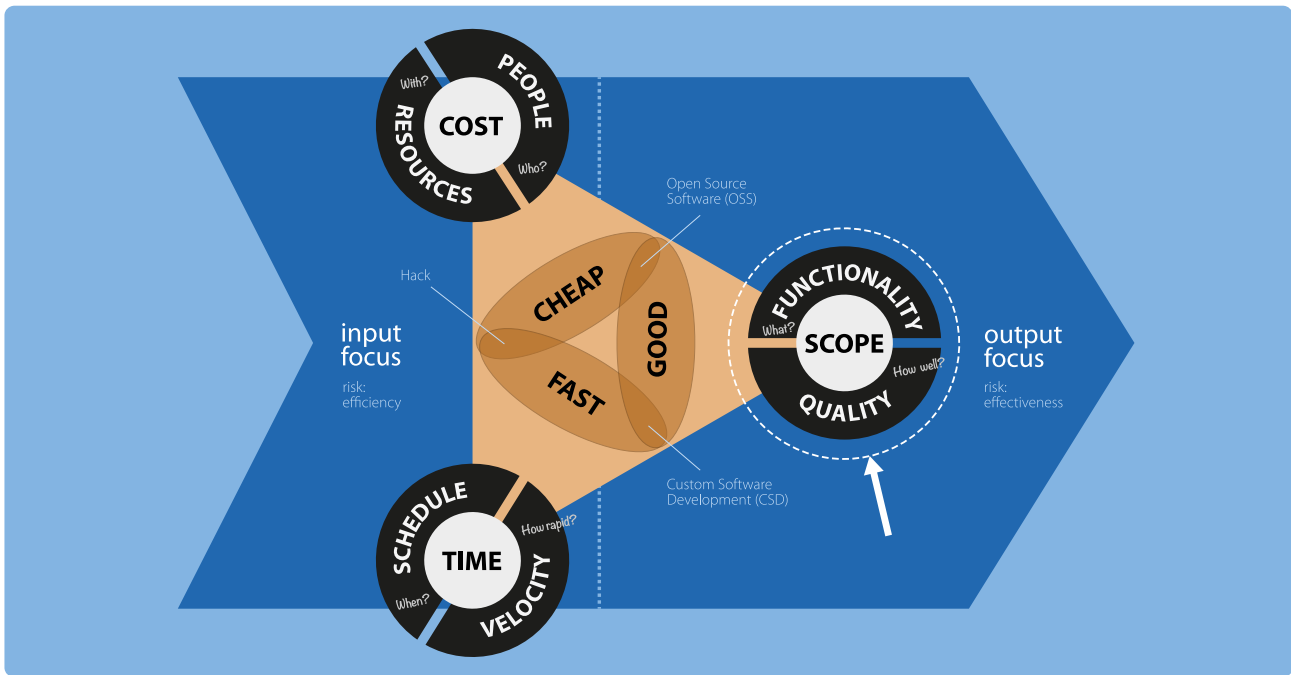




# **Software Engineering in der industriellen Praxis (SEIP)**

**Dr. Ralf S. Engelschall**



#### 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 & constraints

**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?).

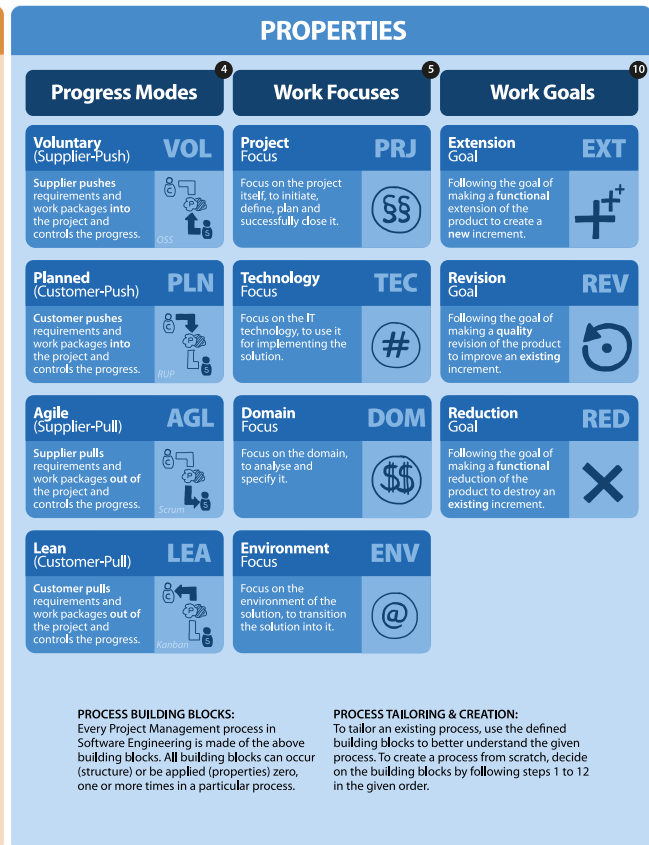
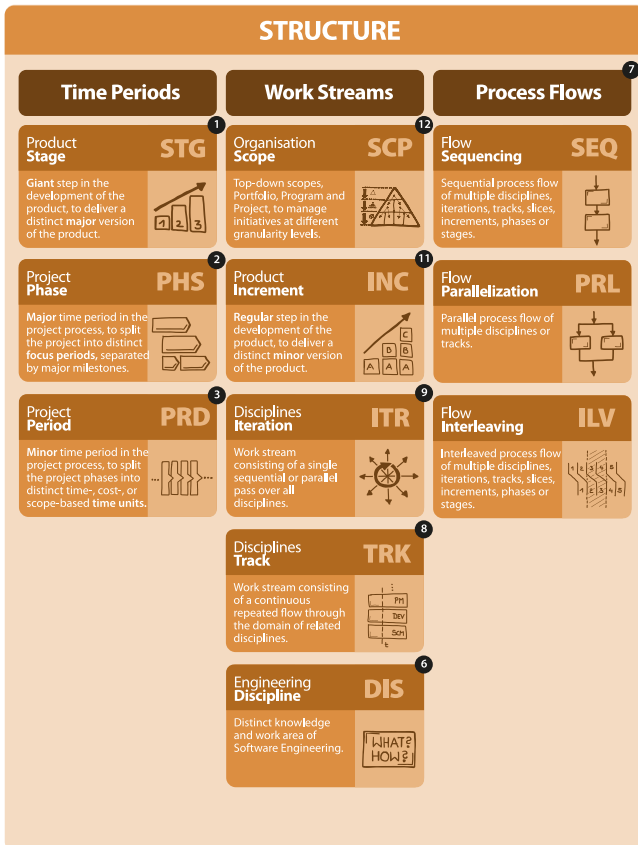
Wenn an einer der drei Stellschrauben bzw. an einer der sechs Aspekte eine Änderung stattfindet, ist das "Iron Triangle" außer Balance und man muss unweigerlich eine oder mehrere der anderen Stellschrauben bzw. Aspekte verändern, um die Balance wieder herzustellen.

