

1. VORGEHENSMODELLE FÜR PROJEKTE

Wie Sie in Kapitel 1.2, S. 15 erfahren haben, wird im Rahmen eines Projekts etwas Neues geschaffen und beispielsweise aus einer Idee ein neues Produkt entwickelt. Ein solches Vorhaben setzt ein strukturiertes Vorgehen voraus. Allgemein bedeutet dies, den Ablauf von der Idee zum neuen Produkt möglichst effizient und zielgerichtet zu gestalten. Dieser Ablauf wird üblicherweise als **Realisierungs- oder Entwicklungsprozess** bezeichnet. Während der erste Begriff für alle Projektarten verwendet werden kann, kommt der zweite Begriff meist bei Entwicklungsprojekten zum Einsatz.

Abbildung 1: Realisierungsprozess als zentraler Prozess eines Projekts



Wenn beim Realisierungsprozess nicht strukturiert vorgegangen wird, müssen folgende Risiken in Kauf genommen werden:

- Ein Produkt wird realisiert, bevor das vom Auftraggeber erwartete Ergebnis klar ist.
- Wichtige Umsetzungsaufgaben bzw. -tätigkeiten werden vergessen.
- Dokumente, die für die Nutzung oder Weiterentwicklung des Produkts wichtig sind, werden gar nicht oder zu spät erstellt.

Solche Versäumnisse können nicht nur zu zeitraubenden Nacharbeiten in einem Projekt führen, sondern sogar dazu, dass ein Projekt erfolglos abgebrochen werden muss.

Beispiele aus der Softwareentwicklung

- Eine Anwendung wird allzu rasch programmiert, ohne dass vorher abgeklärt wird, was die Benutzer genau wollen. Folge: Eine Erhebung und Umsetzung der notwendigen Änderungen ist ähnlich aufwendig wie der bisherige Realisierungsprozess.
- Bei der Inbetriebnahme einer Anwendung wird keine Ausbildung der Benutzer vorgesehen. Folge: Das Ergebnis des Projekts kann nicht effektiv genutzt werden.
- Während der Entwicklung einer Anwendung werden keine Dokumente zur Wartung und I oder Weiterentwicklung erstellt. Folge: Die fehlenden Informationen müssen nachträglich gesammelt und festgehalten werden.

Vorgehensmodelle für Projekte können als «Best Practices»] aufgefasst werden, die Erfahrungen aus der Abwicklung zahlreicher Projekte zusammenfassen und für eigene Projekte (ggf. mit geringfügigen Anpassungen) übernommen werden können. Auf diese Weise kann Zeit gespart und verhindert werden, dass die gleichen oder gleichartige Fehler wiederholt werden. In den folgenden Unterkapiteln werden verschiedene Elemente und Typen von Vorgehensmodellen näher vorgestellt. Darüber hinaus lernen Sie Modelle für ein strukturiertes Vorgehen kennen, die bei IT-Projekten häufig zum Einsatz kommen

a) Elemente von Vorgehensmodellen

Mithilfe eines Vorgehensmodells kann der Realisierungsprozess in einzelne Schritte bzw. Aufgaben zerlegt werden, die so anzuordnen sind, dass die Idee möglichst effektiv und effizient in ein Ergebnis bzw. Produkt umgesetzt wird. Zu diesem Zweck wird der gesamte Prozess in mehrere Phasen aufgeteilt, wobei in jeder Phase bestimmte Aktivitäten ausgeführt und definierte Ergebnisse erarbeitet werden müssen. Jede Phase endet mit einem so genannten Meilenstein.

