**[*Parkplatzerkennung*]**

**Projektmanagement-Plan**

[1. Projektorganisation 1](#_Toc478315547)

[1.1. Rollen & Zuständigkeiten 1](#_Toc478315548)

[1.2. Projektstrukturplan 2](#_Toc478315549)

[2. Projektführung 3](#_Toc478315550)

[2.1. Rahmen- und Projektplan 3](#_Toc478315551)

[2.2. Projektkontrolle und Projektsteuerung 4](#_Toc478315552)

[3. Projektunterstützung 7](#_Toc478315553)

[3.1. Tools für Entwicklung 7](#_Toc478315554)

[3.2. Konfigurationsmanagement 7](#_Toc478315555)

[3.3. Tools zur Projektkommunikation und zum Dokumentenaustausch 7](#_Toc478315556)

[4. Qualitäts- und Testplan 8](#_Toc478315557)

[4.1. Definition of Ready und Definition of Done 8](#_Toc478315558)

[4.2. Testdesign & Testautomatisierung 8](#_Toc478315559)

[4.3. Testfälle 8](#_Toc478315560)

[5. Anhänge 9](#_Toc478315561)

Versionen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen | Status |
| 0.1 | 14.03.2019 | Felix Willrich | 1. Entwurf + Eintragen aller Informationen | 0.1 |
|  |  |  |  |  |

# Projektorganisation

## Rollen & Zuständigkeiten

*<welche Rollen und Zuständigkeiten gibt es in Ihrem Projekt. Wer hat (zur Zeit) welche Rolle?*

*Welche Rolle hat welche wesentlichen Aufgaben/ Funktionen, welche Verantwortlichkeiten und welche Befugnisse?>*

*Beispiel:*

Rolle: Product-Owner: N.N.

Aufgaben: Federführend beim Erstellen und Managen des Product-Backlog, entwickelt die Produktvision…

Verantwortlichkeiten: Kommunikation der Produktvision ans Team und an den Auftraggeber,… Befugnisse: darf Ergebnisse eines Sprints annehmen oder ablehnen, Nachbesserungen oder Änderungen einfordern,…

*Mögliche Rollen können sein: Systemarchitekt, Entwickler, Qualitätsmanager, Tester, Projekt-Controller, Scrum-Master,….*

*in Tabellenform – in Stichworten >*

## Projektstrukturplan

**Ziel:**

**Erfassen von Aufgaben und ihren Abhängigkeiten**

**Vorbereitung für Aufwands- und Zeitplan**

Gliederung des Projekts in plan- und kontrollierbare Elemente. Strukturierung des Projekts nach einem Gesichtspunkt, z.B. welche Komponenten/Teilprodukte sind zu erstellen, oder nach Phasen/Projektaktivitäten oder Iterationen –

*<Zerlegen Sie Ihr Projekt bis zum ersten Release (14.11.) in Teilaufgaben und schließlich soweit, bis Sie Arbeitspakete ableiten können. >*

Ein Arbeitspaket ist eine geschlossene Aufgabenstellung, die zu einem festgelegten Zeitpunkt mit definiertem Ergebnis und Aufwand umgesetzt werden kann,

Einem Arbeitspaket sind zugeordnet

* ein (Haupt)Verantwortlichen
* inhaltliche Beschreibung
* die benötigten Ressourcen
* die Abhängigkeiten zu anderen Arbeitspaketen
* Zuordnung zu einem Meilenstein

