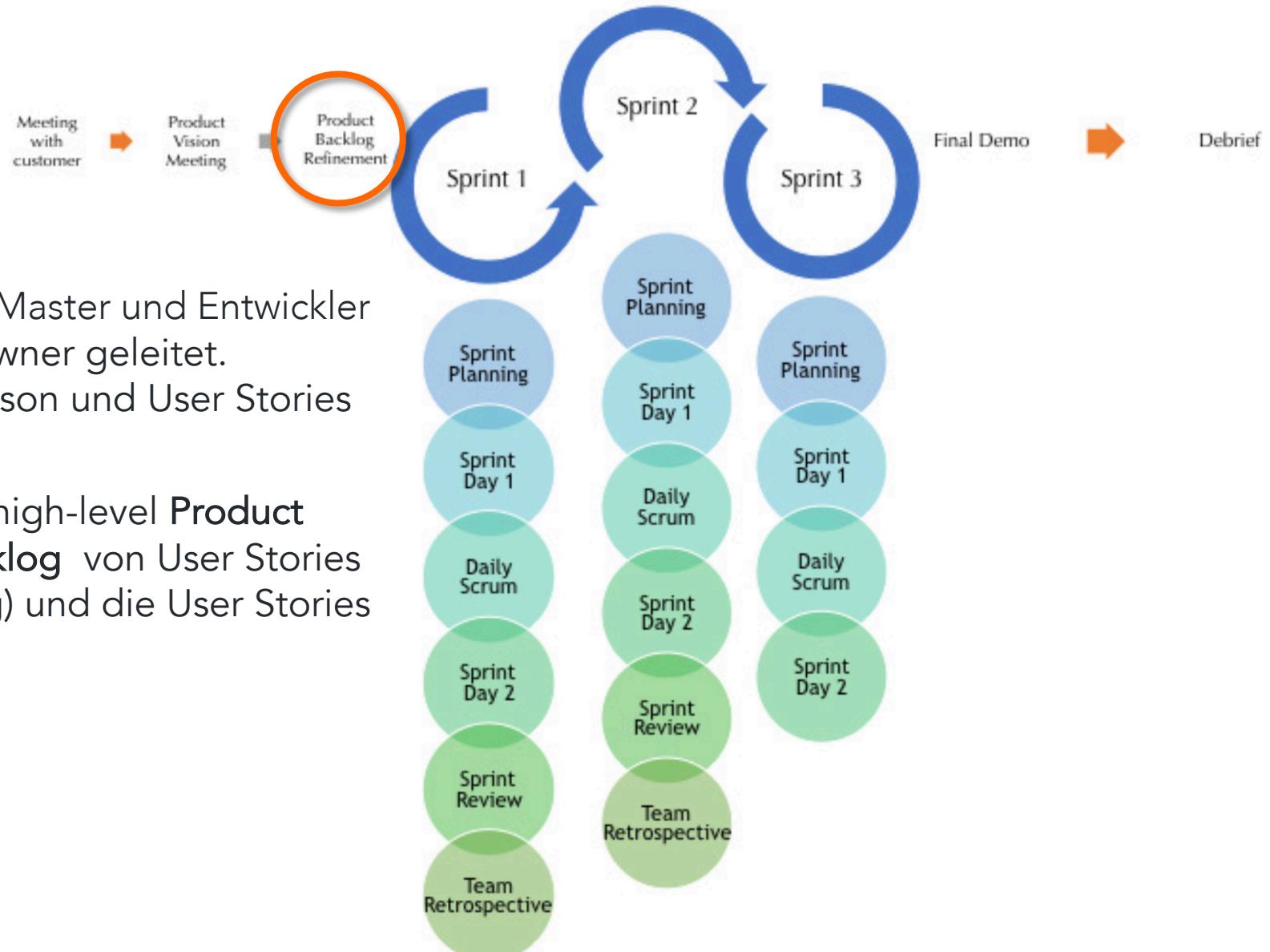


### ■ Product Vision Meeting

Der Product Owner trifft sich mit dem Kunden, um die Produktvision zu besprechen. Das Ergebnis dieses Treffens ist eine (unvollständige) Vision der Geschichte, die er mit dem Team in der „Tale Vision“ noch weiter ausarbeitet.

