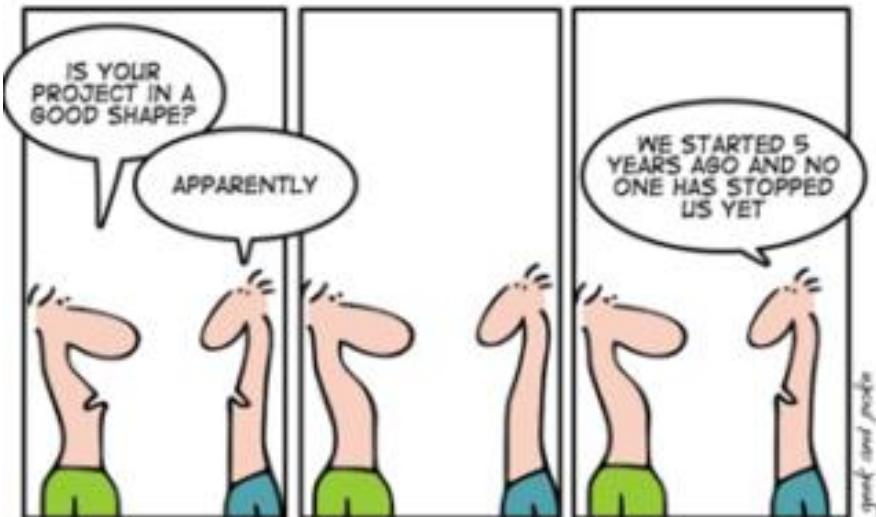


Bachelor-Modul: Software Projekt **Einführung**

Ablauf und Aufgabe des Software Projektes

<https://confluence.swl.informatik.uni-oldenburg.de/display/SWP>



Marco Grawunder

- Grundlegendes zur Veranstaltung
- Aufgabenstellung
- Einzelaufgaben
- Termine, Anmeldung...
- User-Stories

- Achtung:

- Folien enthalten **nur essentielle Informationen**
- Wiki im Confluence mit wichtigen weiteren Informationen
- Unbedingt **mindestens einmal hineinschauen!**
- Bei Widersprüchen zählen die Folien
- Bei Fragen/Unklarheiten bitte melden

<https://confluence.swl.informatik.uni-oldenburg.de/display/SWP>

The screenshot shows a web browser window with the URL <https://confluence.swl.informatik.uni-oldenburg.de/display/SWP>. The page is titled "Software Projekt Home". On the left, there is a sidebar with navigation links: "Seiten", "Blog", "Kalender", "BEREICHSVERKNÜPFUNGEN" (with a note about linking to important pages), "SEITENHIERARCHIE" (with links to "Organisation", "Aufgabenstellung", "Bewertung", "HowTo", "Historisches", and "FAQ"), and "Bereich konfigurieren". The main content area contains sections: "Hinweise" (with red bullet points about group division and meetings), "Ziele" (with text about the goal of the project), "Voraussetzungen" (with text about prerequisites and participation requirements), and a footer section with a link to "Organisation".



- Sichere Kenntnisse aus den Veranstaltungen **PDA** und **OMP**
- Insbesondere die **Programmierkenntnisse in Java**
- Ohne Programmieraufgaben kann das SWP nicht erfolgreich abgeschlossen werden
- Kenntnisse in
 - Informationssystemen
 - Software Technik

müssen im Laufe des Semesters in den jeweiligen parallel liegenden Veranstaltungen gewonnen werden

- **Verbindliche Anmeldung für das Modul bis zum 01.12. (bzw. eine Woche vorher)**
- Vorher einige Übungsaufgaben, die keine unüberwindbaren Hürden sein sollten!





جديد

NEU!

НОВЫЙ

新

NEW

新しいです



NUEVO!

Es ändert sich, vor allem zu Beginn, einiges im SWP!
... später mehr dazu

Das heißt aber leider u.U. auch





Hoffentlich passiert das nicht ...

- **Programmieren im Großen:** Einüben von Methoden, Techniken und Hilfsmitteln des Software Engineering
 - methodische SW-Entwicklung (Software-Lebenszyklus)
 - Qualitätssicherung
 - Projektmanagement
- **Software Entwicklung im Team**
 - Gruppenarbeit, Projektarbeit, Teamgeist → gemeinsame Lösung
 - Schulen der kommunikativen Fähigkeiten
 - Zusammenarbeit mit (anfangs) fremden Personen
- **Bewusstsein entwickeln** für längerfristige **Konsequenzen** von Entscheidungen
 - Aber: „Es ist keine Schande, Code wegzwerfen“
 - Arbeiten unter **Zeitdruck** für längere Zeit
- **Einarbeitung in existierende Systeme/Frameworks**
 - Vorgehen erlernen



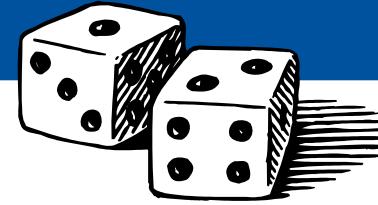
- Gruppensitzungen in Teams (verpflichtend!)

- selbst organisiert und betreut durch Tutor
- wöchentliches Treffen mit dem Tutor
- Untergruppen für Tickets bilden
 - Mindestens **ein weiteres** Treffen der Teilgruppen notwendig
 - Dokumentiertes (!) Pairprogramming
- Größe 8 – 1213



Tipp: Führen Sie teambildende Maßnahmen durch





- alle

- sind **vorbereitet** und nehmen **aktiv** teil
- erledigen ihre Aufgaben **vollständig** und **pünktlich**
- **übernehmen aktiv Aufgaben**
- beachten Gruppenstandards
- sind bereit, ihre Arbeit auf **Fehler** untersuchen zu lassen
 - Grundsätzliche Kritik bitte zunächst unmittelbar an eine Person, dann Gruppe, dann Tutor, dann Dozent
 - Es gibt in jeder Gruppe einen **Konfliktbeauftragten**
 - **Konsequenzen** bei Nichterfüllung von übertragenen Aufgaben festlegen?

- **Jeder übernimmt Teile bei**

- Dokumentation und
- Implementierung
- keine „Doku-Sklaven“!

NEU!



Software Projekt

- Professionelles Ticketsystem mit Scrum-Unterstützung



<http://www.atlassian.com/de/software/jira/overview>

- Zentrale Installation unter: <https://jira.swl.informatik.uni-oldenburg.de>
- Viele Plugins
- Zur Erstellung von User Stories, Tasks und Bugs

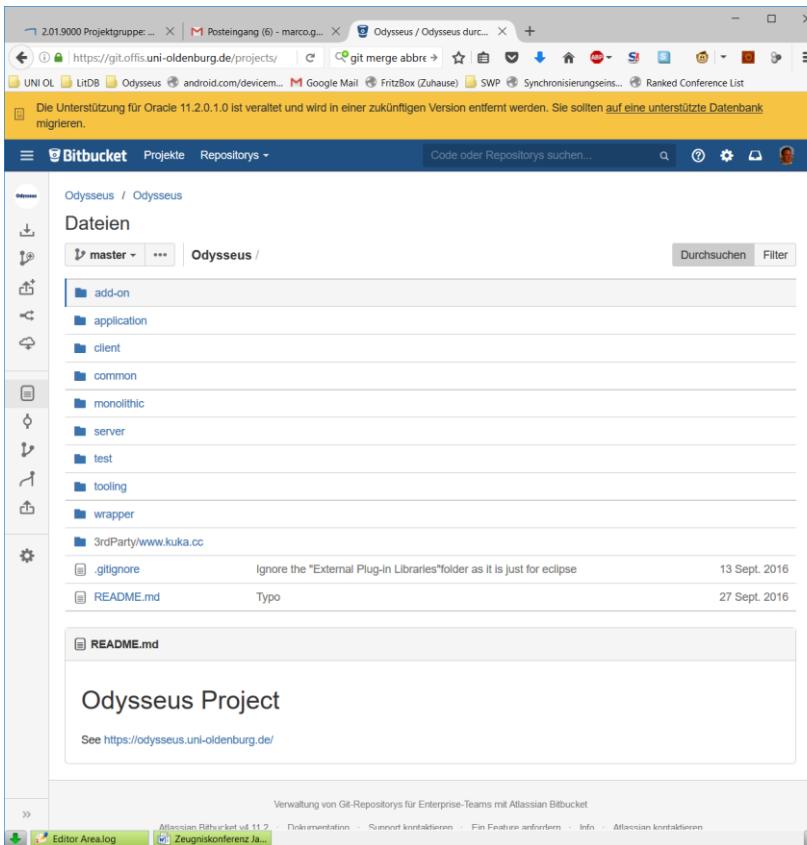
The screenshot shows the JIRA Activity Stream for the Odysseus project. It displays a timeline of events, including commits from Jan Sören Schwarz and Christian Kuka, and changesets from Timo Michelsen. The interface includes sections for 'Assigned to Me' and 'PROJECT SHORTCUTS'.

The screenshot shows the JIRA Issues screen for the Softwareprojekt - Agile Board project. It lists 17 tasks assigned to 'ba14'. The tasks are categorized into 'Active sprints' (Sprint 9 - Tomorrowland) and 'Other issues'. Each task has a detailed description and a small icon indicating its status or type.

