



Bildquelle: unsplash.com / Shahadat Rahman

Softwaretechnik-Labor

Prof. Dr. Christopher Kühn

Warum diese Veranstaltung teilweise Denglish ist ;-)



- **Sprachliche Ergänzung:** Für einige Begriffe, insbesondere solche, die dem Kontext digitaler Technologie entstammen, gibt es kein (annehmbares) deutsches Äquivalent



- **Quellennahes Arbeiten:** Einige Begriffe sind in der Aussage klarer, wenn man sie in der Original-Sprache (hier: Englisch) belässt



- **Zukünftiger Arbeitskontext:** In vielen Unternehmen mit internationaler Ausrichtung mit häufigem Sprachwechsel mischen sich die Begrifflichkeiten (das betrifft nicht nur Englisch). Sie können optional dagegen ankämpfen, oder sich anpassen

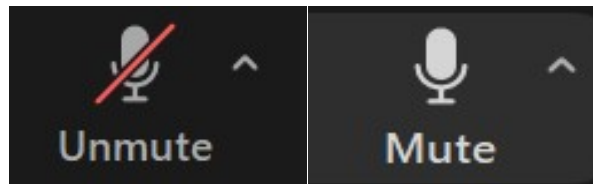


Sofern Sie einmal einen Begriff nicht verstehen, fragen Sie gern nach (das ist explizit so gedacht!), und/oder nutzen Sie Leo oder DeepL

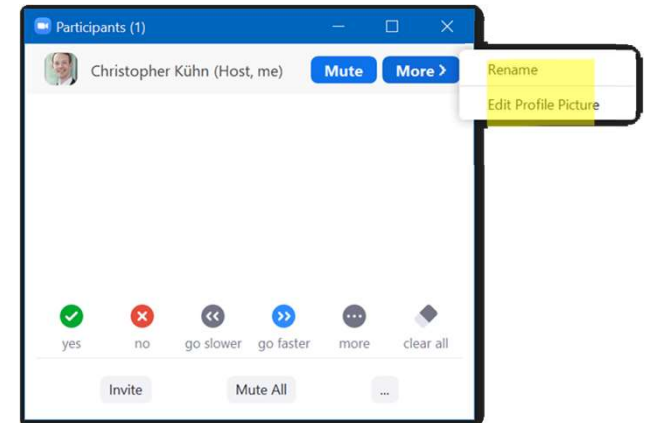
Wie wir gemeinsam eine gelungene virtuelle Veranstaltung schaffen – drei Grundregeln



Schalten Sie Ihre **Kamera** an. Optional: Lächeln oder verwirrt schauen



Schalten Sie grundsätzlich das **Mikro** aus, stellen Sie aber gern zwischendurch **Fragen**



Tragen Sie Ihren **Namen** in Zoom ein, um die Team-Diskussionen zu erleichtern

Quelle: Meme Bild <https://makeameme.org/meme/what-i-look-678fe6eb73>

Übergreifende Lernziele



Was Sie von dieser Veranstaltung erwarten können:

- Die eigenständige praktische Implementierung eines Softwareprojekts in Teamarbeit (Analyse, Planung, Durchführung, Präsentation)
- Das Verstehen des Kontexts der Softwareentwicklung und die Kommunikation mit (Senior) Management-Stakeholdern im Kontext der Softwareentwicklung



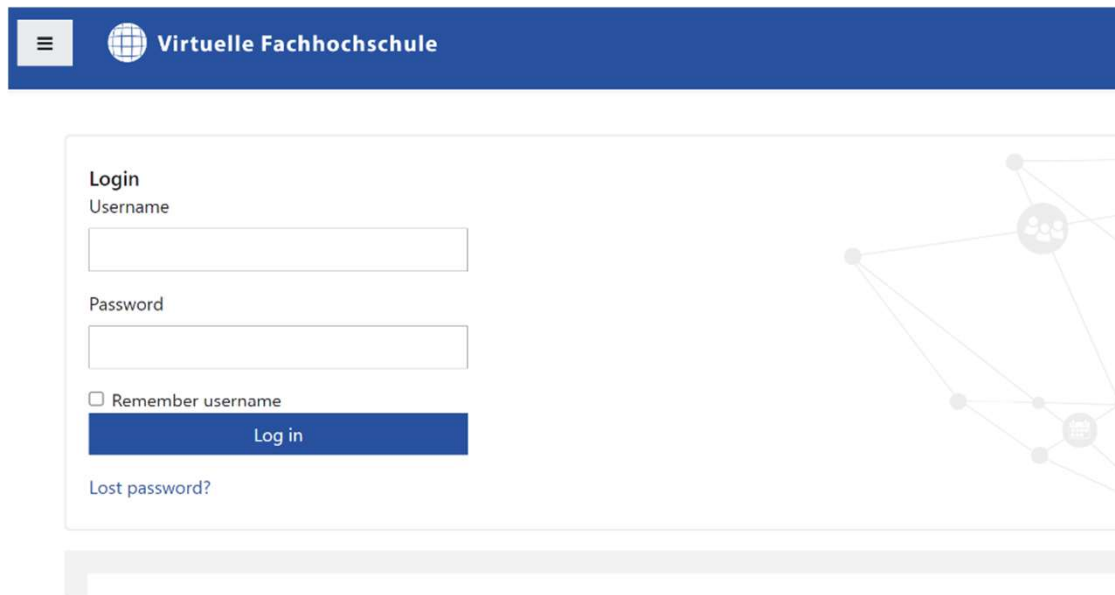
Was nicht im Fokus dieser Veranstaltung steht:

- Die Vermittlung von Grundlagen der Softwareentwicklung (diese sind in vorherigen Modulen abgedeckt worden)



Elearning – Oncampus Moodle

<https://moodle.oncampus.de/>



Menu icon

Virtuelle Fachhochschule

Login

Username

Password

☐ Remember username

Log in

Lost password?

Ankündigungen &
Informationen zu unserem
Kurs an der FH Kiel

Information seitens des Dekanats

Keine Weitergabe

Die Unterlagen, die Sie für Ihr Studium in dieser Lehrveranstaltung erhalten, sind ausschließlich für Ihren persönlichen Gebrauch. Sie dürfen die Materialien insbesondere nicht weitergeben oder anderen Personen zugänglich machen, egal in welcher Form (z.B. nicht über Dropbox, WhatsApp o.Ä.), denn sie sind urheberrechtlich geschützt.