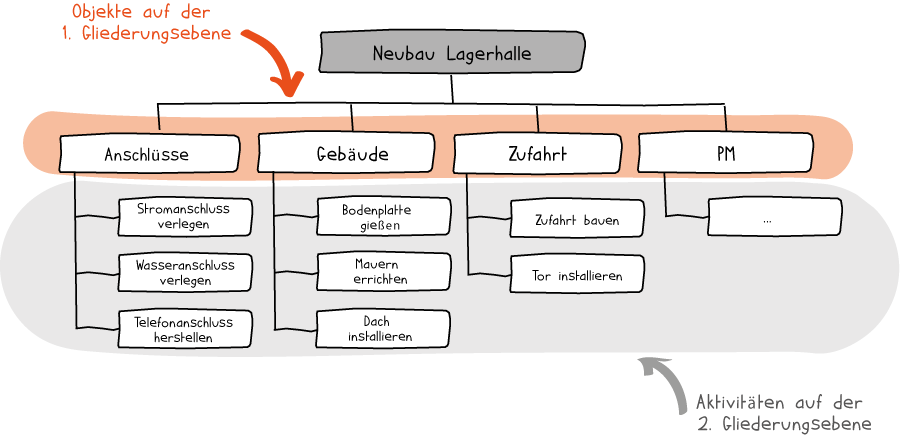


Abbildung 1: Beispiel Projektstrukturplan, © Projekte-leicht-gemacht

# Projektführung

## Rahmen- und Projektplan

Der grobe Ablaufplan ist durch die Terminvorgaben des Teamprojekts in wesentlichen Teilen festgelegt:

* Architektur: 24.10.
* 1. Release: 14.11.
* 2. Release: 28.11.
* 3. Release: 19.12.

So etwas ist bei größeren, agilen Projekten nicht unüblich und wird „Rahmenplan“ oder „Timebox der Projektführungsebene“ genannt.

*<Passend zu diesem Rahmenplan (s. auch Folie 6-7 in der Einführungsveranstaltung) sollen Sie nun einen Projektplan erstellen.>*

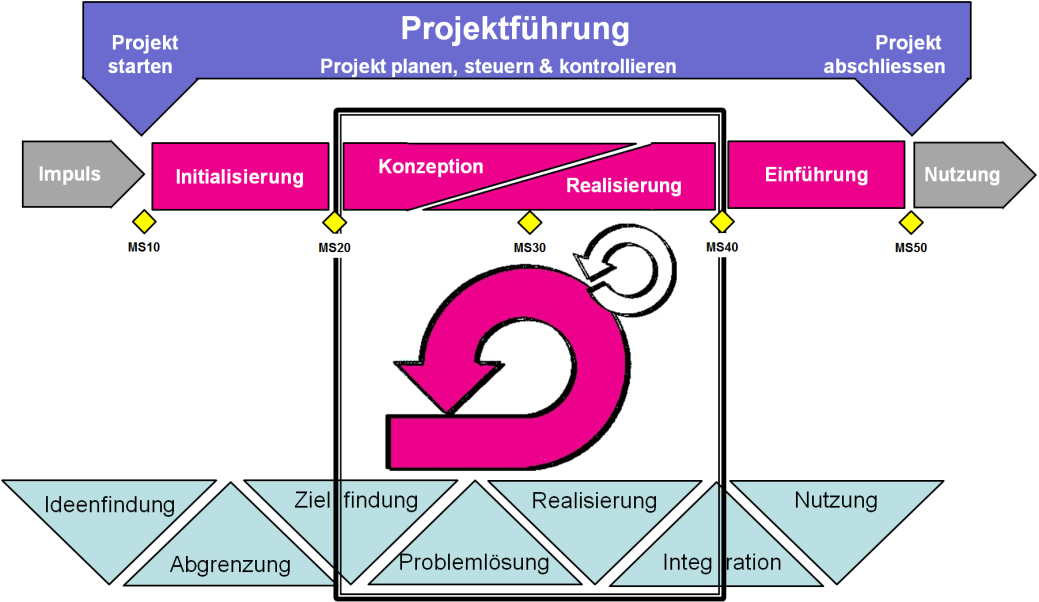


Abbildung 2: SoDa-Prozessmodell nach Hofstätter, Jud basierend auf Jenny03

Durch die Vorgabe des ersten Meilensteins „Architektur“ ist eine Initialisierungsphase vorgegeben, in der wesentliche vorbereitende Aktivitäten stattfinden sollen zu den Themen

