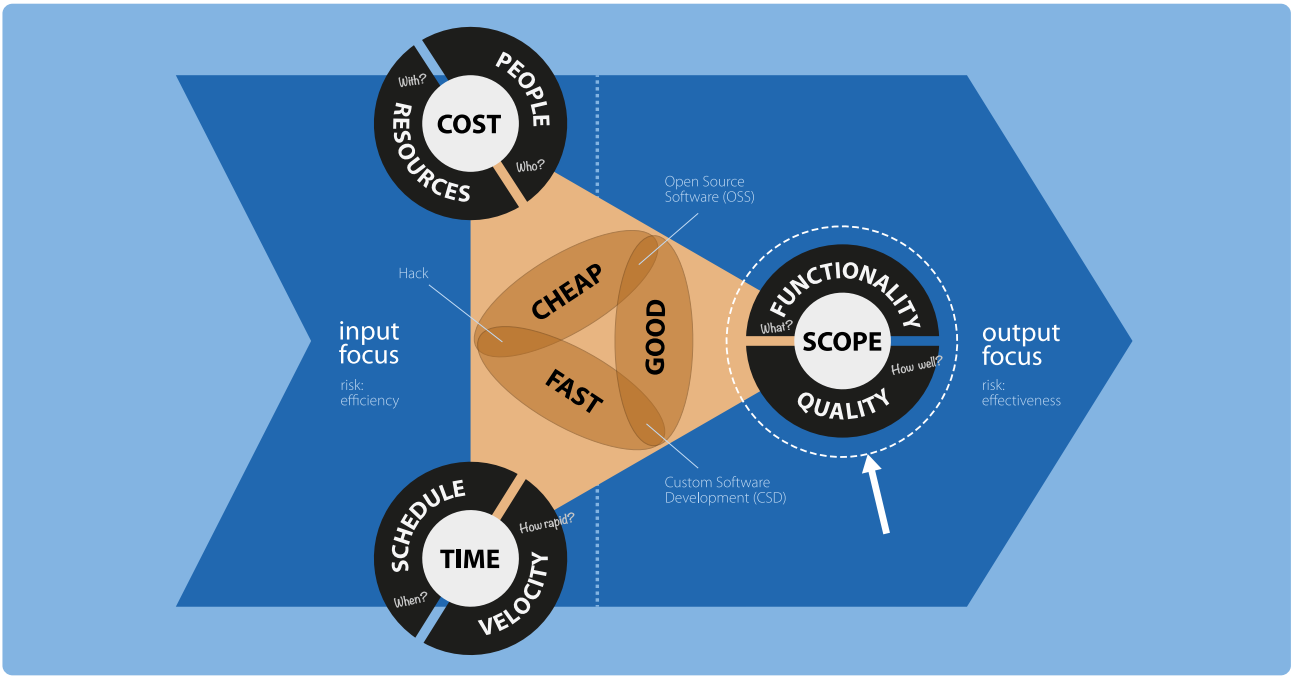


Software Engineering in Industrial Practice (SEIP)

Dr. Ralf S. Engelschall

Project Management Triangle



Definition of a *Project*:
"Temporary endeavor undertaken to create a unique product, service or result."
Temporary in that it has a defined beginning and end in time, and a defined scope and cost.
Unique in that it is not a routine operation, but a one-time, single-goal, and risk-containing operation.

Project Management *Iron Triangle*:
A project is constrained by **time**, **cost** and **scope**. No constraint in this triangle can be changed without affecting the others.
Time splits into **schedule** and **velocity**.
Cost splits into **people** and **resources**. Scope splits into **functionality** and result **quality**.

Project Management *Trilemma*:
"Fast. Cheap. Good. Pick two!"
Each project optimization effort has the choice among **three** favourable options — only **two** of them are possible at the same time.

Project Management ist neben **Software Architecture** die zweite wichtige Disziplin im Bereich **Software Engineering**. Deshalb sollte jeder zumindest ein Grundverständnis über die wesentliche Aufgabe des Project Management haben: die Balance aus dem "Iron Triangle" aus **Time** (Termine), **Cost** (Kosten) und **Scope** (in diesem Kontext üblicherweise Leistung genannt) kontinuierlich zu finden und zu halten.