Keine Aufzeichnung

Außerdem gilt: Sie dürfen Lehrveranstaltungen weder komplett noch teilweise aufzeichnen, weder als Ton- noch als Videoaufnahme. Dies gilt auch für Aufnahmen zum privaten Gebrauch.

Ein Verstoß hiergegen kann von den Lehrenden und der Fachhochschule Kiel geahndet werden.

Sie können in diesem Kurs Miro nutzen (es besteht jedoch kein Zwang)



- Wir werden in der Veranstaltung ein offen zugängliches **Miro-Board** verwenden
- Sie können sich einen eigenen Account anlegen oder anonym teilnehmen (klicken Sie auf den Link)
- Stellen Sie keine Informationen ins Miro, von denen Sie nicht möchten, dass sie JEDE PERSON MIT INTERNETZUGANG sieht (vermeiden Sie persönliche und/oder anderweitig rechtlich geschützte Informationen)
- Wenn Sie sich unwohl fühlen, z.B. wegen mangelndem Datenschutz, können Sie stattdessen auch PowerPoint oder andere (Offline-)Präsentationstools verwenden
- https://miro.com/app/board/uXjVM09BDqw=
- Passwort: softwareonline2024

Vorstellungsrunde

Miro board

https://miro.com/app/board/uXjVM09BDqw=
Passwort: softwareonline2024



- Bitte beantworten Sie die folgenden Fragen im **Miro Board**:
 1. Wo wohnen Sie aktuell? (Pin auf Karte setzen)
 2. Welche Programmiersprachen beherrschen Sie, und auf welchem Niveau (auf einer Skala von 1 bis 10, mit "1 - kaum" bis "10 - ausgezeichnet")?
 3. In welchen Softwareentwicklungsprojekten waren Sie bislang unterwegs?
- Anschließend stellen Sie sich kurz vor (in Zoom) :-)

Interdisziplinäre Wochen

<https://ida.fh-kiel.de/>

IDA



Interdisziplinäres
Arbeiten an der
Fachhochschule Kiel

Achtung: In den IDW finden keine regulären Lehrveranstaltungen statt



Zielsetzung



- Im Rahmen des Projekts sollen **kleine Projekte** von einem Team aus ca. drei Studierenden (+/-) durchgeführt werden. Sie können Ihr Team grundsätzlich selbst über Moodle wählen, wobei auf eine ungefähre Gleichverteilung der Anzahl der Teilnehmer geachtet wird. Später hinzukommende Studierende werden den Teams zugeordnet. Einmal bestehende Teams bleiben – auch bei reduzierter Teilnehmerzahl – bestehen. Bitte wählen Sie Ihr Team mit Bedacht und gehen Sie fair miteinander um.
- Inhaltlich geht es um die **Fähigkeit**, eine softwaretechnische Problemstellung in Teamarbeit zu analysieren, mit den erlernten Methoden eigenständig Lösungswege aufzuzeigen, sowie Teilaufgaben und deren jeweilige Schnittstellen zu definieren und anschließend zu implementieren.
- **Zu liefern** sind (soweit nicht mit dem Kursbetreuer anders vereinbart): eine auf einem Windows- oder Linux-System lauffähige und installierbare Software, die den Anforderungen entspricht, zwei-wöchentliche Tätigkeitsberichte (siehe Vorlage), die Dokumentation für Entwickler und Anwender, sowie Präsentationen (Zwischen-/Endpräsentation mit Demo).
- Auch nimmt der Kursleiter als „Sponsor“ selektiv an **Teammeetings** teil (z.B. Sprint Review, ggf. auch mal eine Sprint Retrospective). Das Zeitfenster für alle Sitzungen ist ausnahmslos die offizielle Zeit der Lehrveranstaltung.

Arbeitsmodus



- Für die Entwicklung können Sie *grundsätzlich* jede **Programmiersprache** verwenden, die die Anforderungen des Projekts erfüllen kann, soweit nicht anders angegeben. Einigen Sie sich in Ihrem Team am besten auf eine Sprache, die Sie als Team am besten beherrschen, und die das Problem am besten löst. Es wird erwartet, dass Sie Ihre technischen Kenntnisse *selbstständig* weiter ausbauen (z.B. über Funktionsreferenzen, Tutorials).
- Wenden Sie bei der Entwicklung die **Scrum-Methodik** an (Empfehlung: 1- oder 2-Wochen-Sprints) und überlegen Sie sich, welche Inhalte Sie in welcher Weise im jeweiligen Sprint Review präsentieren wollen. Schätzen und dokumentieren Sie den zeitlichen Aufwand Ihrer Sprints. Der/die Kursleiter sind in dieser Logik "Sponsoren", die am Ergebnis, d.h. dem Produkt interessiert sind.
- Verwenden Sie zur Erleichterung der (virtuellen) Zusammenarbeit **Werkzeuge**, die dem Stand der Technik entsprechen (z.B. IDEs wie MS Visual Studio [Code] oder Eclipse; Plattformen/Tools wie z.B. Atlassian Jira, Trello, Github, Miro etc.). **ChatGPT ist explizit erlaubt – Sie sind jedoch verpflichtet, Ihre Prompts und deren Ergebnis in einem ChatGPT-Register zu dokumentieren und später abzugeben.**

Vermeidung von Free Riding („Trittbrettfahren“)



- Erfahrungsgemäß funktionieren die meisten Teams nach einer Weile recht harmonisch. Allerdings kommt es in einzelnen Teams immer wieder vor, dass sich einige Teammitglieder (bewusst oder unbewusst) „mitziehen“ lassen, ohne selbst viel zu tun.
- Diese „**Free Rider**“ könnten daher theoretisch eine passable Note erreichen, ohne selbst etwas zum Projekt beigetragen zu haben
- Um das „Free Riding“ weitgehend zu verhindern und Transparenz zu schaffen, sind dem Kursleiter **zwei-wöchentliche Tätigkeitsberichte** mit dem dazugehörigen Workload zuzusenden. Typischerweise sind das die Themen, an denen während des Sprints gearbeitet wurde mit dem geschätzten und dem tatsächlichen Zeitaufwand (kurze Stichworte reichen). Ergänzend nimmt der Kursleiter an den Sitzungen selektiv ohne Vorankündigung teil
- Darüber hinaus ist das **selbstständige Lösen von Konflikten** (etwa inhaltlicher, zeitlicher und persönlicher Natur) eine Kernkompetenz, die in diesem Kurs explizit bewertet wird