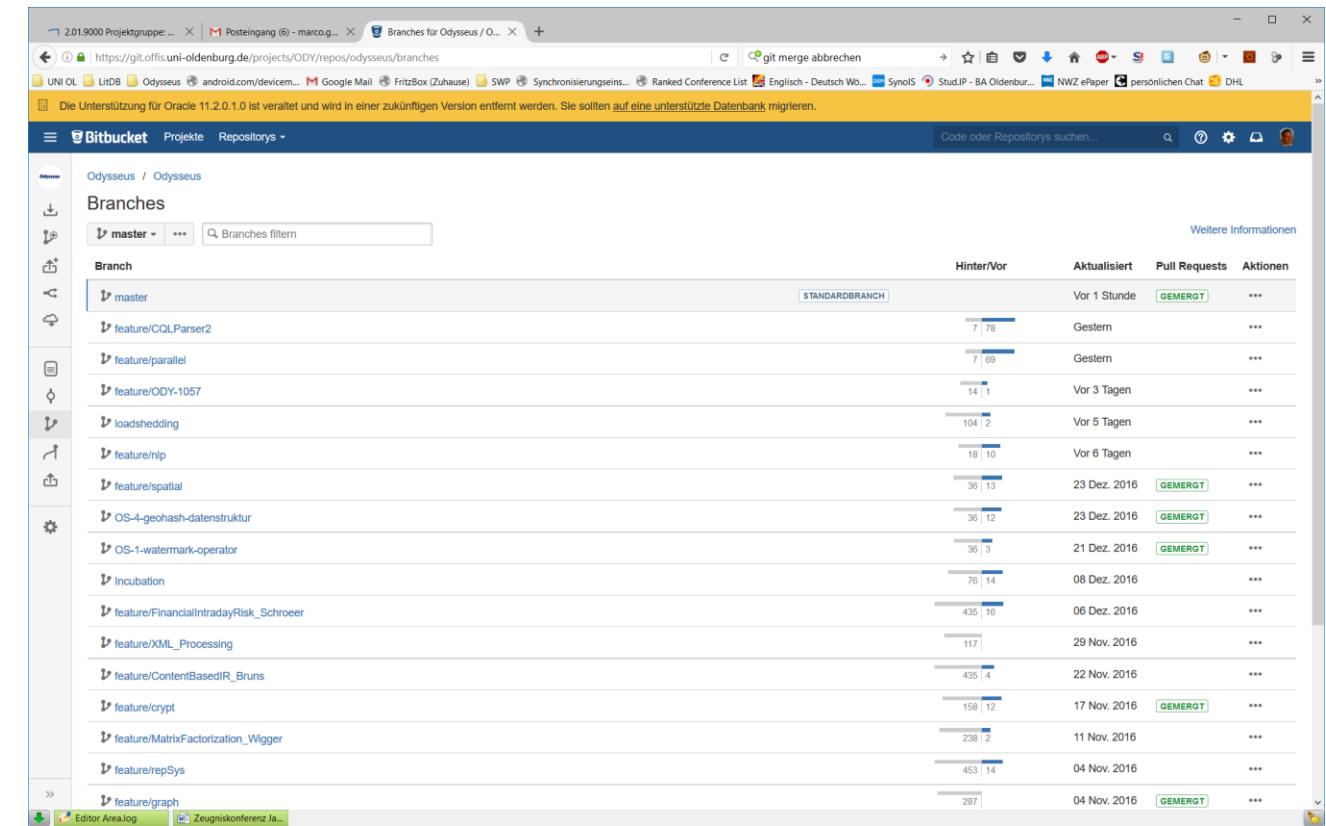


- Wiki
- Zentrale Installation unter: <https://confluence.swl.informatik.uni-oldenburg.de/>
 - Für alle Dokumente geeignet (z.B. Protokolle, Tutorials, etc.)
 - Auch die Dokumentation kann man hier machen, sieht in LaTeX aber natürlich besser aus
 - Initial: Eine Seite mit wichtigen projektrelevanten Basisinformationen inkl. **Gruppenfoto** (und Namen) anlegen!
- Logins sind für alle Atlassian Tools, man muss sich aber bei jedem Tool anmelden

- Quellcodemanagement basierend auf Git (ähnlich zu GitHub)
- **Branches und Pull Requests** verwenden! siehe auch folgende VLs
- z.B. Aus Jira(-Ticket) einen Branch anlegen
- <http://git.swl.informatik.uni-oldenburg.de/projects/>



The screenshot shows the Bitbucket web interface for a repository named 'Odysseus'. The left sidebar lists branches: master, feature/CQLParser2, feature/parallel, feature/ODY-1057, loadshedding, feature/nip, feature/spatial, OS-4-geohash-datenstruktur, OS-1-watermark-operator, Incubation, feature/FinancialIntradayRisk_Schroerer, feature/XML_Processing, feature/ContentBasedIR_Bruns, feature/crypt, feature/MatrixFactorization_Wigger, feature/repSys, and feature/graph. The main area displays the contents of the 'master' branch, including files like add-on, application, client, common, monolithic, server, test, tooling, wrapper, .gitignore, and README.md. A message at the bottom indicates support for Oracle 11.2.0.1.0 is deprecated.



The screenshot shows the Bitbucket interface displaying a list of branches for the 'Odysseus' project. The branches listed are: master, feature/CQLParser2, feature/parallel, feature/ODY-1057, loadshedding, feature/nip, feature/spatial, OS-4-geohash-datenstruktur, OS-1-watermark-operator, Incubation, feature/FinancialIntradayRisk_Schroerer, feature/XML_Processing, feature/ContentBasedIR_Bruns, feature/crypt, feature/MatrixFactorization_Wigger, feature/repSys, and feature/graph. Each branch entry includes a status bar indicating the last update time (e.g., 'Vor 1 Stunde', 'Gestern', 'Vor 3 Tagen', etc.) and a 'MERGED' button if applicable.

Confluence Bereiche Personen Umfragen Kalender Erstellen ... Suchen ? 🔍 Bearbeiten Favorit Beobachten Teilen ...

Seiten SWP2019A

Seiten Blog Kalender

BEREICHSVERKNÜPFUNGEN

Hier können Sie Verknüpfungen zu Ihren wichtigsten Inhalten vornehmen.

SEITENHIERARCHIE

Starten Sie mit dem Hinzufügen einiger Seiten zu diesem Bereich. [Seite erstellen](#).

SWP2019A Home

Erstellt von Marco Grawunder vor 32 Minuten

Welcome to your new space!
Confluence spaces are great for sharing content and news with your team. This is your home page. Right now it shows recent space activity, but you can customize this page in any way you like.

Complete these tasks to get started

- Edit this home page** - Click *Edit* in the top right of this screen to customize your Space home page
- Create your first page** - Click the *Create* button in the header to get started
- Brand your Space** - Click *Configure Sidebar* in the left panel to update space details and logo
- Set permissions** - Click *Space Tools* in the left sidebar to update permissions and give others access

Recent space activity

 **Marco Grawunder**
[SWP2019A Home](#) vor 32 Minuten erstellt

Space contributors

• Marco Grawunder (vor 32 Minuten)

Gefällt mir Sei der Erste, dem dies gefällt. Keine Stichwörter

Schreiben Sie einen Kommentar...

Diese Site wird mit einer kostenlosen **Atlassian Confluence Community-Lizenz** betrieben, die University of Oldenburg gewährt wurde. Confluence heute testen.
Auf dieser Confluence-Installation wird läuft eine kostenlose Gliffy-Lizenz - Bewerten Sie das Gliffy Confluence-Plug-In für Ihr Wiki!

Powered by Atlassian Confluence 6.15.9 · Fehler/Anregungen · Atlassian-News

Bereich konfigurieren « ATTLASSIAN Software Projekt

Registrierungsmail (im Laufe des Tages)

The screenshot shows an email client interface with the subject "[JIRA-SWP] Account created". The recipient is Marco.Grawunder@uni-oldenburg.de, and the message was sent at 11:01 (vor 3 Minuten). The email body contains a JIRA account creation notification for Marco Grawunder. It includes a screenshot of the JIRA login page with the following details:

Username	mgrawunder2
Email	Marco.Grawunder@gmail.com
Full Name	Marco Grawunder

The message continues with instructions to log in and set a password, followed by a "Set my password" button. A note states that the password can be set within 24 hours or requested later. At the bottom, there is a footer note about JIRA administration and a link to the JIRA software page.