Die "Stellschraube" **Time** unterteilt sich in die beiden Aspekte **Schedule** (When?) und **Velocity** (How rapid?). Die Stellschraube **Cost** unterteilt sich in die beiden Aspekte **People** (Who?) und **Resources** (With?). Die Stellschraube **Scope** unterteilt sich in die beiden Aspekte **Functionality** (What?) und **Quality** (How well?).

Wenn an einer der drei Stellschrauben bzw. an einem der sechs Aspekte eine Änderung stattfindet, ist das "Iron Triangle" außer Balance und man muss unweigerlich eine oder mehrere der anderen Stellschrauben bzw. Aspekte verändern, um die Balance wiederherzustellen.

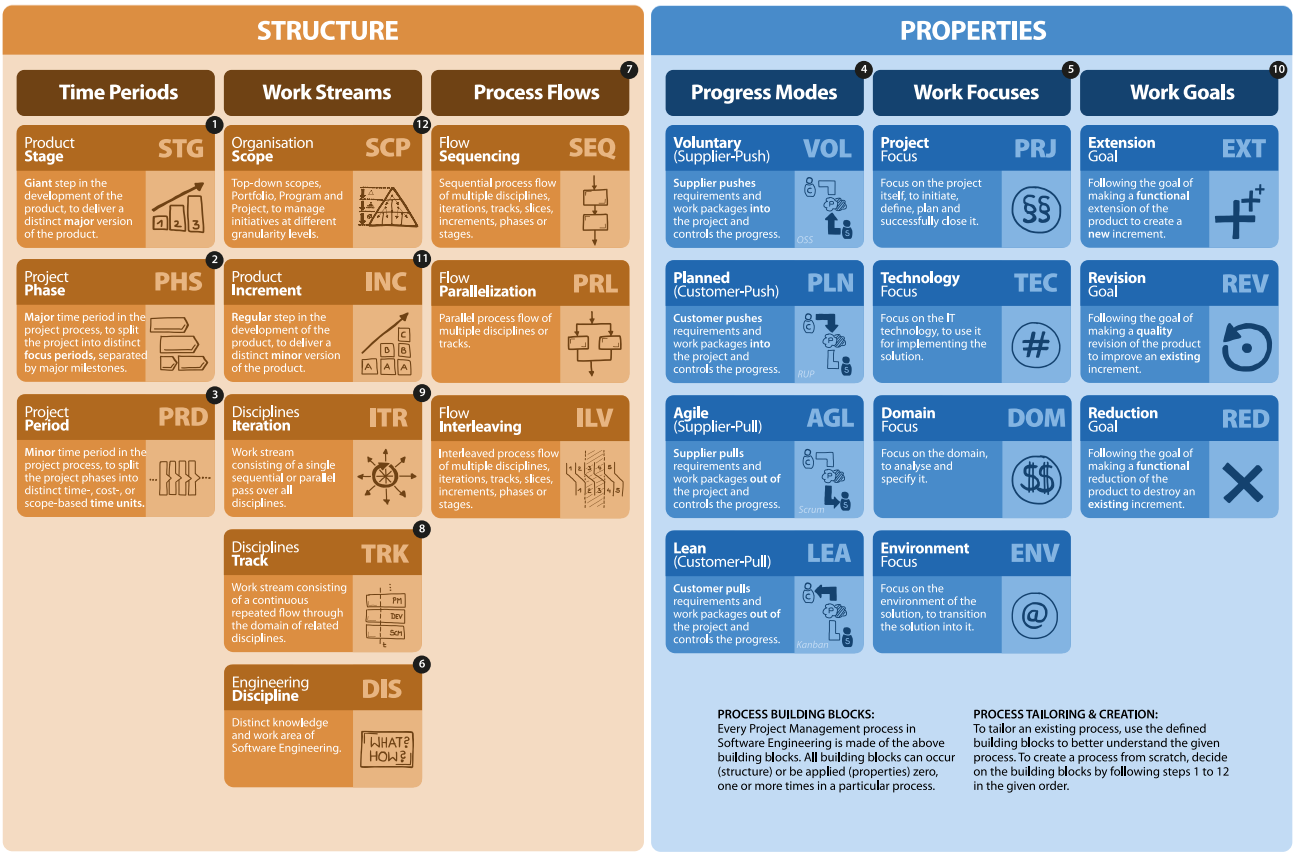
Ebenfalls erwähnenswert ist das **Trilemma**, das besagt, daß man üblicherweise immer nur zwei von drei Dingen gleichzeitig haben kann: entweder billig und gut (Open Source Software), dann aber nicht schnell; oder gut und schnell (Custom Software Development), dann aber nicht billig; oder schnell und billig (der "Quick Hack"), dann aber nicht gut.