Element	Beschreibung	Beispiel
Phase	Zeitabschnitt, in dem ein bestimmter Teilprozess durchgeführt und bestimmte Ergebnisse erarbeitet werden.	Anforderungsanalyse
Meilenstein	Abschluss einer Phase bzw. Zeitpunkt, zu dem bestimmte Ergebnisse erwartet werden. Mit einem Meilenstein ist oft eine Entscheidung über das weitere Vorgehen verbunden.	Spezifikation abgenommen
Aktivität	Konkrete Tätigkeit während einer Phase. Aktivitäten beziehen sich immer auf konkrete Ergebnisse.	System abgrenzen
Ergebnis	Erwartetes Lieferobjekt einer Aktivität oder Phase. Typische Lieferobjekte bei IT-Projekten sind Dokumente oder Programmcode.	Pflichtenheft
Rolle	Funktionale Beschreibung der Personen, die an einer Aktivität bzw. an den jeweiligen Ergebnissen beteiligt sind. Anhand der Rollenbeschreibung kann ermittelt werden, welche Kompetenzen im Projektteam benötigt werden.	Business Analyst

Ein **Vorgehensmodell** beschreibt die Anordnung und Beziehungen dieser Elemente, d. h. den Ablauf der einzelnen Phasen und Aktivitäten, die zu liefernden Ergebnisse der einzelnen Phasen bzw. Aktivitäten sowie die dafür benötigten Rollen und Entscheidungen.

b) Typen von Vorgehensmodellen

In den letzten Jahrzehnten haben sich unterschiedliche Typen von Vorgehensmodellen herausgebildet, die ausführlich beschrieben und praktisch erprobt wurden. Entsprechend gross ist der Fundus an verfügbaren Vorgehensmodellen. Für Projekte in den Bereichen **System- und Softwareentwicklung** lassen sich diese Vorgehensmodelle heranziehen:

- **Sequenzielle Vorgehensmodelle** zeichnen sich dadurch aus, dass eine Phase weitgehend abgeschlossen wird, bevor die nächste Phase beginnt. Solche Modelle sind weit verbreitet, da sie leicht verständlich und relativ einfach anzuwenden sind.
- **Iterative Vorgehensmodelle** zeichnen sich dadurch aus, dass das Endprodukt schrittweise entwickelt wird und die einzelnen Phasen mehrfach, in unterschiedliche Intensität durchlaufen werden. Solche Modelle eignen sich besonders gut für grosse Entwicklungsprojekte
- **Agile Vorgehensmodelle** zeichnen sich durch ein iteratives Vorgehen sowie durch flexible Formen der Projekt- und Arbeitsorganisation aus. Solche Modelle sind in letzter Zeit bei Softwareprojekten sehr populär geworden, setzen aber von allen Beteiligten eine hohe Flexibilität voraus.

Jedes Vorgehensmodell hat seine spezifischen **Vor- und Nachteile**. Je nach Projekt bzw. Unternehmen wird ein bestimmtes Modell nicht in Reinkultur eingesetzt, sondern ggf. mit Elementen aus anderen Vorgehensmodellen angereichert. Oft definieren Unternehmen ein «eigenes» Vorgehensmodell für Projekte und erklären es zum Standard. Der Vorteil liegt darin, dass die Ausbildung der Projektleiter auf dieses Modell konzentriert werden kann. Auch ist ein Wechsel in der Projektführung einfacher zu bewerkstelligen. Als Projektleiter sollten Sie aber grundlegende Kenntnisse über alle Typen von Vorgehensmodellen haben.

a) Verbreitete Vorgehensmodellen

Nachfolgend werden weit verbreitete Vertreter der verschiedenen Typen näher vorgestellt, wobei lediglich auf grundlegende Aspekte eingegangen wird. Die korrekte Anwendung der einzelnen Vorgehensmodelle setzt eine vertiefte Auseinandersetzung mit ihnen voraus.

IPERKA

Die 6-Schritte-Methode IPERKA eignet sich sehr gut um gerade für ungeübte Projektmanagement-Novizen, um die die Thematik „hineinzuwachsen“. Dabei sind die sechs Schritte wie folgt definiert:

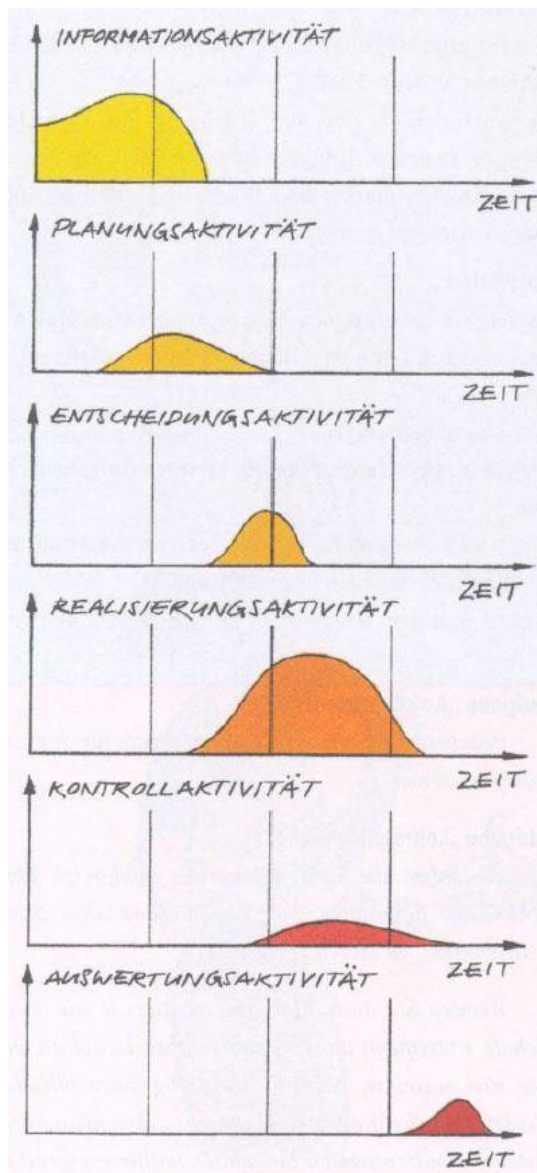


1. **Informieren:** Beim ersten Schritt geht es darum, die gestellte Aufgabe zu verstehen, zu erfassen worum es geht und sich ein Bild vom Ziel zu machen. Besondere Bedeutung kommt der systematischen Informationsbeschaffung zu.
2. **Planen:** Bei der Arbeitsplanung geht es darum, sich den Weg zum Ziel vorzustellen. Jeder einzelne Arbeitsschritt sollte dazu notiert werden und dann in einen sinnvollen Ablauf gebracht werden. Im schriftlichen Arbeitsplan werden zudem die Hilfsmittel festgehalten, der Zeitbedarf geschätzt und bei Arbeiten im Team die Arbeitsverteilung festgelegt. Zielformulierungen werden erstellt: Ich will.../ wir wollen...
3. **Entscheiden:** Sind Information und Planung optimal erfolgt, fällt das Entscheiden in der Regel nicht mehr schwer. Lösungsvarianten können mit Hilfe einer Entscheidungstabelle bewertet werden. Schwierige Entscheide sollten nicht hinausgezögert werden, sonst kostet das sehr viel Zeit oder schlimmsten-falls gibt es gar nichts mehr zu entscheiden, weil dann Sachzwänge vorliegen.
4. **Realisieren:** Das Realisieren oder ausführen kann, wie bei Fertigungsaufgaben üblich, den zeitlichen Hauptteil einer Aufgabe umfassen oder, z.B. beim Durchführen einer Veranstaltung nur einen kleineren Teil beanspruchen. Der in der Planungsphase erstellte Arbeitsplan ist in der Regel einzuhalten und sollte nicht leichtfertig geändert werden.
5. **Kontrollieren und prüfen:** Jede ausgeführte Arbeit ist zu kontrollieren, bevor sie aus den Händen gegeben wird. Kontrollieren heisst z.B. nochmals durchlesen, mit der Aufgabenstellung vergleichen usw. Optimal ist es, wenn die Kontrolle nicht durch dieselbe Person erfolgt wie die Ausführung.
6. **Auswerten:** Beim Auswerten denken wir über die ausgeführte Arbeit nach. Eventuell zusammen mit dem Auftraggeber. Zu diesem Zweck lässt man den ganzen Ablauf

nochmals Revue passieren, vergegenwärtigt sich, was gut, was weniger gut gelaufen ist und was beim nächsten Mal verbessert werden sollte.