This message is automatically generated by JIRA.
If you think it was sent incorrectly, please contact your JIRA administrators
For more information on JIRA, see: <http://www.atlassian.com/software/jira>

- Username ist E-Mail-Adresse
- Bitte einloggen und Passwort setzen
- Bei Problemen bitte melden, vor allem, wenn keine Mail ankommen sollte

- Gruppenverzeichnis bei ARBI (Rechner: Dümmer):
 - /user/proj/sp<JJ>/sp<JJ>gX
 - Liste der Teilnehmer einer Gruppe → Jörg Lehners
 - MailListe: sp<JJ>gX@Informatik.Uni-Oldenburg.DE
- siehe Hinweise im Confluence-Wiki **und** im Stud.IP-Wiki dazu
- <JJ> → aktuelles Jahr ☺

Bewertung: Gruppenleistung

- Gruppennote ist die **Basisnote** für alle Teilnehmer der Gruppe
- Individualbewertungen führen zur Modulnote
- enthält:
 - Produkt, d.h. Software und zugehörige Dokumentation (wie Installationsanleitung) (ca. 50 %)
 - Testen inkl. Testdokumentation (ca. 10 %)
 - Dokumentation des SWPs (u.a. Entwurf) (ca. 20 %)
 - Vorgehen und Arbeiten im Team,
Kommunikation, Projektmanagement (ca. 10 %)
 - Präsentationen/Abnahme (ca. 10 %)

**VERÄNDERTE
REZEPTUR**



- <https://confluence.swl.informatik.uni-oldenburg.de/display/SWP/Gruppennote>
- Dokumentationsanforderungen:
<https://confluence.swl.informatik.uni-oldenburg.de/display/SWP/Dokumentationsanforderungen>



- Gruppensitzungen (**Anwesenheitspflicht!**)

- Regelmäßige und aktive Beteiligung
- >3 x unentschuldigt → keine weitere Teilnahme möglich!
- >6 x insgesamt fehlen → Termin bei mir!
bezogen auf beide Semester!



- Einzelleistungen und Einzelaufgaben (siehe später)

NEU!

- Abzugebende **Übungsaufgaben** zu Beginn des SWPs
- Erstellung Software und Dokumenten (Test, Präsentation von Ergebnissen, Einhaltung von Standards/Richtlinien, ...), Engagement, Moderation
- im Rahmen des Projekts, u. a. Projektplanung
- <https://confluence.swl.informatik.uni-oldenburg.de/display/SWP/Bewertung%3A+Einzelleistung>



Bewertung: Einzelleistung

- Mehrere (4) Individualeinschätzungen im Laufe der Zeit durch Tutoren

Einschätzung	Beschreibung
Ü	Leistung im Verhältnis zur Gruppe überdurchschnittlich
D	Leistung liegt auf dem Durchschnittsniveau der Gruppe
U	Leistung liegt unterhalb des Niveaus der Gruppe, aber noch akzeptabel
N	Leistung inakzeptabel, letzte Warnung (i.d.R. gibt es keine zwei Ns nacheinander)
F	(Failed) Nicht ausreichend, keine erfolgreiche Teilnahme mehr möglich (kann auch direkt vergeben werden!)

**VERÄNDERTE
REZEPTUR**



- Beziehen **sich immer auf die Gruppe**, d.h. gruppenübergreifend nicht vergleichbar!
- Die Einschätzungen sind **KEINE Noten** und werden durch die Tutoren mit den Mitgliedern der Gruppe besprochen
- Ü und U können **nah am Durchschnitt liegen oder weit entfernt** (mit Tutor sprechen)
- Für die finale Note (durch mich) spielen auch die Tendenz und ggf. ausgewöhnliche Umstände eine wichtige Rolle!
- <https://confluence.swl.informatik.uni-oldenburg.de/display/SWP/Bewertung%3A+Einzelleistung>

- „ein bestimmter **Minimalaufwand** darf nicht unterschritten werden, wenn die Studierenden wirklich **verstehen sollen**, um welche **Probleme** es bei der Projektleitung geht“

Aus Zehnder: Führung von Informatikprojekten durch Studentengruppen

- Hinweis: SWP hat 3+6 KP

→ (ca.) $90\text{ h} + 180\text{ h} = 270\text{ h}$ pro Person

→ bei 10 Personen pro Gruppe: $10 * 270\text{ h} = 2700\text{ h}$

→ Nur wenn alle (in die selbe Richtung) mitziehen, kann das SWP mit einigermaßen vertretbarem Aufwand durchgeführt werden

Teamwork ist EXTREM WICHTIG!

→ Wichtig: Planung was getan wird und was nicht (100 % in allen Übungsblättern schafft man auch nicht mit der normalen Stundenzahl)

→ Programmierung verleitet häufig dazu mehr zu tun ... ;-)

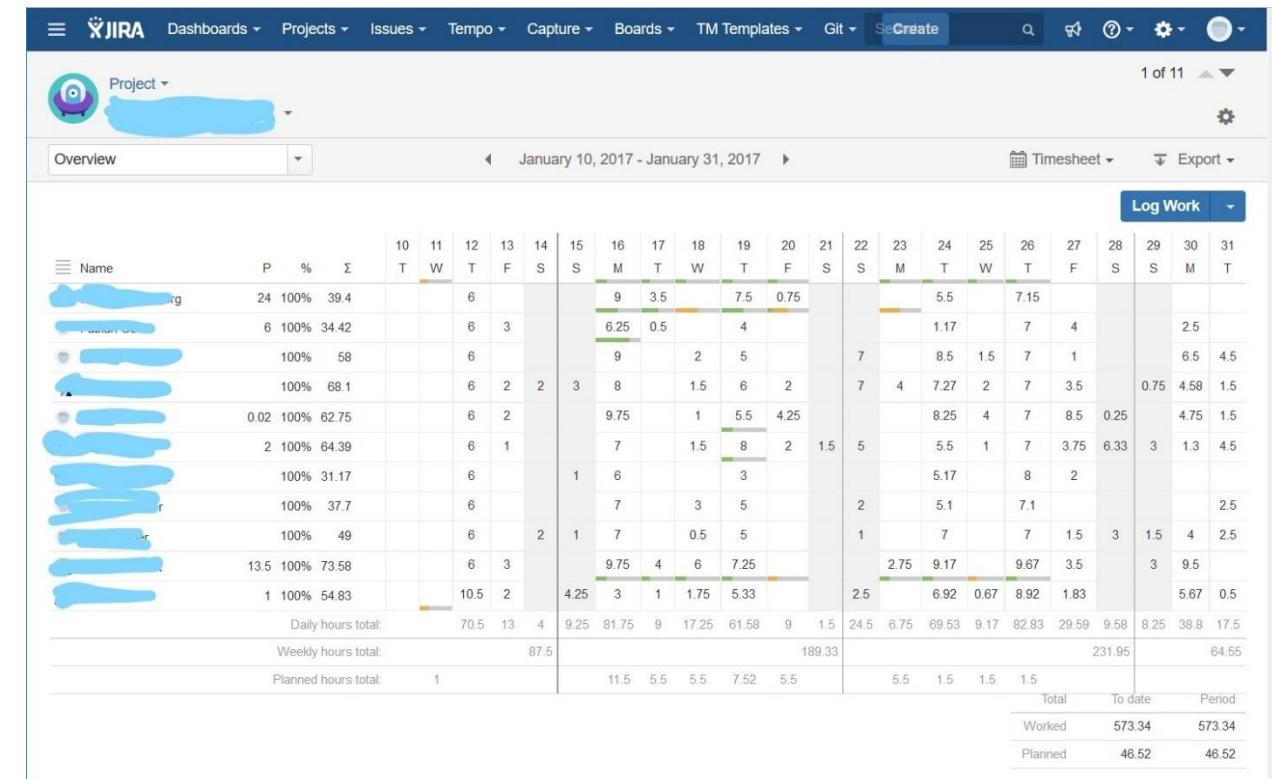
- Trotzdem natürlich:

- Leistung = Arbeit/Zeit :-)

- d.h. nur viel Arbeit bringt alleine keine gute Note!



- wichtig für eine gerechte Leistungsbeurteilung und Arbeitsverteilung in der Gruppe
- ALLES was gemacht wurde (auch die Sitzungen)
- Mit **Tempo Plugin** in Jira
- Möglichst wenig auf allgemeine Tasks (wie Gruppensitzung) buchen
- Zeiteinheiten beachten und angeben
- Statt 0,75 Stunden besser 45 Minuten
- Am besten sofort buchen (später vergisst man das).
- Es gibt Hilfsmittel zum Timetracking



- Iterative Entwicklung einer Software im Team
- Phase 1: Einarbeiten und Verstehen (Meilenstein 1)
 - Vorgegebene Architektur verstehen
 - Grundlegende Konzepte verstehen
- Phase 2: Autonome Durchführung in den Gruppen
 - Meilenstein 2
 - Meilenstein Prototyp
 - Endpräsentation

- Es gibt eine Reihe **vordefinierter Meilensteine**, i.d.R. durch Präsentation abgeschlossen
- Erster Meilenstein jetzt **gemeinsam**, keine Präsentation mehr
 - User Management: Registrieren, Einloggen, Ausloggen und UserRepository
 - Hauptmenü inkl. Liste eingeloggter Nutzer
- Erste Präsentation: (16.–20.12.20)
 - Chat (im Hauptmenu und im Lobby bzw. Spielfenster)
 - Ansätze des Spiels, einfaches Spielen z.B. über die Console
- Zweite Präsentation: (25.-30.5.20)
 - Prototyp, mit Speicherung in relationaler Datenbank
 - Feature Complete
- Endpräsentation: (1.7.-15.7.20)
 - Achtung! Zum Ende des Sommersemesters!

NEU!