Die Nicht-Projekt-Manager sind in der Praxis vor allem bei der Stellschraube **Scope** in der Mitverantwortung, da hier eine Veränderung im Projekt üblicherweise ein tiefgreifenderes technisches Verständnis der Anwendung bedingt.

Fragen

- An welcher "Stellschraube" des **Project Management** sind in der Praxis auch die Nicht-Projekt-Manager stark mitverantwortlich?

Project Management Building Blocks



Jeder Projektmanagement-Prozess (auch Vorgehensmodell) im Software Engineering wird aus dem gleichen Satz von Bausteinen erzeugt. Alle Bausteine können vorkommen (Struktur) oder angewendet werden (Eigenschaften), und zwar null, ein oder mehrere Male in einem bestimmten Prozess.

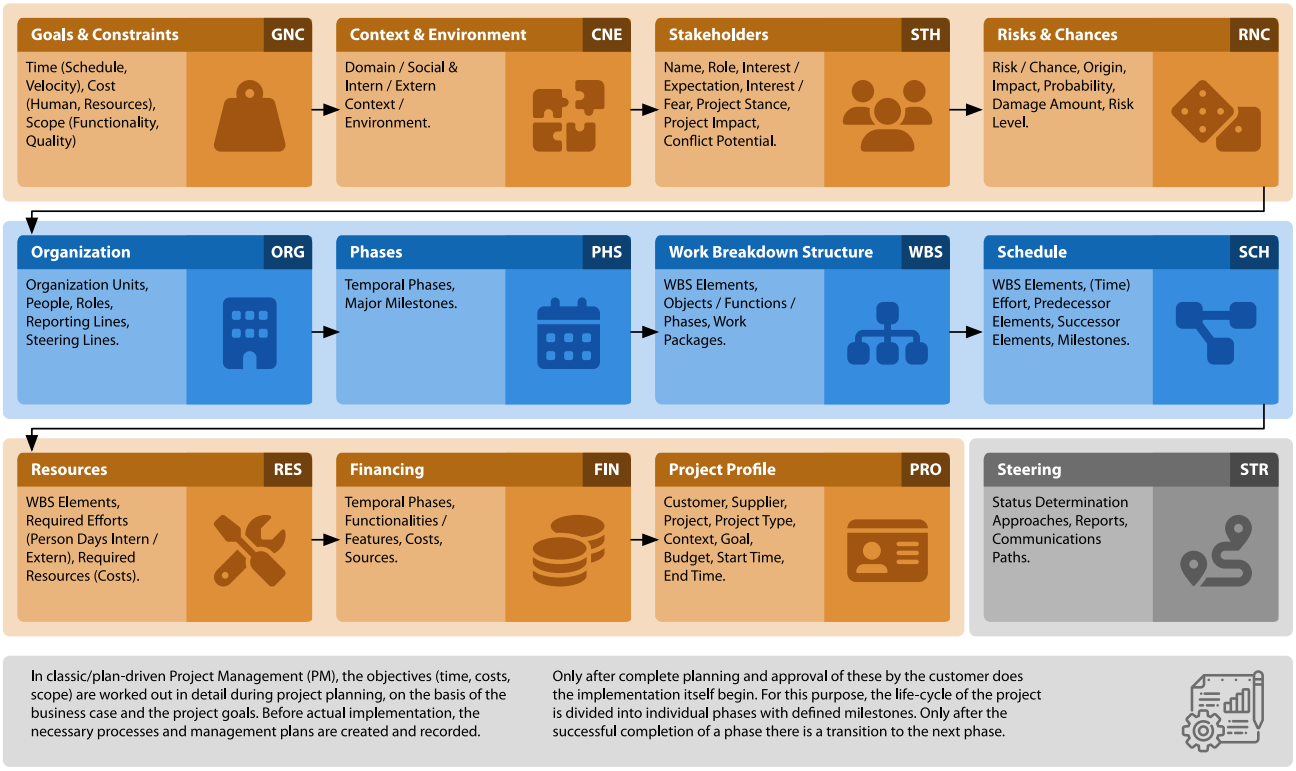
Um einen bestehenden Prozess anzupassen, verwenden Sie die definierten Bausteine, um den gegebenen Prozess zu verstehen. Um einen Prozess von Grund auf neu zu erstellen, entscheiden Sie sich für die Bausteine, indem Sie die Schritte 1 bis 12 in der angegebenen Reihenfolge ausführen.