Die „Tale Vision“ auf dem Miro Board hilft dabei die Vision weiter zu entwickeln und enthält viele kleine Hilfsmittel und Tipps.

## Events



### Product Backlog Refinement

Teilnehmer: Product Owner, Scrum Master und Entwickler

Diese Meeting wird vom Product Owner geleitet.

Eigentlich würde hier die Product Vison und User Stories dem Team vorgestellt werden.

Während dieses Meetings wird die high-level **Product Vision** der Geschichte, in einen **Backlog** von User Stories umgewandelt (vgl. Product Backlog) und die User Stories darin priorisiert.

## Events

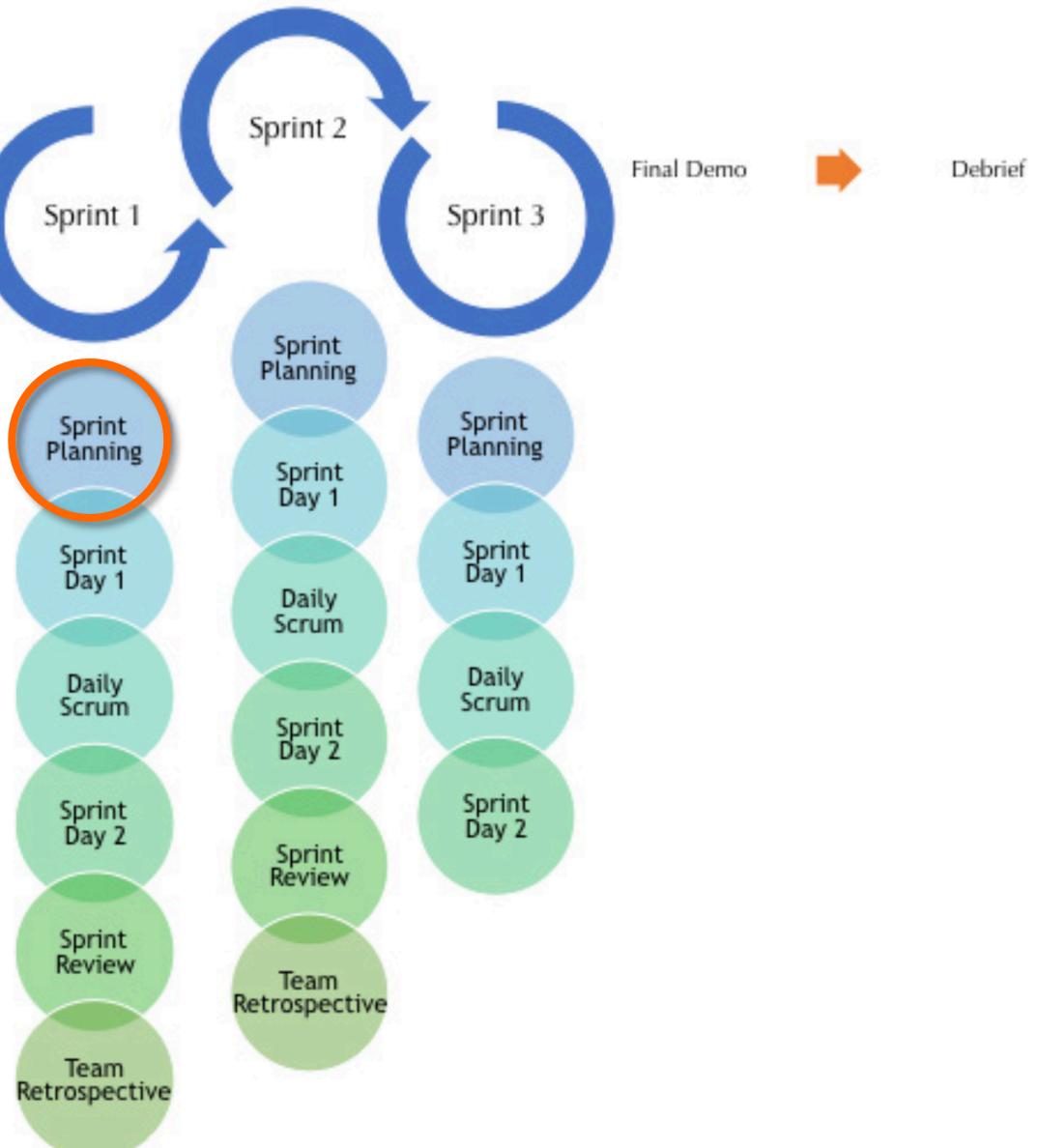


### Sprint Planning

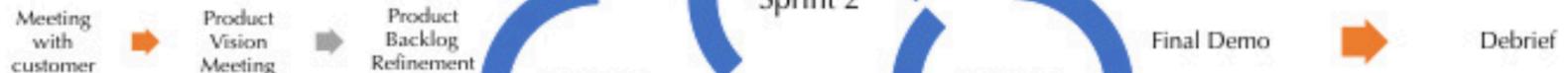
Teilnehmer: Product Owner, Scrum Master und Entwickler

Das Event wird vom Scrum Master organisiert und teilt sich in der Praxis meist in zwei Phasen auf: „Warum“ und „Was“ – also darum, „warum“ der Sprint einen Wert besitzt und „was“ im Sprint getan werden soll.

- **Warum-Teil (mit Product Owner)**
  - Der Product Owner schlägt vor, mit welchem Schwerpunkt im aktuellen Sprint der Wert und Nutzen des Produktes gesteigert könnte. Gemeinsam wird dann das **Sprint-Ziel** vereinbart, das eine Etappe auf dem Weg zum Produkt-Ziel darstellt.
  - Nennung der User Stories – als Selected Backlog – durch den Product Owner, die er mit diesem Ziel verbindet und anschließende Diskussion mit Entwicklern, um die Machbarkeit abzuschätzen.
  - Aufwandschätzung zur Realisierung der User Stories.
  - Füllen des Sprint Backlogs unter Beachtung dessen, was zeitlich geschafft werden kann.
  - Commitment (also die Verpflichtung) der Entwickler das Sprint-Ziel gemeinsam zu erreichen und die besprochenen Inhalte im Laufe des Sprints zu realisieren.