- **Wichtig:**
 - Bitte jetzt schon Termine für die erste Präsentation machen
 - Bitte drei Terminvorschläge (möglichst an unterschiedlichen Tagen machen)



- CapGemini, Hamburg
- 10.12.2019
- Teilnehmeranzahl beschränkt auf 30 Personen (Anmeldung ab morgen über Gruppe im Stud.IP)
- Selbstanreise mit Bahn/Studi-Ticket, Treffen in HH am Bahnhof (Genaueres wird noch bekannt gegeben)
- Inhalt:
 - Wie sieht die Arbeit eines Informatikers bei CapGemini aus
 - Wie läuft Software Entwicklung in der Industrie
 - Vorstellung von Projekten dort





- Software Entwicklung findet i.d.R. nicht mehr „auf der grünen Wiese“ statt
 - **Basissystem** wird vorgegeben
 - es ist erlaubt (und teilweise notwendig), das Basissystem zu verändern
 - Aufbauend auf dem Basissystem sollen Erweiterungen erfolgen
- In den ersten Wochen müssen **ALLE** unabhängig von der Gruppe „Übungsaufgaben“ durchführen (z.B. kleinere Implementierungen auf dem Basissystem)
- **Danach** findet die **verbindliche Anmeldung** zum Software Projekt statt
- **Informationssysteme parallel**
 - Speicherung von Nutzerdaten zunächst nur im Hauptspeicher (im Basissystem umgesetzt)
 - Später (spätestens zum Prototyp) ersetzen durch relationale Datenbank
- **Software Technik parallel**
 - Erster Sprint wird gemeinsam durchgeführt
 - Weitere Vorlesungen sollen helfen, einen Einstieg zu bekommen
 - Inhalte hier immer nur zum „Reinkommen“, in ST bzw. **Selbststudium** vertiefend

NUEVO!



- Scrum: Das Vorgehensmodell im SWP
- Anforderungsermittlung mit User Stories
- Die Atlassian-Tools:
 - Jira (Steuerung des Projektes)
 - Versionskontrolle mit Git und Bitbucket, u.a. Git Workflow
- Vorstellung des Basissystems
- Maven
- Model-View-Presenter (MVP) mit JavaFX
- Dependency Injection
- Testen mit JUnit und Mockito
- Ansätze für sauberen Code und saubere Architekturen, Code Smells
- Netzwerkkommunikation
- (ggf.) Entwurfsmuster

Fragen bis hierher?



Thema

- Bei allen Spielen gilt:
 - Keine Freigabe vom Verlag → nicht mit Originalgrafiken online stellen
 - Umsetzung (von Teilen) des Basisspiels
 - Regeln dürfen vereinfacht und angepasst werden
 - Adaption grundsätzlich erlaubt, bedeutet aber zusätzlicher Aufwand (... ohne, dass sich das deutlich in der Bewertung bemerkbar macht. Software Projekt, nicht Spiele Projekt ;-))
- Wie kann man das Brettspiel möglichst so nachbilden, dass der **Spielspaß** der Originalversion erhalten bleibt/verbessert wird? („Mogeln gehört nicht zum Spielspaß ;-“)
- Kann man **zusätzliche** Dienste bieten?
- Müssen gewisse Aspekte **angepasst** werden?
- Sehr wichtig: Für die Mitspieler auf **Nachvollziehbarkeit** achten
→ z.B. was für Aktionen hat der Gegner vorgenommen



Bilder: Hans im Glück Verlag / Rio Grande Games
<http://www.dominion-welt.de/>

- Kartenspiel für 2-4 Spieler (mit Erweiterung für 6 Spieler) ab 8 Jahre
 - Deutschland: Hans im Glück Verlag (für Rio Grande Games)
 - Spiel des Jahres 2009
 - Basisspiel: Was für eine Welt
 - Viele Erweiterungen: u.a. die Intrige, Seaside
-
- Achtung! Keine offizielle Freigabe. Keine eigene Veröffentlichung des Spiels möglich.
 - Im folgenden Erklärungen für die ursprüngliche Version
 - Am Ende Änderungen bzgl. der neuen Version
 - Es spielt keine Rolle, welche Version umgesetzt wird

- Am Ende des Spiels möglichst viele Reichtümer (Punkte) angesammelt zu haben
- Punkte gibt es mit folgenden Punktekarten (Siegpunktkarten):



- Also: Einfach Punktekarten sammeln ☺ ... ganz so einfach ist es dann doch nicht ;-)

Kartenarten

- **Punktekarten** zählen am Ende, im Spiel sind sie hinderlich (man kann mit ihnen nichts machen)



- Mit **Geldkarten** kann man Karten kaufen (Kupfer, Silber, Gold)



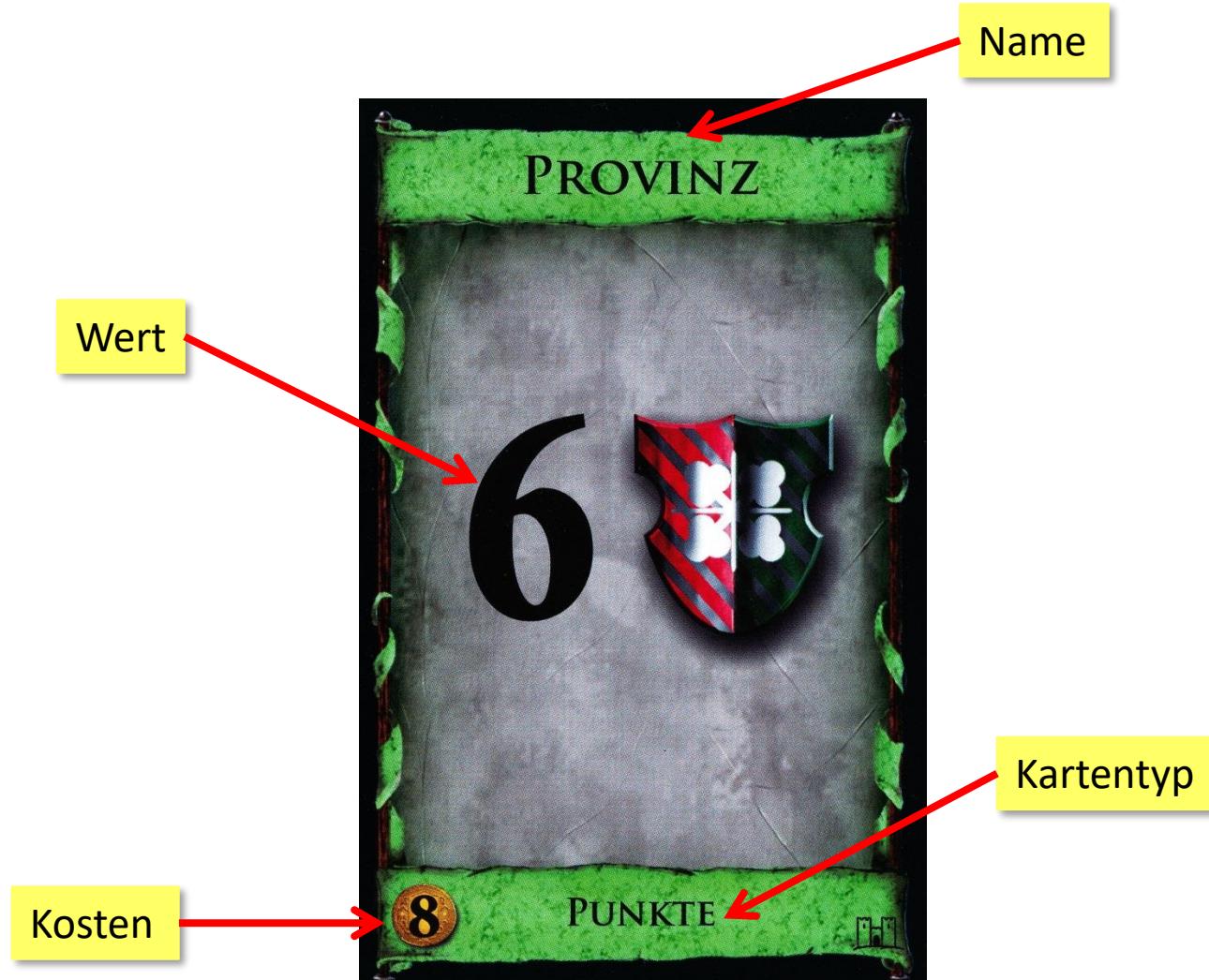
- **Aktionskarten** (auch Königreichskarten) dienen dazu, „Action“ in Spiels zu bringen
- Spiele unterscheiden sich durch unterschiedliche Aktionskartensätze