* Architekturentwurf Ihrer Software
  + Komponenten und ihre Beziehungen
  + Externe und wichtige interne Schnittstellen
  + Kern des Datenmodells
* Infrastrukturplanung
  + Netzwerkdienste, die eingebunden werden sollen (Drucker, Mail,…)
  + Hardware (z.B. spezielle Controller,…)
  + Betriebssysteme, Middleware,
  + Software-Frameworks oder Protokolle, die genutzt werden soll (z.B. MySQL, LDAP…)
* Qualität (bezogen auf das Gesamtprojekt)
  + Anforderungen an IT-Sicherheit
  + UX
  + Performanz
  + …
* Technologie-Festlegung
  + IDEs
  + SW-Frameworks
  + Austausch und Kommunikation zu Entwicklungsartefakten (u.a. Konfigurationsmanagement, Issue-Board)
  + Testframework
  + …
* Vorbereiten der Projekt-Infrastruktur
  + Kommunikation im Projekt (Wiki, Email-Verteiler,…)
  + Austausch von Dokumenten
  + ..

Nach der Initialisierungsphase wird agil vorgegangen in Form von Scrum-Sprints, die dann jeweils eine Planungs-, eine Konzeptions-, ein Umsetzungs- und eine Auslieferungs/Einführungsphase beinhalten.

*<Ihre Aufgabe ist es, einen ersten Projektplan zu erstellen, der die Arbeitspakete der Initialisierungsphase und des ersten Sprints bis zum14.11.. beinhaltet. Für den weiteren Projektverlauf (2. Und 3. Sprint) genügt eine grobe Planung, anhand der Sie abschätzen können, ob und wie Sie Ziele des Projekts bis zum Projektende erreichen können. Für den Projektplan können Sie mit Netzplantechnik oder mit Balkendiagrammen (GANTT-Diagramm) arbeiten.>*

## Projektkontrolle und Projektsteuerung

Nach der ersten Planung muss kontinuierlich kontrolliert und nachgesteuert werden, damit die Projektziele erreicht werden. Nach Harry Sneed unterliegen Softwareprojekte dem sogenannten Teuflischen Viereck.

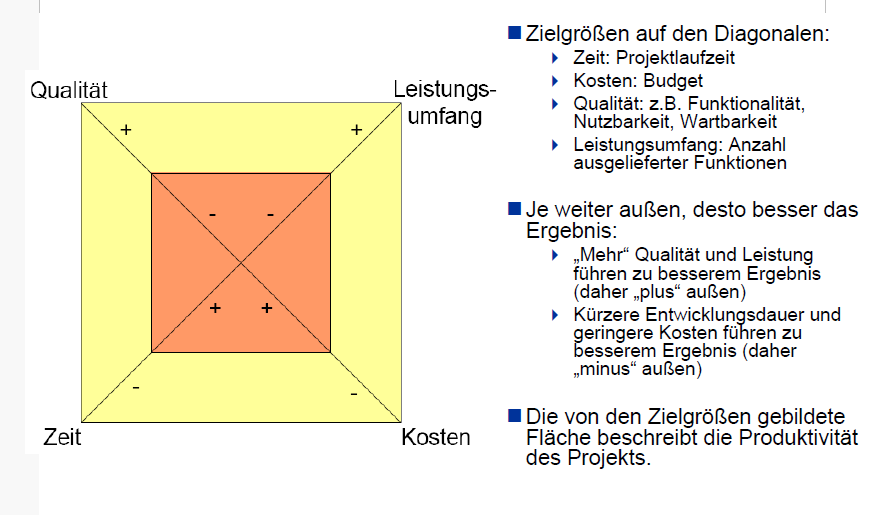


Abbildung 3: Teuflisches Viereck nach Harry Sneed ( aus Skript von B. Bauer)

In Ihrem Fall beschränken sich die Kosten auf die von Ihnen aufgebrachten Aufwände (Arbeitsstunden im Teamprojekt)