## Events

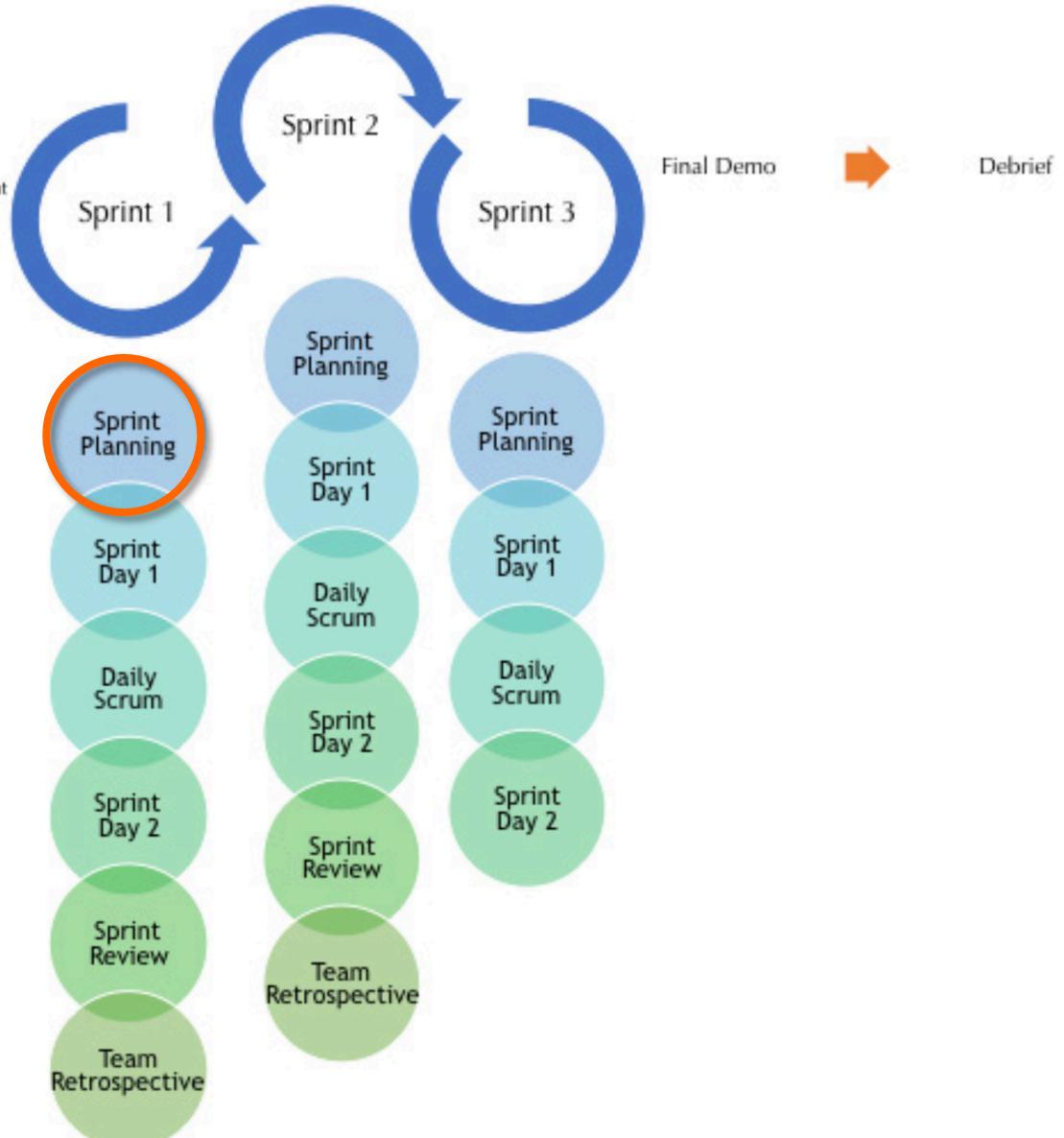


### Sprint Planning

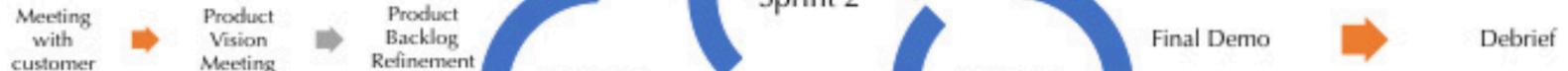
Teilnehmer: Product Owner, Scrum Master und Entwickler

Das Event wird vom Scrum Master organisiert und teilt sich in der Praxis meist in zwei Phasen auf: „Warum“ und „Was“ – also darum, „warum“ der Sprint einen Wert besitzt und „was“ im Sprint getan werden soll.

- **Was-Teil (ohne Product Owner)**
  - Visualisierung des Bearbeitungsstandes der Karten auf eine Taskboard oder Kanban-Board.
  - Entwicklung einer Realisierungsstrategie.
  - Am Ende dieses Teils ist das Team in der Lage, dem Product Owner und dem Scrum Master zu erklären, wie es als selbstorganisiertes Team arbeiten will, um das Sprint Goal zu erreichen und das erwartete Inkrement zu erstellen.



## Events



### Daily Scrum (Meeting)

Dieses Meeting wird vom Scrum Master organisiert und soll die Zusammenarbeit innerhalb des Entwicklungsteams unterstützen, wobei im wesentlichen Informationen über die Fortschritte beim Erreichen des Sprintziels und notwendige Unterstützung/Hilfe besprochen wird.