Zeitplan



- 15.09.2025: Kurseinführung
- **19.09.2025 (Deadline): Team-Selbstzuordnung** – Email mit Teammitglieder an Dozenten (potentiell später hinzukommende Studierende werden zugeordnet)
- **19.09.2025 10:00 Uhr (Deadline):** Rückmeldung der Teams an die Kursleitung (per Email) bezüglich der **Themenwahl** (entweder aufgeführte Themen, oder zu genehmigende eigene Themen).
- 22.09.2025: Rückmeldung Kursleitung zur Themenzuordnung bzw. zur Genehmigung/Ablehnung der eigener Themen
- von 22.09. bis 08.12.2025: Teilnahme des Kursleiters an den (Scrum-)Meetings/Diskussionen der Teams (1h im Rahmen der Kurszeit; bitte vor dem ersten Team **dauerhafte Zoom-Terminserie** im Kurszeitfenster per Email an den Kursleiter senden).
- **06.12.2025: Präsenztermin:** Präsentation aktuelle Demo und Key Learnings aus dem Projekt, Austausch mit anderen Teams
- 15.12.2025: Endpräsentation (ca. 15min pro Team; per Zoom) zum finalen Stand der Software
- 02.01.2026: Abgabe Dokumentation

Aufgabenstellung



- **Mögliche Aufgaben dieses Semester:**

1. In Kooperation mit **Ernst Krebs GmbH & Co. KG: Logistikoptimierung & Kundenangebotskalkulation**. Erstellung eines Programms als Vorlage von optimierten Preisen für Fracht und Rohstoffe und/oder Verwertungs-/Entsorgungen für Kundenanfragen. Erstellung eines Angebotes für Kunden sowie von Tagesplanung für LKW und Kieswerksplanung
2. Entwicklung einer KI-basierten Anwendung zur **automatischen Erfassung** der Inhalte von verschiedenen **PDF-Eingangs- und Ausgangsrechnungen**. Die Inhalte der Rechnungen sollen unabhängig vom Rechnungsformat automatisch erfasst und klassifiziert werden (über OCR und KI), und anschließend strukturiert in eine Drittanwendung (z.B. Excel) übertragen werden. Es soll Webinterface zum (überwachten) Lernen bei der inhaltlichen Erfassung erstellt werden. Es soll zudem klassifiziert werden, ob es sich um eine Eingangsrechnung oder eine Ausgangsrechnung handelt
3. Experimentell: **Fleetmanagement für LoRa-IoT-Geräte** (Fokus: MeshCore)

- **Alternativaufgabe (nach Absprache bis 19.09.2025):**

Suchen Sie sich aus Ihrem professionellen Umfeld eine praktische Herausforderung, für die Sie eine vergleichbar komplexe Software vollständig neu erstellen (z.B. eine KI-Lösung), und die natürlich noch nicht teilweise, vollständig oder in ähnlicher Form gelöst wurde. Beschreiben Sie Ihr Projekt auf einer Seite entlang folgender Abschnitte: Top-down Kurzbeschreibung des Vorhabens (1-2 Sätze), Ausgangssituation mit Herausforderung und Ziel, geplantes Vorgehen / Workstreams, geplante Funktionalitäten der Software, Sprache/Entwicklungsframework

Deep Dive zu den Wahlthemen (1/3)



1. Logistikoptimierung & Kundenangebotskalkulation

Frameworks/Libraries: frei

Sprache: tbd. Webinterface typischerweise PHP/JavaScript, Datenbank SQL

a) Erstellung eines Programms, als Vorlage von optimierten Preisen für Fracht und Rohstoffe und/oder Verwertungs-/Entsorgungen für Kundenanfragen. Erstellung eines Angebotes für Kunden sowie von Tagesplanung für LKW und Kieswerksplanung.

b) Vorgehensweise:

- Erfassen aller Sand -und Kiesgruben / Lagerplätze / Hafen- /Bahnumschlagplätze / sowie aller Verwertungsmöglichkeiten und Deponien (insbesondere aller Werke, Zu- und Abgänge mindestens 1x jährlich)
- Erfassen räumliche Lagen, Preise und Materialien und zeitlich Verfügbarkeit (Prüfung neuer Preise 2x jährlich oder auf Grund besonderer Anlässe)
- Verknüpfen von Mengen und Transportmöglichkeiten – Das heißt im Verbindung mit Ressourcen, Mengen Anzahl der benötigten Fahrzeuge, Typen der Fahrzeuge: Sattel, 4-Achser, 3-Achser, Schwerlasttransporte ggf. Container- Mulden Fahrzeuge.

- Erstellung einer Vorlage/ Maske mit Einbindung von Genehmigungsverfahren z. B. Schwerlastverkehr, Straßensperrungen (angekündigt/wirklich bestehend)
- Möglichkeit auch durch Vorgabe von unternehmerischen Entscheidungen (z. B. wir setzen eigene Ressourcen ein)
- Ausgangsbasis:
 - Conwin (CRM-System)
 - Anlagenstandorte (fremd und eigen)
 - Preise
 - Artikel/Sorten inkl. Qualitäten
 - Kunden
- Anfragen:
 - Leistungsverzeichnisse aus Ausschreibungen (große Mengen und komplex)
 - Emailanfragen (klein bis mittlere Mengen)
 - Telefonische Anfragen (meist Privatkunden)
- ... (siehe Folgeseite)

Deep Dive zu den Wahlthemen (2/3)



1. Logistikoptimierung & Kundenangebotskalkulation

- Ergebnisse:
 - Zusammenfassung der Mengen je Artikel/Sorte inkl. Qualität
 - Lieferzeitpunkt und Laufzeit -> Tagesmengen
 - Auswahl geeigneter Lieferwerke aufgrund der geforderten Qualität
 - Rudimentäre Kalkulation Material und Fracht
 - Vorschläge für Lieferwerke (nach Entfernung oder Preisvorteil)
 - Angebot für den Kunden