Aufbau einer Geldkarte



Aufbau einer Punktekarte



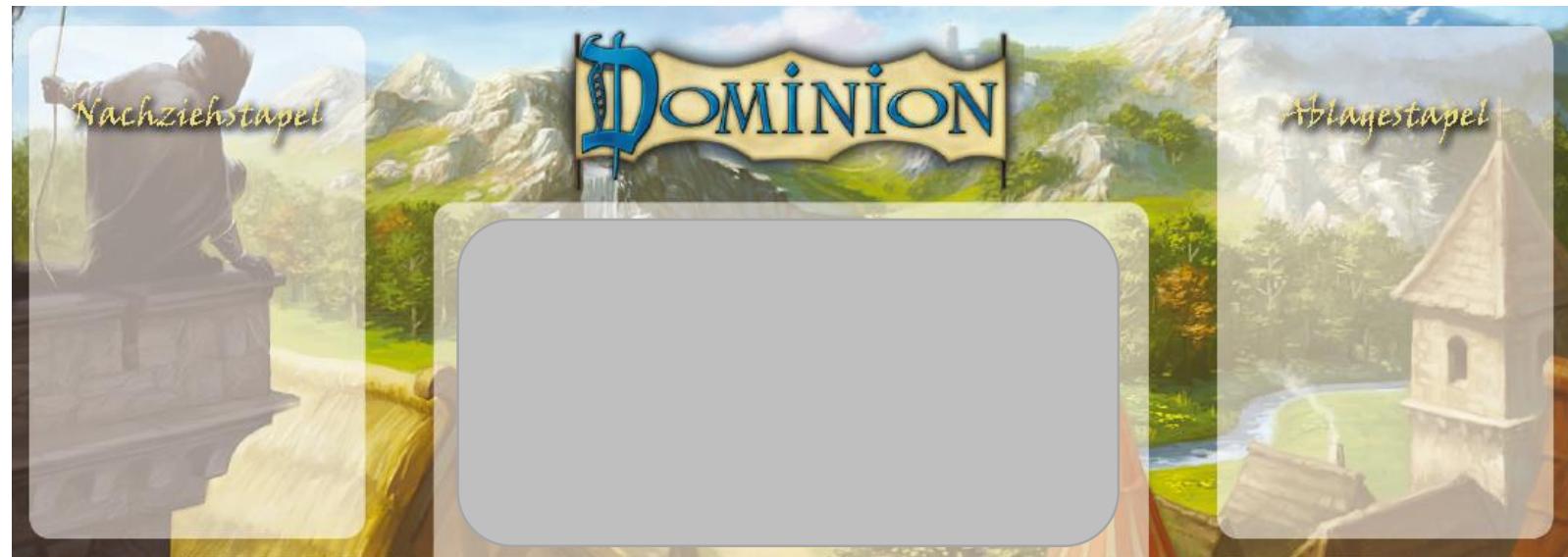
Aufbau einer Aktionskarte



Jeder Spieler



- bekommt 7 Kupferkarten und 3 Anwesenkarten
- mischt **seinen Stapel** und legt ihn verdeckt neben sich (**Nachziehstapel**)
- Von diesem Stapel zieht nun jeder Spieler 5 Karten auf die Hand (**Handkarten**)



- In die **Mitte des Tisches** kommt der **Vorrat**:
 - (Spieleranzahlabhängige) Anzahl von Geld-, Punkte- und Fluchkarten
 - 10 Arten von Aktionskarten (mit je 10 Karten, Ausnahme: Garten)
- Die Müllkarte: Der Platz für entsorgte Karten





- Dominion wird in Runden gespielt
- Spieler an der Reihe (i.d.R. 5 Handkarten) führt Zug aus
- Drei Phasen:
 1. Aktion: Der Spieler **kann** Aktionskarten ausspielen
 2. Kauf: Der Spieler **kann** vom Vorrat Karten kaufen.
 3. Aufräumen: Der Spieler **muss** alle seine (gespielte und nicht gespielte) Karten ablegen (→ offen auf den Ablagestapel) und 5 Karten für den seinen nächsten Zug nachziehen



- Spieler darf Aktionskarten auslegen
- Normalerweise nur eine Aktion pro Zug, aber Aktionskarten können die Anzahl möglicher Aktionen erhöhen
- Ausspielen:
 - Offen vor sich hinlegen
 - Alle Anweisungen von oben nach unten ausführen (auch wenn einzelne Anweisungen nicht möglich sind)
 - Alle Anweisungen müssen vor nächster Aktion (z.B. weitere Aktionskarte) vollständig erledigt sein
- Beispiele für Aktionen:
 - Karten vom Nachziehstapel auf die Hand ziehen
 - Karten entsorgen (aus dem Spiel nehmen) oder ablegen (auf den eigene Ablagestapel)
 - Neue Karte aus dem Vorrat nehmen



Kaufphase

- Spieler **kann** Karten aus dem Vorrat kaufen
- Normalerweise **eine Karte pro Zug**, aber Aktionskarten können die Anzahl möglicher Käufe erhöhen
- Spieler **muss** Geldkarten für den Kauf vor sich auslegen
- Aktionskarten **können** mit virtuellem Geld den Geldvorrat für diesen Zug erhöhen
- Gekaufte Karten kommen auf den Ablagestapel (können also in dieser Runde nicht genutzt werden)
- Bei mehreren Käufen können die auslegten Karten und das virtuelle Geld beliebig auf die Käufe verteilt werden







- Aktionen sind die Dinge, die die spielerische Umsetzung schwer machen
- Es gibt ganz unterschiedliche Arten von Aktionen
 - Aktionen, die **nur den aktuellen Spieler** betreffen und sich auf normale Aktionen wie nachziehen oder ablegen beziehen
 - Aktionen, die **Interaktionen mit anderen Spielern** beinhalten und u.U. sogar zu Regeländerungen führen
- Für das SWP muss nur ein Spiel mit allen Aktionen aus 1 und mindestens der Hälfte der Aktionen aus 2 (siehe folgende Folien) möglich sein. Andere Aktionen sind optional (bringen aber natürlich Zusatzpunkte ... und zusätzlichen Spielspaß)
- Wichtig:
 - Identifizieren Sie eine Menge von (konfigurierbaren) **Basisaktionen**
 - Versuchen Sie **zunächst nicht, komplette Karten** zu umzusetzen, sondern einzelne Basisaktionen
 - Denken Sie an das Command- und Strategiepattern
- Scans der Aktionskarten im Stud.IP. Auf keinen Fall verbreiten!!

Aktionskarten Stufe 1 (Erste Edition)



Aktionskarten Stufe 2 (Erste Edition)



Aktionskarten Stufe 3 (oben) und 4 (unten) (Erste Edition)



Aktionskartenklassifikation (Erste Edition)

	Allein	Mit anderen Spielern
Einfach	Kapelle Dorf Holzfäller Kanzler Schmiede Jahrmarkt Laboratorium Markt	Burggraben Miliz Hexe Ratsversammlung
Komplex	Keller Werkstatt Festmahl Geldverleiher Thronsaal Umbau Bibliothek Mine Abenteurer	Bürokrat Dieb Spion



- Von der Erstausgabe wurden folgende Kartensätze aussortiert: Abenteurer, Dieb, Festmahl, Holzfäller, Kanzler und Spion.
- Die neuen Königreichkarten sind:
 - Händlerin: +1 Karte +1 Aktion +1 Geld für ausgespieltes Silber
 - Vasall: +2 Geld, oberste Karte ablegen (wenn Aktionskarte, dann ausführbar)
 - Vorbotin: +1 Karte +1 Aktion. Karte von Ablagestapel auf Nachziehstapel
 - Wilddiebin: +1 Karte +1 Aktion +1 Geld
 - Banditin: ein Gold vom Vorrat auf Ablagestapel. Mitspieler müssen ggf. Geld entsorgen
 - Torwächterin: +1 Karte +1 Aktion. Oberste zwei Nachziehkarten ansehen und beliebig ablegen, zurücklegen oder entsorgen
 - Töpferei: Karte im Wert von maximal fünf Geld auf die Hand nehmen, dafür eine beliebige Handkarte auf den Nachziehstapel legen
- Es ist für das SWP egal, welche Version umgesetzt wird

Aktionskarten Stufe 1



Aktionskarten Stufe 2



Aktionskarten Stufe 3 (oben) und 4 (unten)