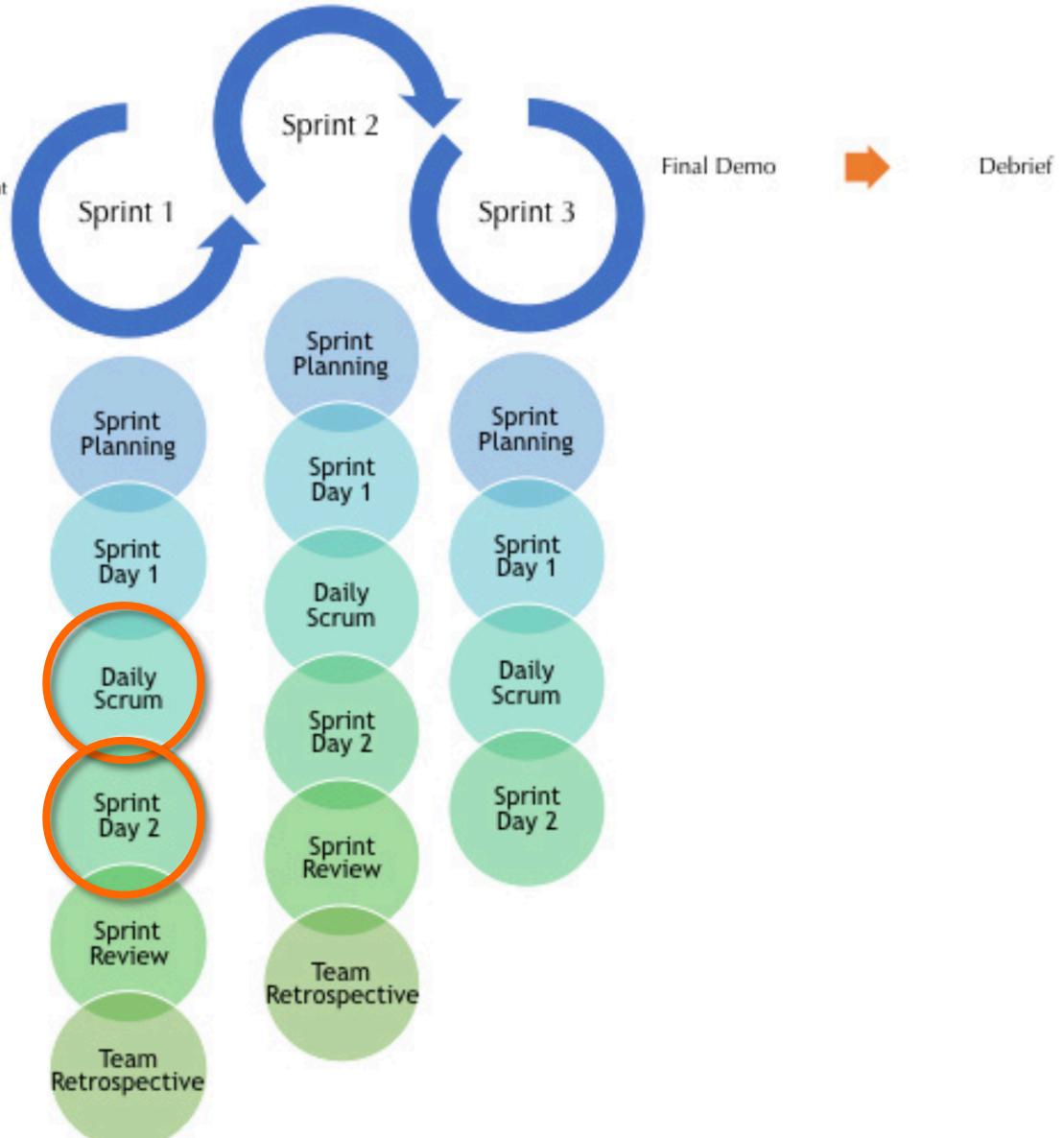
Am ersten Tag des Sprints gibt es kein Daily Scrum.

In der Praxis haben sich drei Fragen bewährt, die jeder Entwickler beantworten sollte:

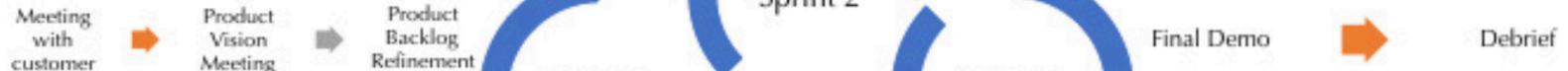
- Was habe ich seit gestern getan, um dem Scrum Team zu helfen, dass Sprint-Ziel zu erreichen?
- Was mache ich bis morgen, um dem Scrum Team beim Erreichen des Sprint-Ziels zu unterstützen?
- Was behindert mich oder das Scrum Team daran, das Sprint-Ziel zu erreichen?

### Sprint Day

Die Entwickler setzen User Story Cards (Tickets) in eine Geschichte im vorbereiteten Google Doc um.