Deep Dive zu den Wahlthemen (3/3)



2. ER/AR-Klassifizierung

Frameworks/Libraries: frei (z.B. [Easy]OCR, YOLOv8 oder TensorFlow/Keras, NLP libraries)
Sprache: KI typischerweise Python, Webinterface typischerweise PHP/JavaScript

Ähnliche Projektbeispiele:

<https://medium.com/analytics-vidhya/invoice-information-extraction-using-ocr-and-deep-learning-b79464f54d69>

https://www.researchgate.net/publication/331822005_Invoice_Classification_Using_Deep_Features_and_Machine_Learning_Techniques

<https://www.google.com/search?client=firefox-b-d&q=ocr+classification+invoice#ip=1>

<https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp/>

Leistungsbewertung



- **Beurteilung aus Anwendersicht, z.B.:**
 - Erfüllung der funktionalen Anforderung
 - Verwendbarkeit der entwickelten Software (z.B. Eingabehilfen, Benutzerdialoge, Menüführung)
 - Funktionsfähigkeit/ Korrektheit der Ergebnisse
 - Stabilität der entwickelten Programme (z.B. Verhalten bei Fehleingaben)
 - UX/UI Design
- **Beurteilung aus Entwicklersicht, z.B.:**
 - Korrekte Anwendung und eigenständige Weiterentwicklung der in den Vorlesungen vermittelten Inhalte (wichtigstes Einzelkriterium!)
 - Strukturierung des Projektes
 - Datenmodellierung, Sinnvolle Normalisierung der Datentabelle
 - Programmstil (Namenskonventionen, Vermeidung von Redundanzen etc.)
 - Verständlichkeit / Nachvollziehbarkeit des Codes (insbesondere Kommentierung, Bezeichnung der Variablen)
- **Teamarbeit, z.B.:**
 - Konfliktmanagement (etwa inhaltlicher, zeitlicher oder persönlicher Konflikte)
 - Transparenz bei der verteilten Entwicklung (via Jira/Trello oä)
- **Dokumentation, z.B.:**
 - Inhaltliche Qualität
 - Struktur / Nachvollziehbarkeit
 - Formalien (korrekte und vollständige Quellenangabe; Rechtschreibung)
 - Vollständigkeit (Modelle, Testdokumentation etc.)
- **Präsentation(en), z.B.:**
 - Sprint-Sitzungen
 - Präsentation

Übersicht

APPENDIX

Kommunikation und Stakeholder
Vorgehensmodell
(Entwicklungsansätze und
Projektsteuerung, inkl. SCRUM)

Übersicht

APPENDIX

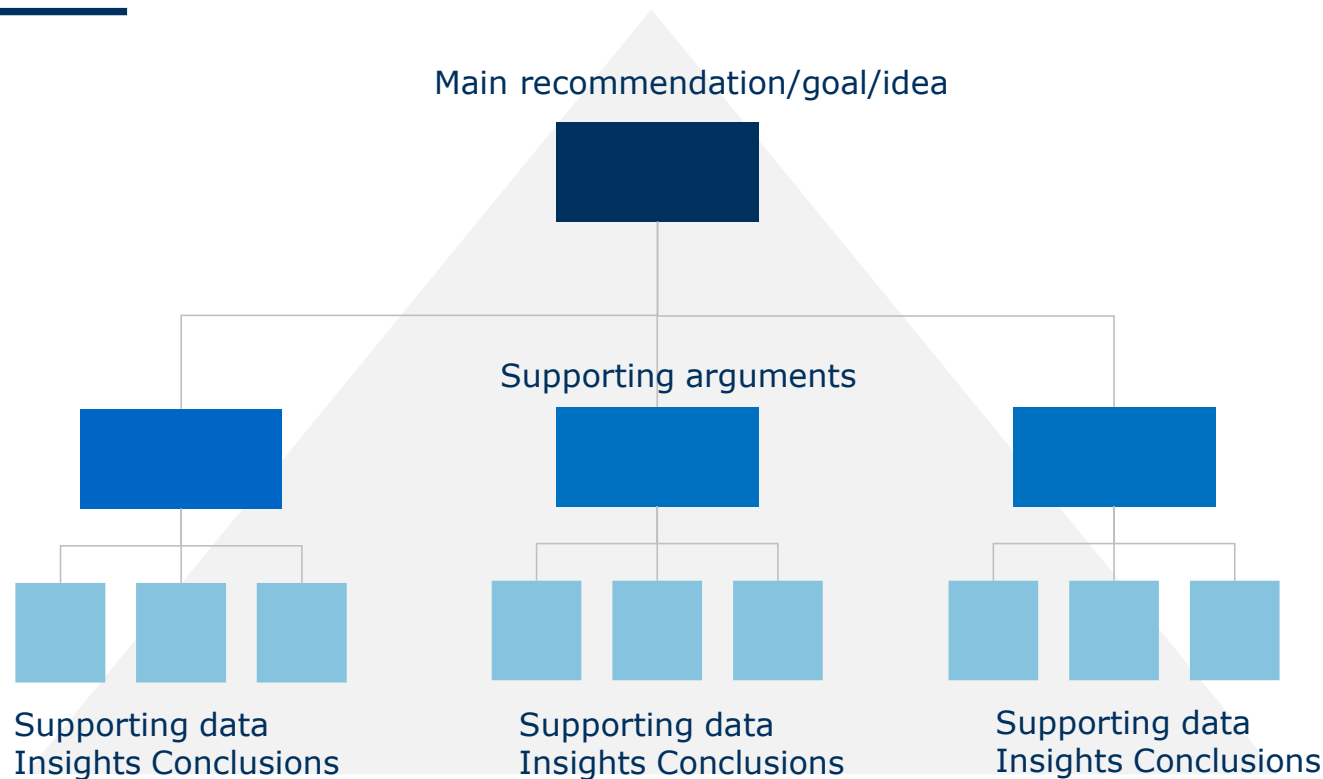
Kommunikation und Stakeholder

Vorgehensmodell
(Entwicklungsansätze und
Projektsteuerung, inkl. SCRUM)