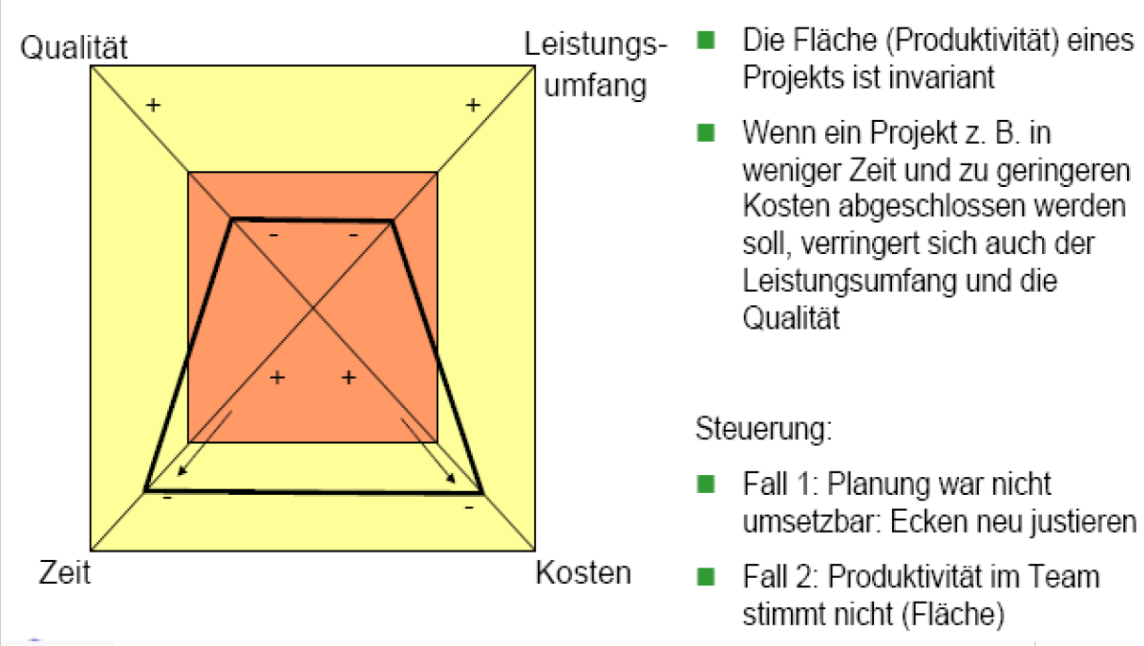


Abbildung 4: Teuflisches Viereck – Aufgaben für Kontrolle und Steuerung

Bei der Projektkontrolle und –steuerung werden die Projekt-Innensicht und Außensicht unterschieden. Die Instrumente der Projekt-Außensicht werden von den Auftraggebern und den Modulvorgaben für das Teamprojekt vorgegeben. Hier geht es daher um die Instrumente für die projekt-interne Kontrolle und Steuerung.

Aufgabe der Projektkontrolle ist die Fortschrittskontrolle in Bezug auf

* Leistungen (=Funktionen des Produkts/Teilprodukte)
* Zeit
* Kosten
* Qualität
* und Risikobewertung

Ziel ist eine (regelmäßige) Bestandsaufnahme als Basis für die weitere Planung, die Identifikation von Problemen und die Definition von Gegenmaßnahmen.

In der Projektsteuerung werden auf Basis der aktualisierten Pläne,

* Maßnahmen für die nächsten Schritte
* Reaktionen auf erkannte Probleme
* Gegenmaßnahmen definiert,

entschieden und angestoßen

*<Ihre Aufgabe ist es, für jede Dimension des Sneed’schen Teufelsvierecks zu benennen.*

* *Wer (welche Rolle)*
* *Wie (mit welchen Maßnahmen)*
* *Wann (relativ zu welchem Meilenstein, oder wann im Sprint)*
* *Mit welchem Ziel*

*das Projekt-Controlling durchgeführt werden soll.*

*Konzentrieren Sie sich auf die beiden in Abbildung 4 benannten Herausforderungen in der Projektsteuerung:*

* *Planung war nicht umsetzbar*
* *Produktivität stimmt nicht*

*>*

Beispiel:

Kosten-Controlling: Jedes Mitglied Ihres Teams **erfasst seine Zeitaufwände aufgabengenau mit xyz**. Die Auswertung wird wöchentlich zusammengetragen im Team gemeinschaftlich bewertet in Bezug auf die Produktivität in Form von umgesetzten Features/eingesetzte Zeit.