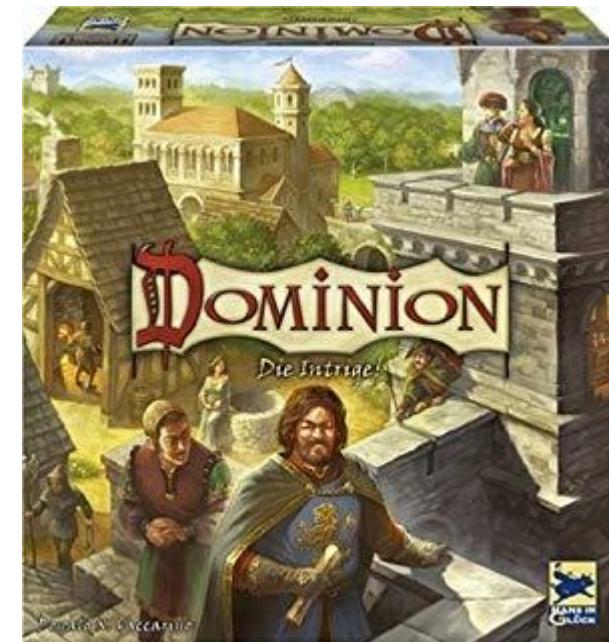
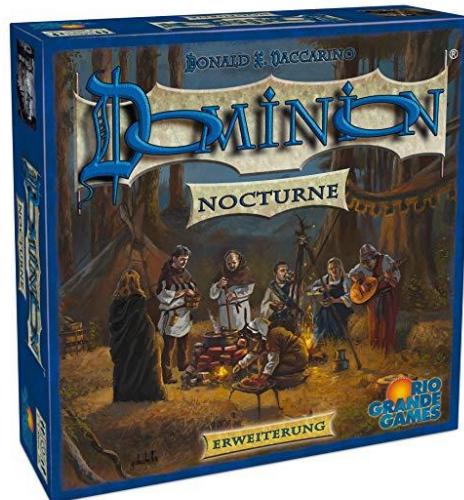
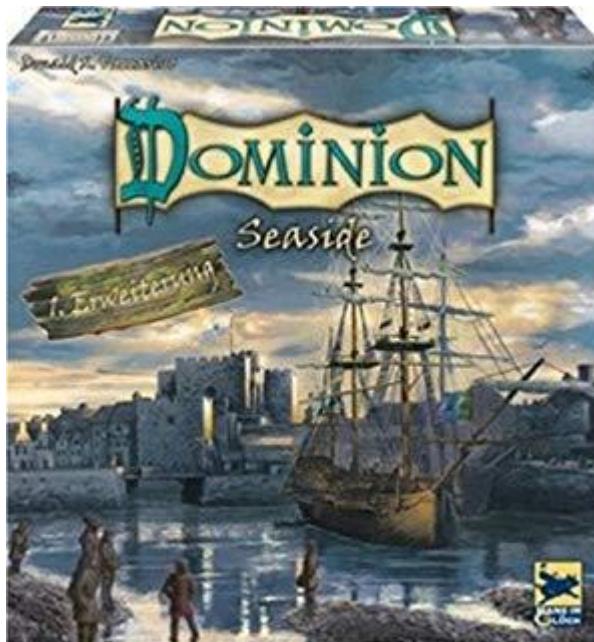


Aktionskartenklassifikation (bezieht sich auf die alte Version)



	Allein	Mit anderen Spielern
Einfach	Kapelle Dorf Schmiede Jahrmarkt Laboratorium Markt Händlerin Vorbotin	Burggraben Miliz Hexe Ratsversammlung
Komplex	Keller Werkstatt Geldverleiher Thronsaal Umbau Bibliothek Mine Vasal Wilddiebin Torwächterin Töpferin	Bürokrat

- Es sind grundsätzlich auch ganz andere Szenarien auf der selben Basisidee denkbar
- Achtung! Aufwand nicht unterschätzen!
- Andere Spielvarianten (z.B. Seaside, Die Intrige, etc.) → u.U. höherer Spielspaß, da höhere Interaktion → aber auch höherer Aufwand!



Fragen bis hierher?



Konkrete Aufgaben und Teilaufgaben



- Client-Server-Umsetzung (**Spiellogik auf Server!**): Komplett Java-(**11**)-basiert
- Basissystem wird vorgegeben
- (mindestens einfacher) KI-Spieler
 - Z.B. Zufallsbasiert nach den Regeln, KI-Spieler soll wie „normaler“ Spieler behandelt werden
- Aus didaktischen Gründen (nicht optional!):
 - „beliebig“ viele registrierte Nutzer
 - mehrere **Spielrunden** gleichzeitig im **Server**
 - Spieler können an mehreren Spielen vom selben Client aus gleichzeitig teilnehmen
- Nutzerverwaltung
 - gekapselter Zugriff (→ wichtig für das Testen)
 - Am Ende (Prototyp) relationales DBMS, zunächst Verwaltung der Nutzer im Hauptspeicher
 - einfache Austauschbarkeit der des DBMS-Servers (unterschiedliche Verbindungen, unterschiedliche Hersteller)
 - nur mit JDBC (kein Hibernate oder ähnliche OR-Mapper)



- Ansatz muss **Model-View-Presenter** Anforderungen erfüllen (siehe VL später dazu)
- Es muss mit möglichst wenig Aufwand möglich sein, andere Client-Arten anzubinden (z.B. Consolen-Anwendung) → u.a. Spiellogik muss auf Server sein
- Visuelles Thema/Design **darf** im vorgebenden Rahmen variiert werden (muss aber nicht!!)
- Achtung!
 - Wichtig ist guter, erweiterbarer und wartbarer Code
 - Wenn das Spiel dann **zusätzlich noch optisch** gut aus sieht, um so besser.
 - **Bedienbarkeit** ist aber trotzdem wichtig!
- Überlegen Sie genau, welche **Feature** Sie umsetzen können und welche weggelassen werden sollten
 - Lieber etwas weniger sehr gut umsetzen, als vieles schlecht! Qualität vor Quantität!
 - **Fokus** auf das Wichtige: z.B. keine Freundeslisten, private Chats oder 3D
 - Sie dürfen, wenn notwendig, die Regeln vereinfachen/verändern!
 - Sie haben nur **begrenzte Ressourcen** zur Verfügung und müssen das auch vertreten!

Frameworks nutzen!

- Abwägen zwischen „Selbermachen“ und „Einarbeitung“
- Vorgeschrieben (Basissystem):
 - Für den DB-Zugriff JDBC (nicht zu generische DB-Schnittstelle, **Kein Hibernate etc.**)
 - Kommunikation mit Netty
 - Client: JavaFX verwenden
- Viele Hilfsklassen bei Google Guava <https://github.com/google/guava>
 - U.a. **EventBus von Guava!** → im Basissystem
- Hier: <http://commons.apache.org/> gibt es viele nützliche Dinge (z.B. Mechanismen zum Logging oder für Konfigurationsdateien) → Log4J2 im Basissystem
- Testen mit Junit 5 → Bereits im Basissystem, GUI-Tests müssen per default **aus** sein
- Nutzen Sie ein Mocking-Framework (z.B. Mockito <https://github.com/mockito/mockito>)
- Das Basissystem ist mit Maven umgesetzt
- In den kommenden Wochen einige VL-Inhalte zu den obigen Themen

- TA 1: Projektmanagement organisieren

In jeder Iteration (jedem Sprint) ein Stück ergänzen!

- TA 2: Anforderungsanalyse

- TA 3: Entwurf

- TA 4: Implementierung/Test

Hin und wieder:

- TA 5: Zwischenpräsentation durch Gruppenabordnung

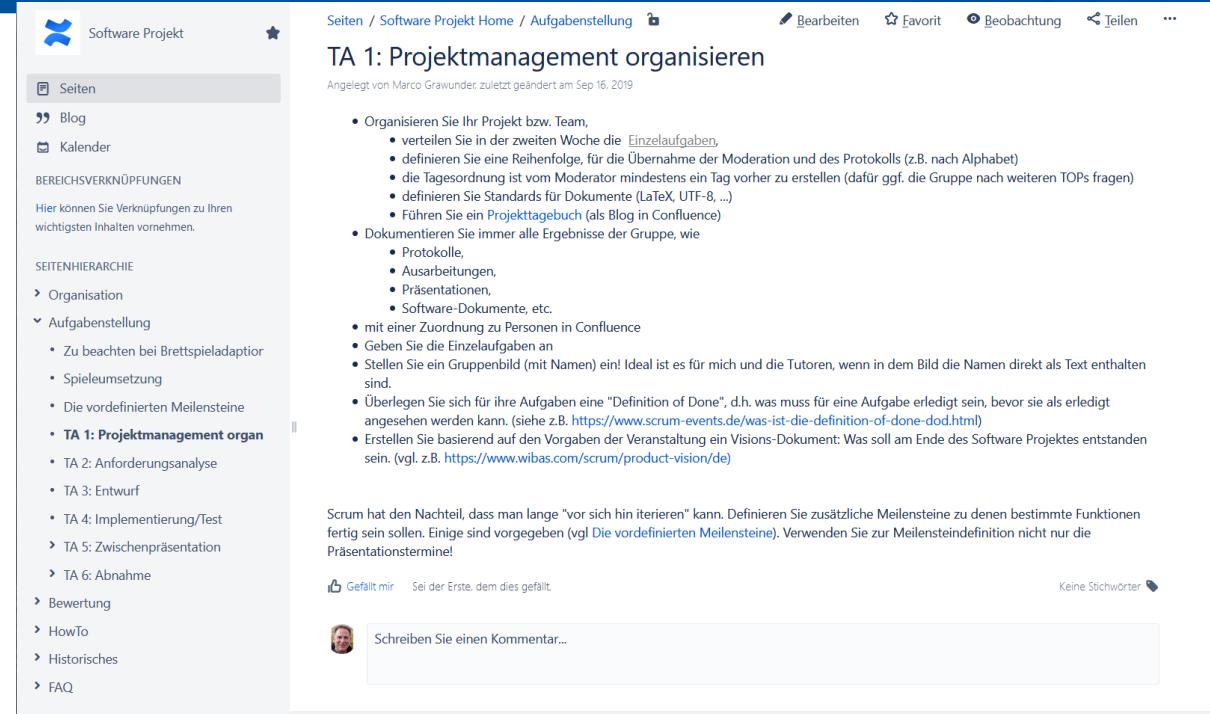
Am Ende:

- TA 6: Abnahme

- Hinweis:

- Software Design („UML“) ist wichtig! **Nicht erst am Ende machen!** Diagramme helfen bei der Kommunikation und beim Verständnis

- Doku zu Beginn nicht zu detailliert (eher stichpunktartig, aber auf jeden Fall nicht „liegen lassen“!)