Exkurs: Eine einfache Möglichkeit, Verantwortlichkeiten in SteerCos festzuhalten, ist eine "Next steps"-Seite

| Action | Responsible | Deadline |
|--------------------------------------|------------------|-----------------------|
| Create first draft of digital vision | Head of Strategy | SteerCo on 01.10.20xx |
| ... | ... | ... |
| ... | ... | ... |
| ... | ... | ... |
| ... | ... | ... |

Exkurs: Top down Kommunikation hilft dem Management, die Ergebnisse schneller zu verstehen



Top down



Bottom up

Quelle: Minto

Sie sollten bei der Präsentation einige praktische Tipps für Inhalt, Form und Vortrag berücksichtigen (1/3)

Inhalt Präsentation



Adressaten-gerechte Gestaltung

Welchen **Hintergrund** und **Informationsstand** haben die Adressaten? Wo muss man sie (auch bei zeitlichen Abständen) abholen? Manchmal ist das „KLV-Prinzip“ hilfreich (verständlich für Kinder, Laien und Vorstände – auch inhaltlich und sprachlich). (Ggf. Pre-read versenden)

Story

Welche **Geschichte** möchten wir insgesamt erzählen? Welche Seiten und Inhalte sind dafür notwendig? (Dummy Master nur mit Titeln anlegen.) Wie wird der Übergang von einer zur nächsten Seite erzählt? (Praxis-Tipp: Top-down erzählen – Ergebnis zuerst, Argumente und Details danach)

Handlungs-aufforderung

Warum steht der Inhalt auf der Seite? **Was** sollen die Adressaten mit der Information **machen**? (z.B. Entscheidung treffen) Was nimmt man aus der Seite mit, ohne sie im Detail lesen zu müssen? (z.B. „action titles“ / „so-what“ Boxen [blau], call-outs [gelb])

Kennzeichnung Qualität

Inhaltliche Qualität und etwaige „Baustellen“ als solche **kennzeichnen** („If in trouble, put a bubble“), z.B. über einen „draft“, „work in progress“, „to be validated“/„to be aligned“ **Sticker**

Stichhaltigkeit

Welche Fragen könnten gestellt werden? Seien Sie auf vereinzelte **Detailfragen** vorbereitet. Annahmen (z.B. für ihre Berechnungen) sollten im Appendix klar dokumentiert sein

! Wir empfehlen <x> Stunden zu investieren, um eine in Inhalt und Form ansprechende Präsentation zu erstellen

1 Sie sollten bei der Präsentation einige praktische Tipps für Inhalt, Form und Vortrag berücksichtigen (1/3)

Legende
(z.B.
Erklärung
Shading)

■ Details im
Appendix

Inhalt Präsentation



Adressaten-
gerechte
Gestaltung

Welchen **Hintergrund** und **Informationsstand** haben die Adressaten? Wo muss man sie (auch bei zeitlichen Abständen) abholen? Manchmal ist das „KLV-Prinzip“ hilfreich (verständlich für Kinder, Laien und Vorstände – auch inhaltlich und sprachlich). (Ggf. Pre-read versenden)

Story

Welche **Geschichte** möchten wir insgesamt erzählen? Welche Seiten und Inhalte sind dafür notwendig? (Dummy Master nur mit Titeln anlegen.) Wie wird der Übergang von einer zur nächsten Seite erzählt? (Praxis-Tipp: Top-down erzählen – Ergebnis zuerst, Argumente und Details danach)

Handlungs-
aufforderung

Warum steht der Inhalt auf der Seite? **Was** sollen die Adressaten mit der Information **machen**? (z.B. Entscheidung treffen) Was nimmt man aus der Seite mit, ohne sie im Detail lesen zu müssen? (z.B. „action titles“ / „so-what“ Boxen [blau], call-outs [gelb])

Kennzeichnung
Qualität

Inhaltliche Qualität und etwaige „Baustellen“ als solche **kennzeichnen** („If in trouble, put a bubble“), z.B. über einen „draft“, „work in progress“, „to be validated“/„to be aligned“ **Sticker**

Stichhaltigkeit

Welche Fragen könnten gestellt werden? Seien Sie auf vereinzelte **Detailfragen** vorbereitet. Annahmen (z.B. für ihre Berechnungen) sollten im Appendix klar dokumentiert sein

Shading

! Wir empfehlen <x> Stunden zu investieren, um eine in Inhalt und Form ansprechende Präsentation zu erstellen

„So-what“ Box

Sie sollten bei der Präsentation einige praktische Tipps für Inhalt, Form und Vortrag berücksichtigen (2/3)

Form Präsentation



Lesbarkeit / Aufmerksamkeit

Qualitätssicherung

Kontext

- Inhalte **strukturieren** und dabei Lesefluss berücksichtigen (idealerweise: von links nach rechts, von oben nach unten)
 - **Schrift** lesbar - idealerweise nicht kleiner als 12pt, und einheitliche Schriftgröße innerhalb der Seite (Ausnahmen: Titel, Fußnote, Legende)
 - Bei textlastigen Seiten ggf. vereinzelt **Grafiken** ergänzen
 - **Farben** begrenzen auf 2-3. Kontrastfarben zur Hervorhebung verwenden
 - Verwendete **Symbolik** wie Farbcodes erläutern (Legende oben auf der Seite)
 - Bei **Hintergründen** farblichen Kontrast zum Text sicherstellen
 - Selektives **Bolding** (für Überschriften / wichtigste Elemente)
 - „**Flippen**“ vermeiden (durch Verschiebung von Objekten von Seite zu Seite)
 - Inhalt je Seite begrenzen, ggf. **Seiten aufteilen**. Bei vollen Seiten (wie dieser): notfalls Trennlinien oä ergänzen
-
- **Typos** korrigieren (auch die PowerPoint Rechtschreibprüfung nutzen)
 - **Berechnungen** nochmals überprüfen auf Sinnhaftigkeit, Klarheit (Rechenweg, Einheiten etc.) und kalkulatorische Richtigkeit! (am besten nicht durch den Ersteller)
 - **Quellen** und **Fußnoten** prüfen (falls vorhanden)
 - **Seitenzahlen** einfügen bzw. Überdeckung vermeiden
-
- Bei langen Präsentationen **Kapitel-Tracker** einfügen
 - Strukturgrafiken sowie **Shadings** verwenden, um von einer Seite zur nächsten zu kommen