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?



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?



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.



thinking & planning

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?

## Agile Manifesto & Scrum Values

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:



Individuals and Interactions over Processes and Tools  
Working Software over Comprehensive Documentation  
Customer Collaboration over Contract Negotiation  
Responding to Change over Following a Plan

Scrum Theory:  
Empiricism  
Lean Thinking

Scrum Values:  
Courage  
Focus  
Commitment  
Respect  
Openness

Essence of Agility:  
Driving On Sight  
Deliver Regularly

That is, while there is value in the items on the right, we value the items on the left more."

— Manifesto for Agile Software Development  
<http://agilemanifesto.org>

### Sprint

The heartbeat of a Scrum process, where ideas are turned into value. They are fixed length events of 2-4 weeks. A new Sprint starts immediately after the conclusion of the previous Sprint.

## Scrum Roles

### Product Owner

Captures the needs of customers and stakeholders.  
Determines delivery date and delivery content.  
Is responsible for product and project success on ROI and TCO basis.  
Defines and prioritizes the Product Backlog, depending on business value.  
Accepts or rejects deliverables.

### Developers

Typically a maximum of 8 people.  
Is cross-functional, by coping, as a team, all necessary Software Engineering disciplines.  
Members are usually assigned full-time, and membership can only change between Sprints.

### Scrum Master

Is responsible for adherence of the process to the Scrum Theory and the Scrum Values, and acts as a coach and mediator.  
Helps to remove obstacles, and protects Developers from external disturbances.  
Is responsible to drive the learning process and self-management of Developers.  
Has no authority to give disciplinary instructions to the Developers.  
Should not be a Developer in parallel.

## Scrum Events

### Sprint Planning

First meeting (max. 8h) in the Sprint. Developers, with Product Owner, select the highest priority features of Product Backlog. Developers break down features to tasks and estimate efforts. Developers and Product Owner agree on Sprint Goal. Developers also make commitment to Sprint Goal.

### Daily Scrum (Stand-Up)

Short (0,25h) daily meeting during the Sprint, always in same place and at same time. Everyone answers 4 questions: *What have I achieved since the last Daily? What would I like to achieve by the next Daily? What are the obstacles for me to do this? Are there any obstacles to reaching our Sprint Goal?* Discussions only afterwards.

### Sprint Review

Second to last meeting (max. 4h) of the Sprint, before the Sprint Retrospective. The Developers present the Increment they have achieved during the Sprint. Stakeholders give feedback. Only real artifacts or "live demonstrations" are allowed. The whole world is invited to participate!

### Sprint Retrospective

Last meeting (max. 3h) in the Sprint. The "Process, Skill and Individual" dimensions are reviewed as well as "Lessons Learned" are discussed. Improvements are addressed immediately. Supports the mantra of Continuous Improvement (KAIZEN).

## Scrum Artifacts & Commitments

### Product Backlog

Emergent, ordered list of what is needed to improve the product, formulated as User Stories and managed by the Product Owner. User Story granularity is from small to large.

### Sprint Backlog

Is managed by the Developers. Contains the User Stories that are planned for implementation in the current Sprint. These are split into tasks by the Developers.

### Increment

Usable and thoroughly verified state of the product toward the Product Goal. Additive to all prior Increments.

### Product Goal

Long-term goal for the product, formulated as a future state of the product, and provided by the Product Owner. Only one Product Goal is worked towards at any time.

### Sprint Goal

Short-term goal for the current Sprint. What is the priority to be achieved in the Sprint? Why should we start the Sprint (Motivation)?

### Definition of Done (DoD)

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

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 einfacher, leichtgewichtiger, 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 ein 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?