Die Bezeichnung 6-Schritte-Methode erzeugt den Eindruck, dass es sich dabei um genau abgegrenzte Bearbeitungsschritte handelt, welche nacheinander in Angriff genommen werden. Dies kann bei gewissen Aufgaben durchaus so sein. Häufig ist jedoch, dass sich **die einzelnen Schritte nicht scharf abgrenzen lassen** und genau genommen eher Bearbeitungsphasen darstellen, welche sich zeitlich überlappen können.

Wie in der Grafik zu erkennen ist, beginnt z.B. das Planen schon, wenn die Informationsphase noch nicht ganz abgeschlossen ist. Oder es wird notwendig sein, während der Planungsphase Entscheidungen zu treffen.



Hermes

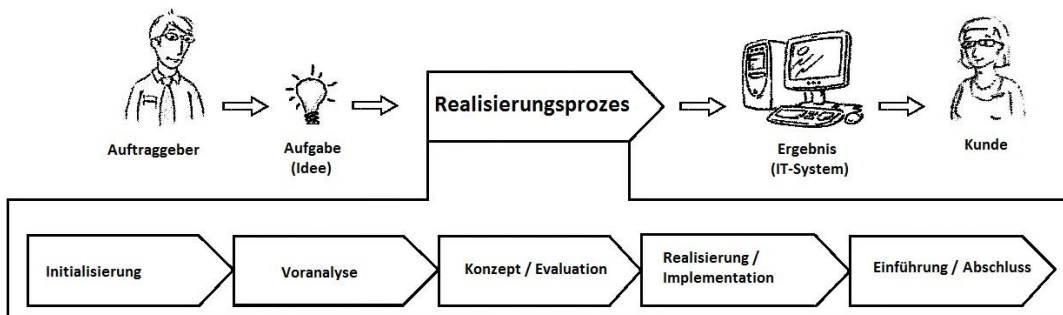
Bei Hermes handelt es sich um eine schweizerische Projektführungsmethode, die vom **Informatiksteuerungsorgan des Bundes (ISB)** publiziert und gepflegt wird. Sämtliche Unterlagen über Hermes können kostenlos vom Internet heruntergeladen werden. Dazu gehören folgende Dokumente:

- **Hermes Manager:** Pocket Guide mit den wichtigsten Begriffen (Format A6)
- **Hermes Grundwissen:** Kurzfassung zu den Grundlagen (Format A5)

Hermes gilt als Standard bei Informatikprojekten des Bundes, es kann aber grundsätzlich von jedermann verwendet werden. Inzwischen ist eine auf Kleinprojekte zugeschnittene Variante von Hermes erhältlich. Diese gliedert den Projektablauf in folgende **Phasen**:

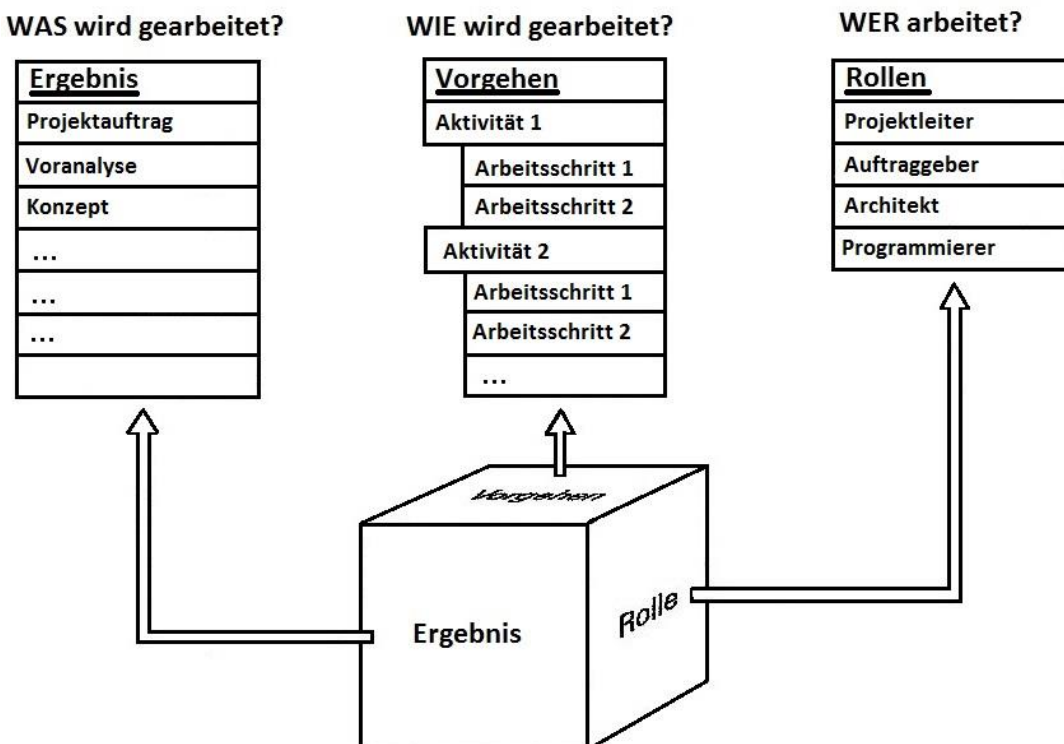


Abbildung 2: Phasen eines IT-Kleinprojektes



Für jede Phase beschreibt Hermes die zu erarbeitenden **Ergebnisse** (inkl. Dokumente), die dafür nötigen **Aktivitäten** und die dafür vorgesehene Rolle im Sinne eines Vorschlags. Als Projektleiter erhalten Sie damit eine wertvolle Planungshilfe für eigene IT-Projekte.

Abbildung 3: Beschreibungsobjekte bei HERMES



Hermes ist ein umfassendes und mächtiges Vorgehensmodell für IT-Projekte, das für die Abwicklung eines Kleinprojekts entsprechend «abgespeckt» werden kann bzw. muss. Die Anpassung eines Vorgehensmodells an ein konkretes Projekt nennt man auch „Tailoring“.

Rational Unified Prozess (RUP)



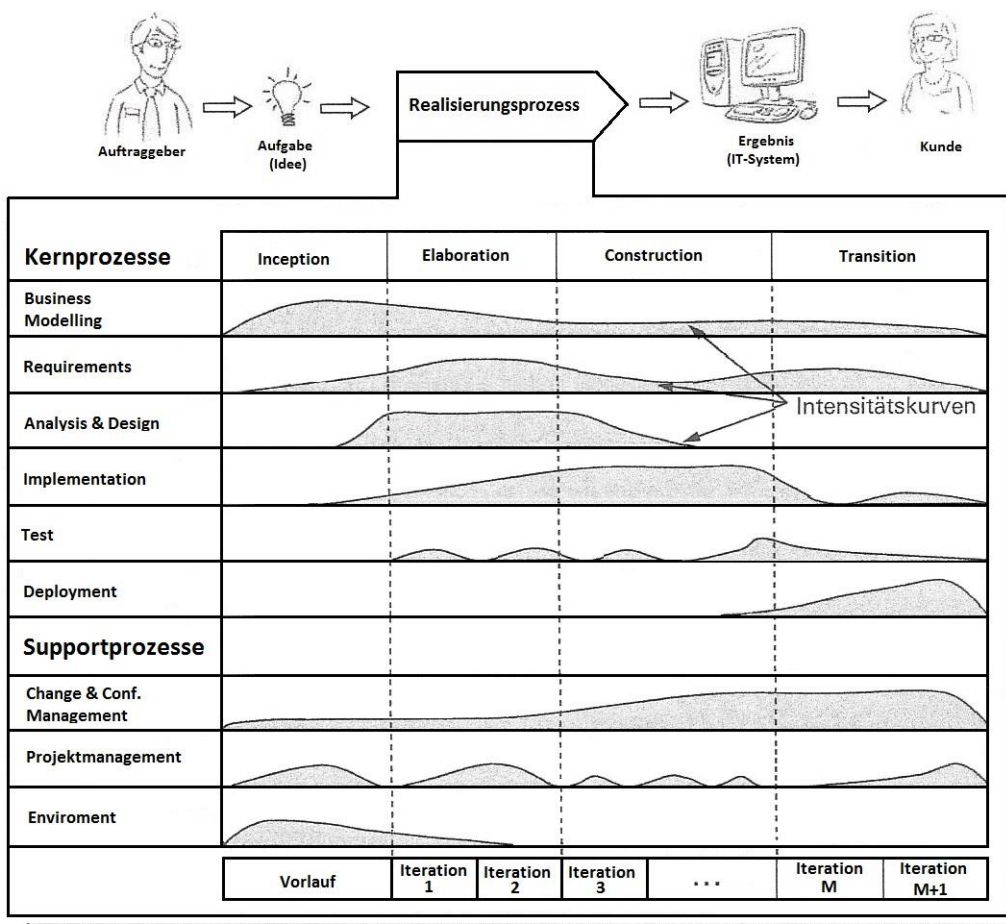
Die RUP-Methode wurde von der Firma Rational entwickelt und eignet sich besonders für die Führung von Softwareprojekten, da sie stark auf die Bedürfnisse der Softwareentwicklung ausgerichtet ist und entsprechende Softwareunterstützung bietet. RUP umfasst diverse Richtlinien, Vorgaben und Checklisten und eine automatisierte Tool-Unterstütz., Die