Diese Seite ist zu voll – eine Aufteilung auf mehrere Seiten wäre sinnvoll

Sie sollten bei der Präsentation einige praktische Tipps für Inhalt, Form und Vortrag berücksichtigen (3/3)

Vortrag



Vor dem Vortrag

- **„Dry-run“** (Testdurchlauf) durchführen und von befreundeten Teams Rückfragen stellen lassen
- **Rollen-/Rede-Aufteilung** für den Vortrag und ggf. auch für bestimmte Rückfragen vornehmen
- (Bei **Nervosität**: einfach mal tief durchatmen. Sie sind gut vorbereitet. Und es ist nur ein Vortrag. Und zudem einer von Vielen – für Sie und den Adressaten)
- Im **Remote-Betrieb**: Video-/Audio-Einstellungen und Einwahldaten vorher checken, eine ruhige Umgebung aufsuchen

Während des Vortrags

- **Kontext** herstellen und **Meeting-Ziele** vorstellen (ggf. Team vorstellen, falls noch kein Kontakt bestand)
- **Langsam** und **deutlich** sprechen, vereinzelt Pausen machen

Nach dem Vortrag

- Follow-ups zu den gemeinsam definierten **nächsten Schritten** (wer macht was mit wem bis wann?)

Diese Seite ist deutlich leichter verdaulich – durch Struktur, Schriftgröße, Zeilenabstände, Bolding

Literatur

- Stark (2020): Digital Transformation of Industry. Springer Nature Switzerland: Cham.
- Minto, B.: Barbara Minto: "MECE: I invented it, so I get to say how to pronounce it". McKinsey. Online verfügbar unter <https://www.mckinsey.com/alumni/news-and-insights/global-news/alumni-news/barbara-minto-mece-i-invented-it-so-i-get-to-say-how-to-pronounce-it>, zuletzt geprüft am 02.02.2021.
- Zwikael, O.; Smyrk, J. R. (2019): Project Management. Springer International Publishing: Cham.

Übersicht

APPENDIX

Kommunikation und Stakeholder

Vorgehensmodell (Entwicklungsansätze und Projektsteuerung, inkl. SCRUM)

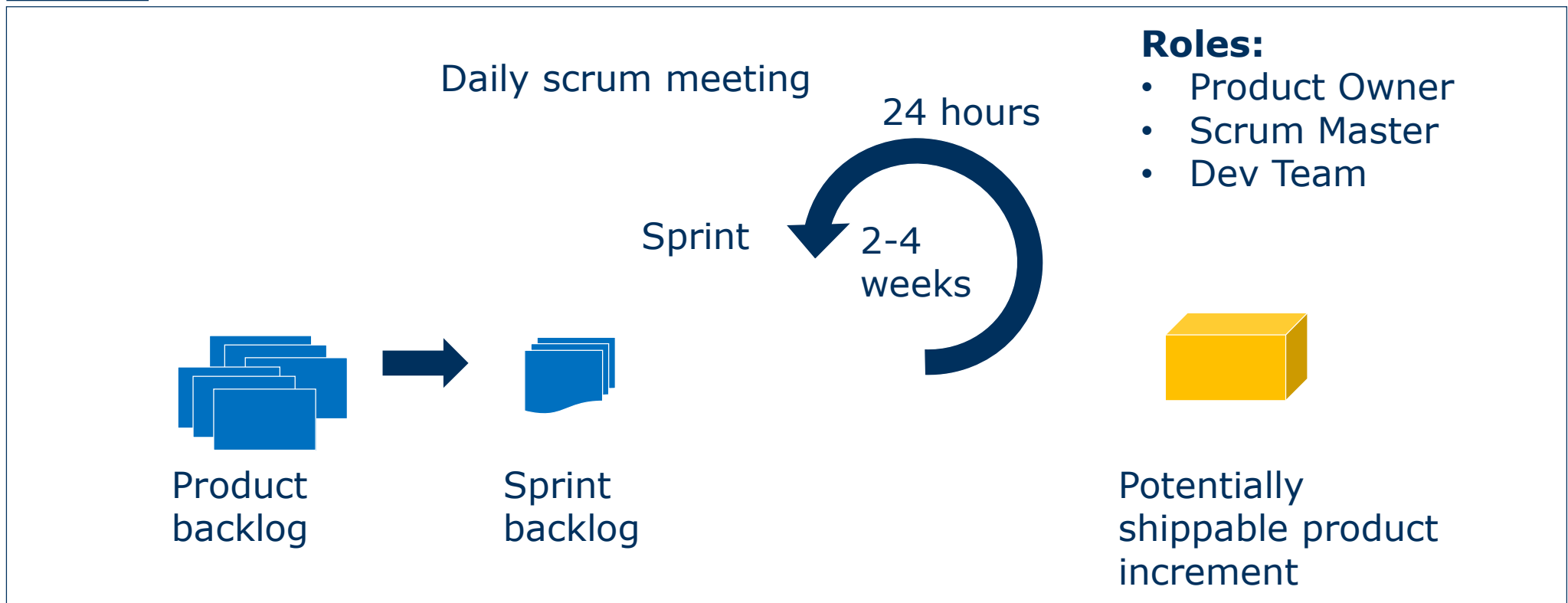
Wer mehr zum Thema erfahren möchte, der lese den originalen Scrum Guide von Schwaber & Sutherland (auf Deutsch oder Englisch)



<https://www.scrum.org/resources/scrum-guide>

Quelle: Schwaber/Sutherland (2020) - <https://www.scrum.org/resources/scrum-guide>

Scrum im Überblick



Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>

Scrum Framework Bestandteile

■ Nachfolgend detailliert

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Commitments

- Product Goal
- Sprint Goal
- Definition of Done

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>

Deep Dive Roles: Scrum Team



Scrum Team

- 10 or less people (typically 5-9 people), no sub-teams / hierarchies
 - Official guide names "Scrum Master", "Product Owner" and "Developers"
 - Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint (e.g., Devs, UX/UI Designer, SMEs)
- Members should be full-time with exceptions (e.g., SMEs)
 - Teams are self-organizing
 - Responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Deep Dive Roles: Product Owner