The screenshot shows a Confluence page titled "TA 1: Projektmanagement organisieren". The left sidebar contains a navigation menu with links like "Seiten", "Blog", "Kalender", "BEREICHSVERKNÜPFUNGEN", "SEITENHIERARCHIE", and "FAQ". The main content area displays a list of tasks for organizing project management, such as defining roles, creating agendas, and documenting results. A note at the bottom discusses the Scrum iteration cycle.

Seiten / Software Projekt Home / Aufgabenstellung

TA 1: Projektmanagement organisieren

Angelegt von Marco Grawunder zuletzt geändert am Sep 16, 2019

• Organisieren Sie Ihr Projekt bzw. Team,

- verteilen Sie in der zweiten Woche die [Einzelaufgaben](#),
- definieren Sie eine Reihenfolge, für die Übernahme der Moderation und des Protokolls (z.B. nach Alphabet)
- die Tagesordnung ist vom Moderator mindestens ein Tag vorher zu erstellen (dafür ggf. die Gruppe nach weiteren TOPs fragen)
- definieren Sie Standards für Dokumente (LaTeX, UTF-8, ...)
- Führen Sie ein [Projekttagbuch](#) (als Blog in Confluence)

• Dokumentieren Sie immer alle Ergebnisse der Gruppe, wie

- Protokolle,
- Ausarbeitungen,
- Präsentationen,
- Software-Dokumente, etc.

• mit einer Zuordnung zu Personen in Confluence

- Geben Sie die Einzelaufgaben an
- Stellen Sie ein Gruppenbild (mit Namen) ein! Ideal ist es für mich und die Tutoren, wenn in dem Bild die Namen direkt als Text enthalten sind.

• Überlegen Sie sich für ihre Aufgaben eine "Definition of Done", d.h. was muss für eine Aufgabe erledigt sein, bevor sie als erledigt angesehen werden kann. (siehe z.B. <https://www.scrum-events.de/was-ist-die-definition-of-done-dod.html>)

• Erstellen Sie basierend auf den Vorgaben der Veranstaltung ein Visions-Dokument: Was soll am Ende des Software Projektes entstanden sein. (vgl. z.B. <https://www.wibas.com/scrum/product-vision/de>)

Gefällt mir Sei der Erste, dem dies gefällt. Keine Stichwörter

Wichtige Hinweise zu den Teilaufgaben im Confluence!



- unabhängig von eigentlicher Gruppenaufgabe
- durch **eine oder zwei** Personen (je nach Aufwand/Teilbarkeit) zu lösen.
 - Einarbeitung in das Thema
 - Präsentation des Werkzeugs/Ansatzes bzw. der Aufgabe in der Gruppe (**10 Minuten Kurz-Vortrag**)
 - **Jeder** muss einen Vortrag halten, Vorträge am besten am Anfang so schnell wie möglich durchführen. **ALLE VORTRÄGE IM ERSTEN SEMESTER!!**
 - Hochladen des Vortrags in Confluence
 - Fokus des Vortrags: Bezogen auf die Gruppe und das SWP
 - Erstellung von Musterdokumenten
 - Qualitätssicherung von Dokumenten
 - Beratung der Gruppenmitglieder
 - Über die gesamte Laufzeit des Projekts (evtl. Tausch im SoSe)
 - **Jeder** muss einmal Vortrag halten. Entweder Einzelaufgabe oder Präsentation von Ergebnissen von Teilgruppen
- **NEU!** Wahl der Einzelaufgabe in der **zweiten** Sitzung!



- **Scrum-Master:**
 - Sorgt dafür, dass der Scrum-Prozess am Laufen bleibt, leitet Daily Scrums, Reviews und Retrospektiven
 - Product Owner (etwas anders als in Scrum → Das Team zusammen mit den Tutor)
- **Jira und Projektplanung: (→ Jira Plugin: Portfolio)**
 - Projektplan erstellen, aktualisieren, Meilensteine überwachen, **Projekttagebuch** führen, Stundenpflege überwachen
- **Bitbucket/Git-Beauftragter**
 - Erstellen eines KM-Handbuchs/KM-Knigge → HowTo use Bitbucket, Pattern, Anti-Pattern, ...
 - Überwachung des Bitbucket-Workflows (Branches und Pull-Requests)
- **Testbeauftragter:**
 - Erstellung Testplanung, Musterdokumente, Vorlagen, Kontrolle der Tests, JUnit, Mockito
 - **Nicht dafür da, die Tests zu schreiben!**
- **Review- und Cruciblebeauftragter**
 - Sorgt dafür, dass in Bitbucket Reviews gemacht
 - Leitet mindestens ein Gruppenweites Codereview
- NEU!** • **Konfliktbeauftragter**
 - Informiert sich über Ansätze des Konfliktmanagements und steht ggf. als Mediator zur Verfügung



- **Codequalitätsbeauftragter:**
 - Überwachung von Codierungsstandards
 - Stellt „Code-Smells“ vor
 - Tools zur Überwachung der Codequalität z.B. SonarLint, Findbugs, Checkstyle
- **Patternbeauftragter (ggf. auch zwei)**
 - Wichtige Pattern (wie MVP, Observer, Command-Pattern)
 - Antipattern (wie Singleton)
- **Dokumentations- und Backupbeauftragter, Confluence, LaTeX-Beauftragter, Protokoll- und Dokumentverwaltung:**
 - Erstellung/Anpassung von Vorlagen, Hilfe
 - Musterdokumente, Standards, Ablagestrategie, Qualitätssicherung, Bereitstellungsstrategie
 - Sorgt dafür, dass die passenden Dokument erstellt, mitgepflegt und gesichert werden (kein Doku-S!)

Wenn alles von oben vergeben ist:

- Mavenbeauftragter
 - Sorgt dafür, dass sich alles (inkl. der Tests) problemlos mit Maven bauen lässt
- JavaFX und ControlsFX
- Google Guava
- Programmierparadigmen
 - z.B. Dependency Injection (mit Guice)
- DB-Zugriff:
 - Installation/Überwachung der DB
 - JDBC/Hibernate?
- Spezialisten für verschiedene Themen: Regeln, Weitere Frameworks, ...
- Weitere Infos: <https://confluence.swl.informatik.uni-oldenburg.de/pages/viewpage.action?pagId=393423>



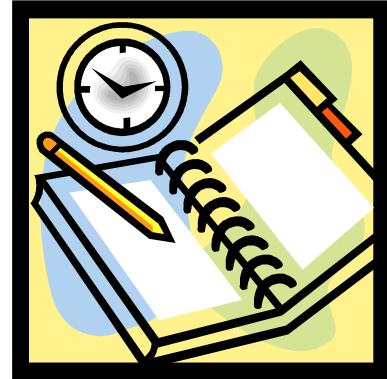
Diese Aufgaben wechseln wöchentlich:

- Sitzungsleitung
 - **Leitung** der Gruppensitzung (Wichtig!!), Support durch Scrum-Master
 - **Tagesordnung** definieren (vorher)!!
- Protokollführung
 - Erstellung eines Protokolls der Gruppensitzung
- **Zu Beginn jeder Gruppensitzung („Daily Scrum“):**
 - findet ein kurzes „**Briefing**“ („Blitzlicht“) statt, in der **jede/r berichtet**, was sie/er in der letzten Woche **für das Projekt** getan hat.
- **Am Ende jeder Sitzung neue Aufgabenverteilung („End Scrum“)**
 - Tickets (wer macht was)
 - Leitung/Protokoll (Liste!)
- Tutor ist nur Berater, leitet keine Sitzungen, kann aber um Rat gefragt werden (und greift ein, wenn es zu sehr „aus dem Ruder läuft“, erlaubt aber auch Fehler ...)



Gruppenanmeldungen

- Über Stud.IP (läuft bereits)
- Gruppe kann sich natürlich (nach Absprache mit Tutor) auch eigenen Raum/Termin suchen (ggf. über mich Anfrage an Raumbüro)
- Finale Vergabe gleich nach VL (ab ca. 14 Uhr)
- Jede Gruppe bekommt eine Stud.IP Veranstaltung (ich trage ein)
- Wechsel der Gruppe nur mit Tauschpartner oder wenn noch Platz in Gruppe **über mich**
- Beginn der Tutorien **MORGEN!**
 - Leider einige Gruppen kein erstes Treffen vor nächster VL
- Bei Problemen bitte schnell melden
- **WER IN MEHR ALS EINER GRUPPE IST, FLIEGT AUS BEIDEN RAUS!**



Fragen bis hierher?



Ich weiß ...



... gleich vorbei ...