Fragen

? Ist ein spezieller Projektmanagement-Prozess im Software Engineering entscheidend?

Intellectual Content: Version 1.0.8 (2022-08-11), Authored 2016-2022 by Dr. Ralf S. Engelschall
Graphical Illustration: Version 1.0.9 (2022-09-11), Copyright © 2016-2021 Dr. Ralf S. Engelschall <http://engelschall.com>, All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contents only.

Plan-Driven Project Management



In classic/plan-driven Project Management (PM), the objectives (time, costs, scope) are worked out in detail during project planning, on the basis of the business case and the project goals. Before actual implementation, the necessary processes and management plans are created and recorded.

Only after complete planning and approval of these by the customer does the implementation itself begin. For this purpose, the life-cycle of the project is divided into individual phases with defined milestones. Only after the successful completion of a phase there is a transition to the next phase.

Intellectual content: Version 1.0.0 (2021-12-20), Authored 2022 by Dr. Ralf S. Engelschall based on concepts from Steve McConnell
Graphical illustration: Version 1.0.0 (2022-01-10), Copyright © 2022 Dr. Ralf S. Engelschall <http://engelschall.com> – All rights reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

Im klassischen, plan-getriebenen Projektmanagement (PM) werden die Ziele (Termine, Kosten, Leistung) bei der Projektplanung detailliert ausgearbeitet, auf der Grundlage des Business Case und der Projektziele. Vor der eigentlichen Umsetzung werden die notwendigen Prozesse und Managementpläne erstellt und festgehalten.

Erst nach vollständiger Planung und Freigabe dieser durch den Kunden beginnt die eigentliche Umsetzung. Zu diesem Zweck wird der Lebenszyklus des Projekts in einzelne Phasen mit definierten Meilensteinen unterteilt. Erst nach dem erfolgreichen Abschluss einer Phase erfolgt der Übergang in die nächste Phase.