Product Owner

- Accountable for **maximizing the value** of the product resulting from the work of the Scrum Team
- Define the **features** of the product
- Perform **product backlog management**:
 - Developing and explicitly communicating the Product Goal;
 - Creating and clearly communicating Product Backlog items;
 - Ordering Product Backlog items; and,
 - Ensuring that the Product Backlog is transparent, visible and understood.
- Accept or reject **work results**

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Deep Dive Roles: Scrum Master



Scrum Master

- Accountable for establishing Scrum and enacting **Scrum values** and **practices**
- Ensure that the team is fully functional and productive
- Serves the **team** in several ways, e.g.:
 - Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox
 - Causing the removal of impediments to the Scrum Team's progress
- Also serves the **Product Owner** (e.g., Facilitating stakeholder collaboration; techniques for effective Product Goal definition and Product Backlog management) and the **organization** (e.g., Leading, training, and coaching the organization in its Scrum adoption)

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Scrum Framework Bestandteile

■ Nachfolgend detailliert

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

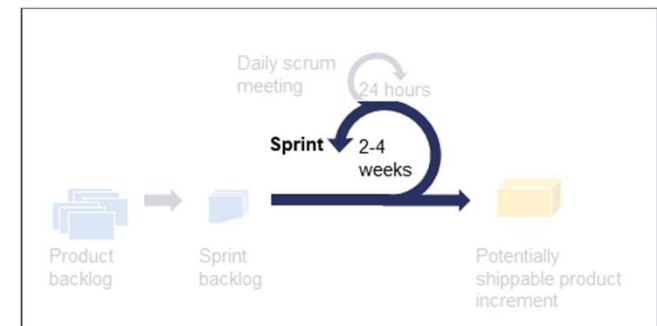
Commitments

- Product Goal
- Sprint Goal
- Definition of Done

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>

Sprints

- Scrum projects make progress in a **series of “sprints”**
- Typical **duration** is **2–4 weeks** or a calendar month at most
- New Sprint starts immediately after the conclusion of the previous Sprint
- Product is **designed, coded, and tested** during the sprint



- **During the Sprint:**
 - No changes are made that would endanger the Sprint Goal;
 - Quality does not decrease;
 - The Product Backlog is refined as needed; and
 - Scope may be clarified and renegotiated with the Product Owner as more is learned

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

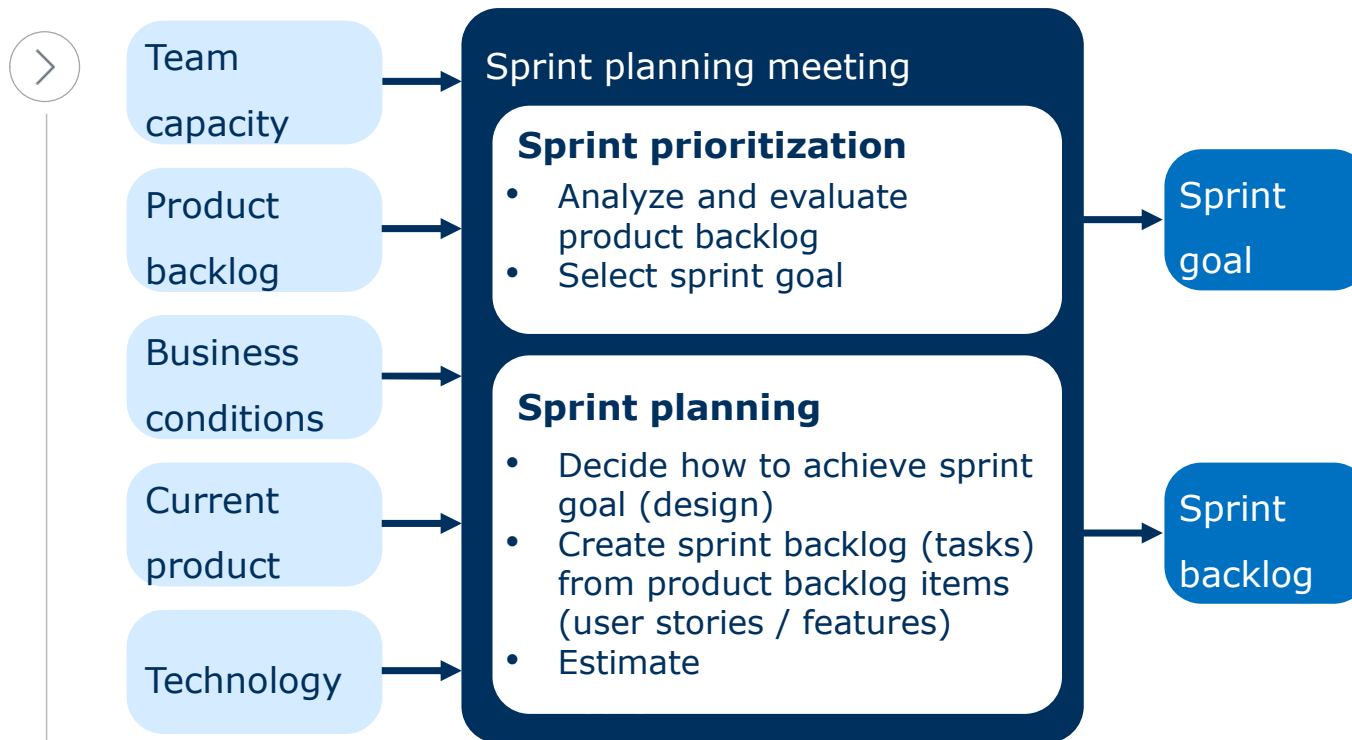
Sprint Planning

Sprint planning addresses three topics:

Topic One: Why is this Sprint valuable?

Topic Two: What can be Done this Sprint?

Topic Three: How will the chosen work get done?



Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Zusammenspiel Product Backlog / Sprint Backlog

Product backlog:

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

...