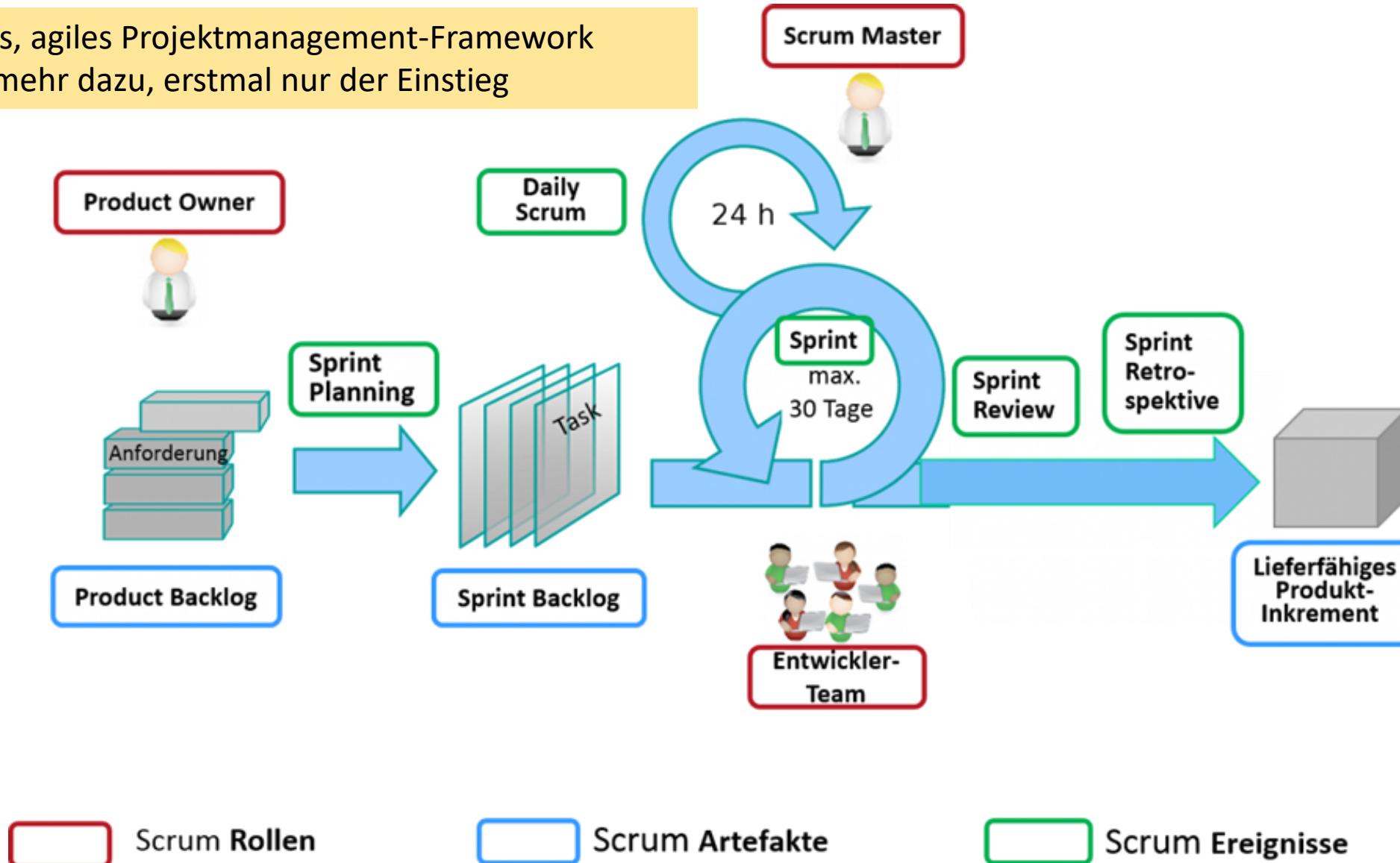
a alamy stock photo

JBKP0M
www.alamy.com

UND AB GEHT'S ...



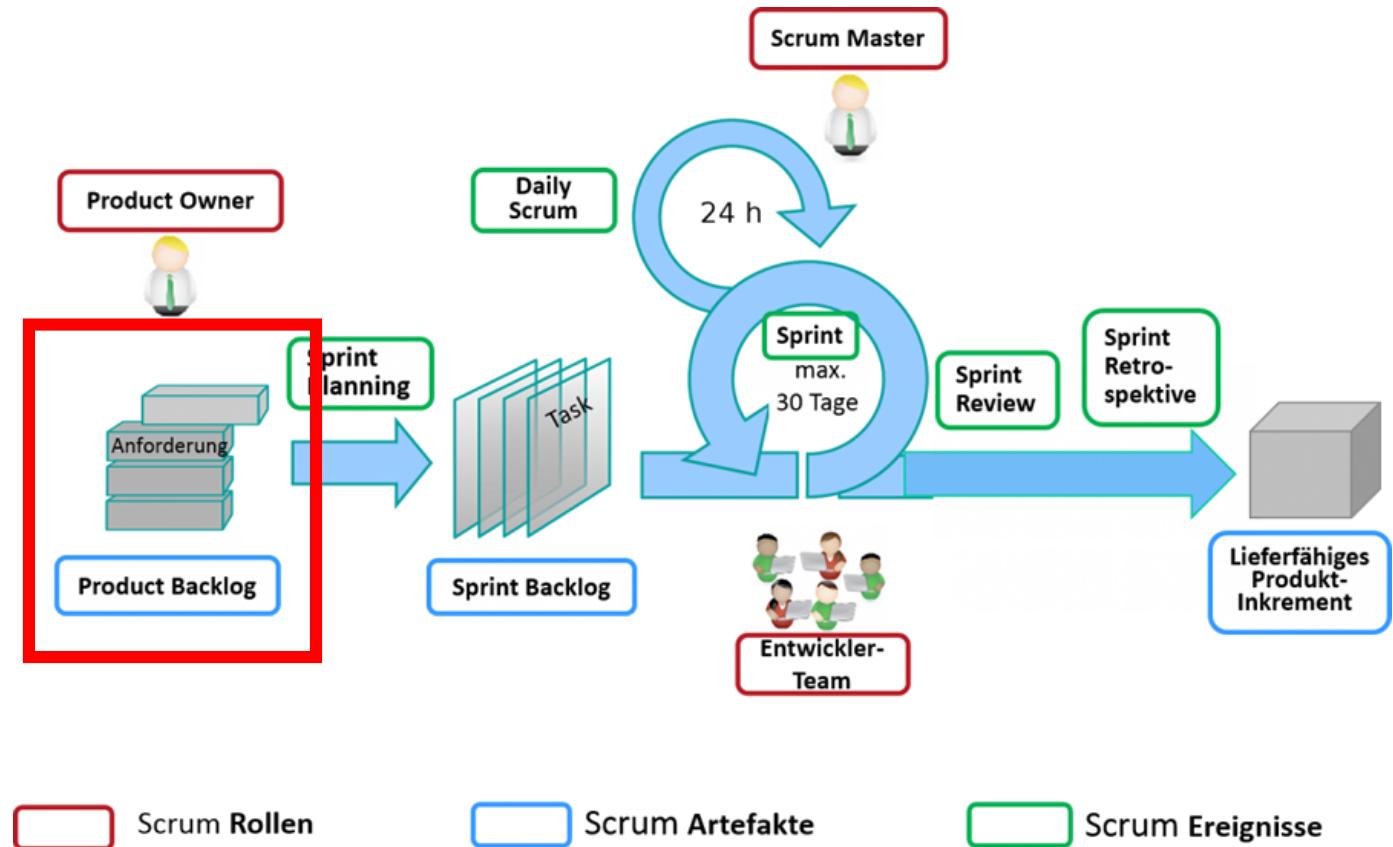
Scrum: modernes, agiles Projektmanagement-Framework
Nächste Woche mehr dazu, erstmal nur der Einstieg





- **Abgabe ist notwendig, um weiterzumachen!**
 - Es geht zunächst nicht um richtig oder falsch, sondern darum, sich damit zu **beschäftigen!**
 - Natürlich muss aber der Wille erkennbar sein...
- Finden Sie sich innerhalb ihrer Gruppe (maximal) in Dreier-Gruppen zusammen (diese Gruppen sind nur für die Übungsaufgaben)
 - Versuchen Sie, eine **Produktvision** zu entwickeln
 - Definieren Sie **User-Stories** für alle Anforderungen, die sich **nicht auf das eigentliche Spiel** beziehen (siehe dazu Folien und Confluence)
 - Tragen Sie sich dafür innerhalb ihrer Gruppen-Stud.IP-Veranstaltung in eine der Untergruppen ein
 - Warten Sie mit der Gruppenfindung nicht, bis zur ersten echten Gruppensitzung (die könnte erst kommende Woche nach der VL sein)
 - Geben Sie ihre Lösung **bis zum Beginn der kommenden Vorlesung** ab indem Sie sie in ihren **Gruppenordner** (Stud.IP Gruppenordner) als PDF hochladen
 - Wir werden dann gemeinsam, den ersten Sprint angehen.
 - Beschäftigen Sie sich mit dem **Spiel** und den **Einzelaufgaben**.

- Das Backlog
- Problem:
 - Wie baut man eine Software, die die Wünsche des Kunden entspricht?
- Ansatz:
 - Anforderungsermittlung durch Gespräche mit dem Kunden
 - Ein Ansatz dafür (und im SWP verwendet) sind User-Stories



- Nächste Woche geht es mit Scrum weiter ...
 - Denken Sie an die User-Stories!
 - Spielen Sie das Spiel
-
- Alle mit Problemen bei der Gruppenwahl bitte JETZT zu mir kommen.

Fragen?