*Kommentar: Damit können Sie offensichtlich feststellen (=Controlling), ob einzelnen Teammitgliedern die „Kosten“ in Form ihrer eingesetzten Zeit nach oben oder unten aus dem Ruder laufen… . Die gemeinschaftliche Bewertung im Team erlaubt es darüber hinaus, auch die Produktivität einzuschätzen:*

* *Wird an den „richtigen“ Aufgaben gearbeitet?*
* *ist die Arbeit effektiv und effizient (oder wird viel Aufwand geleistet, aber am Ende sind zu wenig User Stories umgesetzt, getestet,…)?*

# Projektunterstützung

## Tools für Entwicklung

*<Listen Sie alle Festlegungen, die Sie bereits getroffen haben.*

*Zu jeder Werkzeug-Entscheidung sollte in Stichworten,*

* *Geltungsbereich*
* *Ausgewähltes Werkzeug / Plugin oder Framework*
* *Entscheidungsrahmen (Gründe, Randbedingungen,..)*

*angegeben werden>*

## Konfigurationsmanagement

*<Listen Sie alle Festlegungen, die Sie bereits getroffen haben.*

*Zu jeder Entscheidung sollte in Stichworten,*

* *Geltungsbereich*
* *Ausgewähltes Werkzeug / Plugin oder Framework*
* *Entscheidungsrahmen (Gründe, Randbedingungen,..)*

*angegeben werden >*

## Tools zur Projektkommunikation und zum Dokumentenaustausch

*<Listen Sie alle Festlegungen, die Sie bereits getroffen haben, siehe oben.*

# Qualitäts- und Testplan

## Definition of Ready und Definition of Done

Definition of Ready (DoR): Was muss ein Backlog Item mindestens erfüllen, damit es im nächsten Sprint Planning ausgewählt werden kann?

Beispiele:

* der Aufwand einer User Story ist geschätzt,
* die Story ist von allen verstanden,
* die Story umfasst maximal … Low-Level Anforderungen,
* es gibt klare Akzeptanzkriterien
* …

Definition of Done (DoD): Was muss ein Backlog Item erfüllen, damit Sie am Ende eines Sprints als vom Product Owner als fertig deklariert werden kann?

Beispiele:

* mindestens n Entwickler haben daran gearbeitet,
* alle automatisierten Test Unit- und Funktionstests laufen,
* die Dokumentation wurde erstellt/erweitert,
* der PRoduct Owner hat die Story abgenommen
* …

*<Legen Sie in Ihrem Scrum Team fest, mit welchen Definitionen Sie arbeiten wollen. je eine Liste für die DoD und die DoR>*

## Testdesign & Testautomatisierung

Neben funktionalen Tests ergeben sich häufig projektspezifische Aspekte, die speziellere Tests erforderlich machen. (Mobile) Benutzbarkeitstests, Sensor- und Aktortest, Robustheit gegen Sensorfehler oder Kommunikationsverlust,… gehören in diese Kategorie.

*<Welche Testnotwendigkeiten gibt?*

*Sollen Ansätze und Werkzeuge zur Unterstützung dieser Tests und zur Testautomatisierung genutzt werden?*

*Wenn ja, welche?>*

## Testfälle

<noch nicht notwendig> (nur Testfälle für nicht automatisierte Tests, die nicht als «definition of done» formuliert sind)

# Anhänge

<später > (Sprintpläne / Sprintreview-Protokolle / Meilensteinberichte)