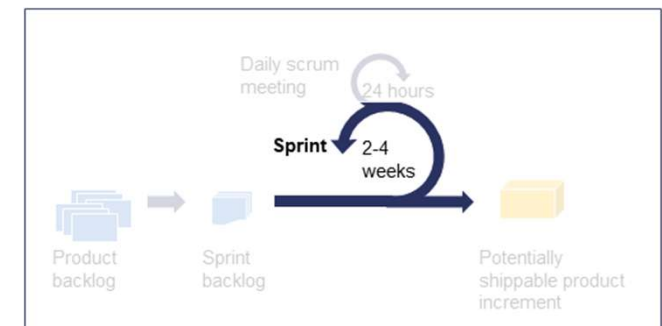


Sprint backlog:

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

Daily Scrum meeting

- **Inspect progress** toward the Sprint Goal and adapt the Sprint Backlog as necessary
- Meeting parameters:
 - **15-minutes**
 - same time and place every working day of the Sprint
- Primarily **for** team members, Scrum Master, product owner (guest can be invited)
- Helps to
 - Improve **communications**
 - identify **impediments**
 - promote **quick decision-making**, and
 - eliminate the need for other meetings
- BUT: is **not** a **longer problem solving session!**



Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Sprint Review

- Team presents what it **accomplished** during the sprint
- Meeting parameters:
 - **Maximum 4h** for 4 weeks sprint (**usually shorter**)
 - 2-hour prep time rule
 - Informal, no slides
 - *In a virtual setup, one person can share the screen for the demo presentation*
- Typically takes the form of a **demo** of new features or underlying architecture
- However: It's a **working session** – avoid limiting it to a presentation
- Whole team participates, guests are welcome (typically key stakeholders that can see the progress towards the Product Goal)

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Sprint Retrospective

Purpose: plan ways to increase quality and effectiveness

Meeting parameters:

- Done after every sprint
- **Maximum of 3h** for 4 weeks sprints (usually shorter)
- *In a virtual setup, you can use a Miro board with stickies for the "what went well" and "what do we want to improve"*

Whole team participates

- ScrumMaster
- Product owner
- Team
- Possibly customers and others

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Schwaber/Sutherland (2020)

Scrum Framework Bestandteile

■ Nachfolgend detailliert

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Commitments

- Product Goal
- Sprint Goal
- Definition of Done

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>

Scrum Artifacts & Commitments

Artefacts:



- Emergent, ordered list of what is needed to improve the product
- Single source of work undertaken by the Scrum Team



- is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how)
- is a plan by and for the Developers

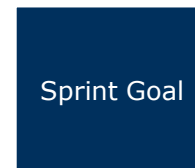


- concrete stepping stone toward the Product Goal
- In order to provide value, the Increment must be usable.

Commitments:



- describes a future state of the product which can serve as a target for the Scrum Team to plan against
- is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal



- is the single objective for the Sprint
- committed by developers, but it provides flexibility in terms of the exact work needed to achieve it



- is a formal description of the state of the Increment when it meets the quality measures required for the product.
- The moment a Product Backlog item meets the Definition of Done, an Increment is born.

Quelle: Schwaber/Sutherland (2020)

Example artefacts (1/2)

Sample product backlog:

| Backlog item | Estimate |
|--|----------|
| Allow a guest to make a reservation | 3 |
| As a guest, I want to cancel a reservation. | 5 |
| As a guest, I want to change the dates of a reservation. | 3 |
| As a hotel employee, I can run RevPAR reports (revenue-per-available-room) | 8 |
| Improve exception handling | 8 |
| ... | 30 |
| ... | 50 |

Sample sprint goal:

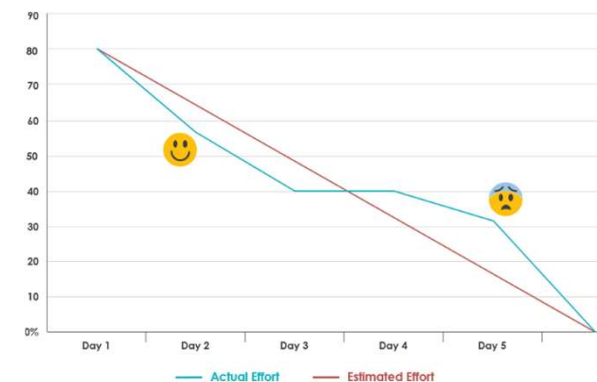
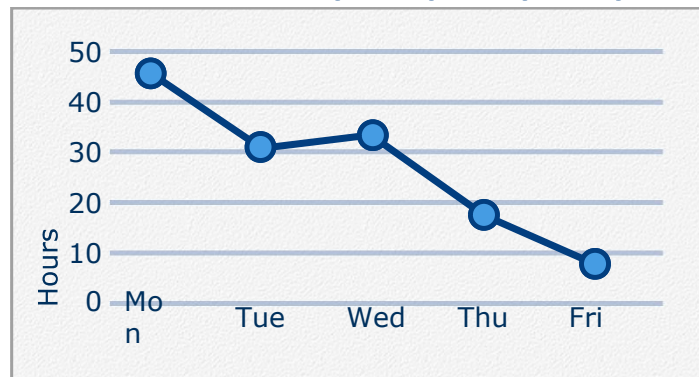
Make the application run on SQL Server in addition to Oracle.

Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>

Example artefacts (2/2)

Sample burndown chart:

| Tasks | Mon | Tues | Wed | Thur | Fri |
|-------------------------|-----|------|-----|------|-----|
| 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 | | | | |



Quelle: Mountain Goat Software and Mike Cohn, <https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation>; Image: <https://www.visual-paradigm.com/scrum/scrum-burndown-chart/>

Beispiel: Ein vereinfachtes PMO Reporting Template



Workstream

Responsible

Current activities/progress

Time planned/time spent

Status

Workstream A Person 1

- ...
- ...

... h / ... h



Workstream B Person 1

- ...
- ...

... h / ... h



Workstream C Person 2

- ...
- ...

... h / ... h



Workstream D Person 2

- ...
- ...

... h / ... h



Workstream E Person 3

- ...
- ...

... h / ... h



(In addition you could add a timeline / due dates, budget, benefits, risks, dependencies, decision needs, etc. ...)