## Events

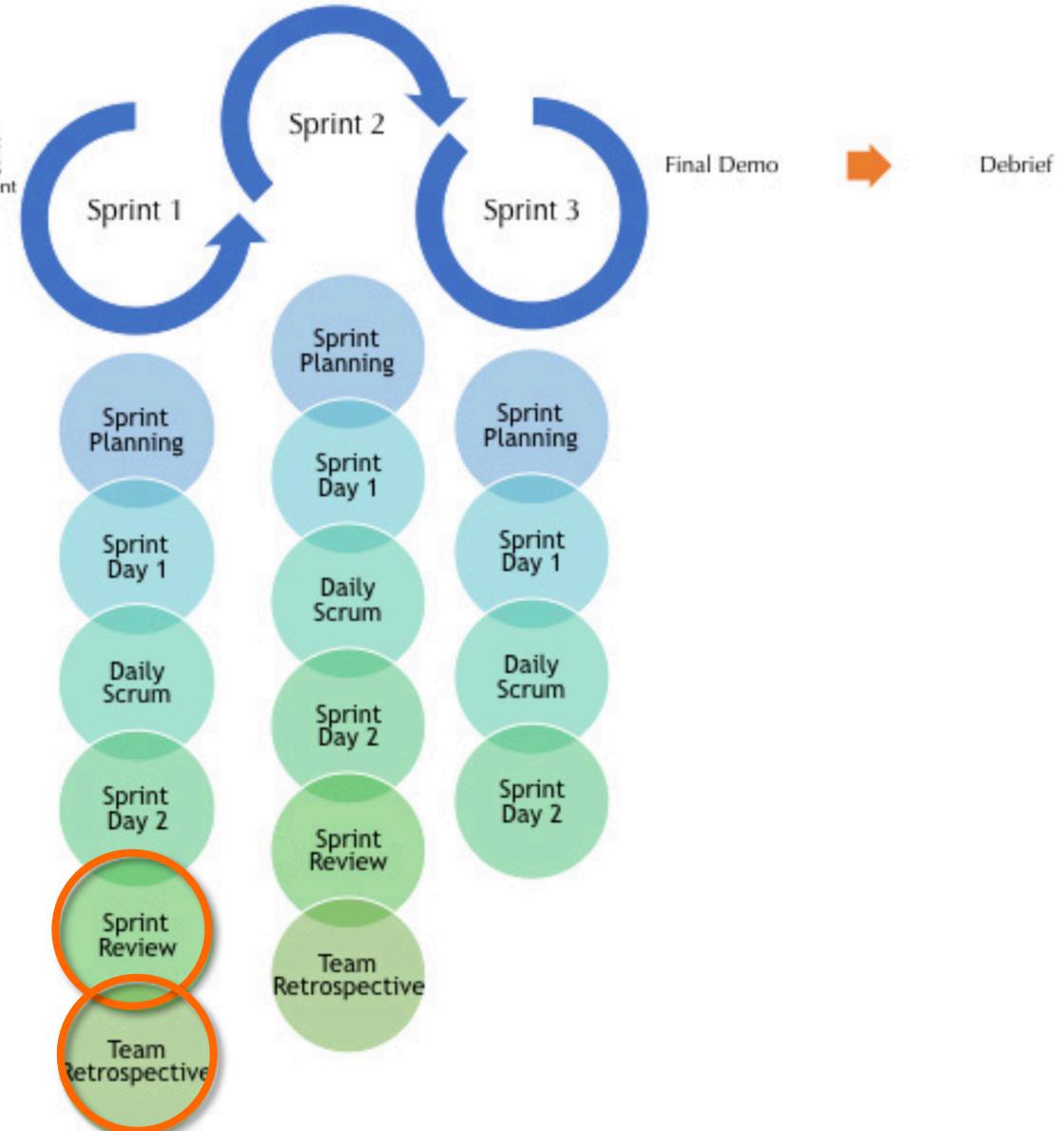


### ■ Sprint Review (mit PO)

Meeting organisiert vom Scrum Master, in dem alle Beteiligten gemeinsam das Inkrement prüfen. Ein Teammitglied liest den Text der Geschichte vor. Der Product Owner akzeptiert die abgeschlossenen User Stories (Tickets) und verschiebt die Story Cards in die Spalte „Done“ auf dem Miro Board. Alle Lücken und Mängel werden auf Haftnotizen erfasst, damit sie im nächsten Sprint Planning besprochen werden können.

### ■ Team Retrospective (mit PO)

Meeting organisiert vom Scrum Master, dessen Ziel darin besteht, zu sehen, wie das Team während des Sprints gearbeitet hat, und 1-3 konkrete Verbesserungen zu definieren, die im nächsten Sprint umgesetzt werden. Nach dem letzten Sprint gibt es keine Retrospektive.