Fragen

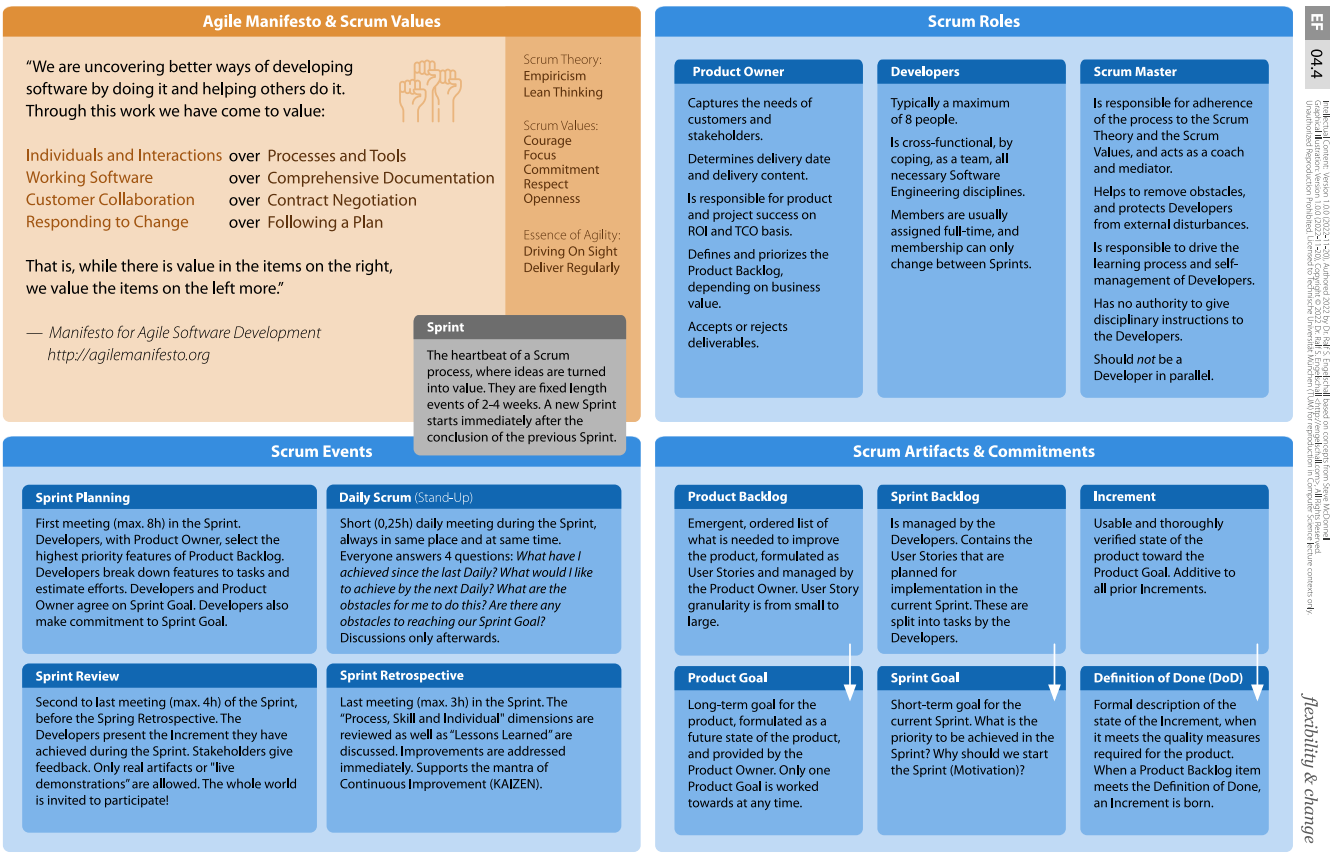
- Was ist das größte Problem von plan-getriebenem Projektmanagement?

ENGINEERING FUNDAMENTALS

Agile Project Management

TECHNISCHE UNIVERSITÄT MÜNCHEN

EF 04.4



Intellectual content: Version 1.0.0 (2021-12-20). Authored 2022 by Dr. Ralf S. Engelschall based on concepts from Steve McConnell. Graphical illustration: Version 1.0.0 (2022-04-10). Copyright © 2022 Dr. Ralf S. Engelschall <http://engelschall.com>. All rights reserved. Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contents only.

Agilität ist eine Denkweise für die Entwicklung von Software, bei der man Wert legt auf (gemäß dem "Manifest für agile Softwareentwicklung"): Individuen und Interaktionen über Prozessen und Werkzeugen, funktionierende Software über umfassender Dokumentation, Zusammenarbeit mit dem Kunden über Vertragsverhandlungen und Reagieren auf Veränderungen über dem Befolgen eines Plans.

Scrum ist ein einfaches, leichtgewichtiges, methodisches Rahmenwerk, welches (laut dem "Scrum Guide") "Menschen, Teams und Organisationen dabei hilft, Werte durch adaptive Lösungen für komplexe Probleme zu schaffen", und welches die Denkweise der Agilität verfolgt und unterstützt.

Scrum setzt eine Umgebung voraus, in der: ein Product Owner die Arbeit für ein komplexes Problem in ein Product Backlog einordnet; ein Entwicklerteam während eines Sprints eine Auswahl der Arbeit in einen nutzbaren Wertzuwachs umsetzt; der Product Owner, das Entwicklerteam und seine Stakeholder die Ergebnisse inspizieren und das weitere Vorgehen für den nächsten Sprint anpassen; schließlich wird dieser Prozess einfach wiederholt.

Fragen

- Welches Hauptproblem adressiert Agile Software-Entwicklung?