## Events



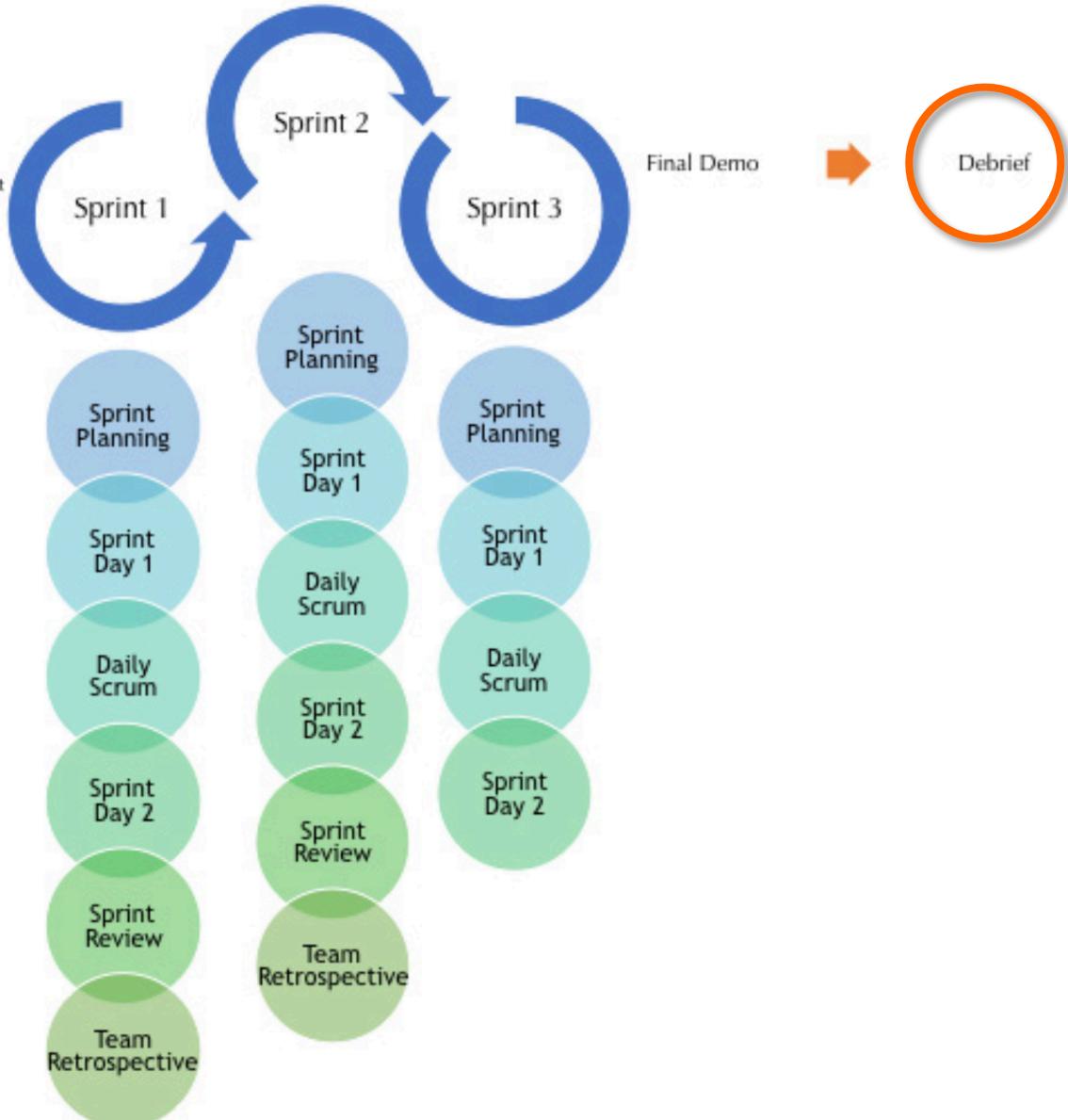
### Debrief

Dies ist der wichtigste Teil der Übung!!!

Die Studierenden reflektieren, was sie während des ScrumTale-Workshops gelernt haben.

Diese Erkenntnisse sind relevant für die Bonuspunkte!

Die zuvor vorgestellten Fragen können als Einstieg in die Diskussion genutzt werden.



## Praktikum



Viel Spaß und  
Erfolg!