Methode unterscheidet die Beschreibungsobjekte **Rolle, Aktivität und Ergebnis** und gliedert die Projektführung in die Dimensionen Disziplinen und Phasen. Disziplinen sind wichtige (grössere) Aufgabenbereiche in einem Projekt während die Phasen einzelne r Sequenzen des Projektablaufs darstellen. Letztere haben folgende Bedeutungen:

- **Inception:** Beginn bzw. Initialisierung des Projekts mit den Kernprozessen «Fachliche Aufgabenstellung verstehen und modellieren»
- **Elaboration:** Ausarbeitung des Projekts mit den Kernprozessen «Anforderungen analysieren und Lösung entwerfen»
- **Transition:** Inbetriebnahme der Projektergebnisse mit den Kernprozessen «Lösungen implementieren und testen und ausliefern»

Je nach Phase sind die einzelnen Disziplinen in unterschiedlicher **Intensität** auszuführen.

Abbildung 4: Intensität bei den Kern- und Supportprozessen nach RUP



Erläuterungen zur Grafik

- Die Intensität des Kernprozesses **Business Modelling** ist bei Projektbeginn hoch und nimmt danach kontinuierlich ab.
- Der Supportprozess **Project Management** ist eine Disziplin, die in allen Phasen eine zu- und abnehmende Intensität aufweisen.
- Die Phase **Inception** dient als Vorlaufphase. Danach wird iterativ entwickelt. Iteration 1 bis Iteration N bedeutet, dass in jeder Phase entworfen, programmiert und getestet wird, damit möglichst schnell vorzeigbare Zwischenergebnisse entstehen (z.B. eine lauffähige Applikation mit bestimmten Funktionen).

Der Anwendung eines iterativen Vorgehensmodells bei der Softwareentwicklung liegt die Erfahrung zugrunde, dass es kaum möglich ist, gleich zu Beginn alle Funktionen genau so zu definieren, dass sie bis zum Ende des Realisierungsprozesses gültig sind. Vielmehr zeigt die Praxis, dass im Verlauf eines Entwicklungsprojekts immer wieder neue Wünsche aufkommen und neue Anforderungen zu berücksichtigen sind. Entsprechend spielen die Supportprozesse **Change- & Configuration Management** von Beginn an eine wichtige Rolle und deren Bedeutung nimmt in der Phase «Construction» laufend zu, wenn früh erste Lösungsansätze zu erkennen sind.

Scrum



Improving the Profession of Software Development

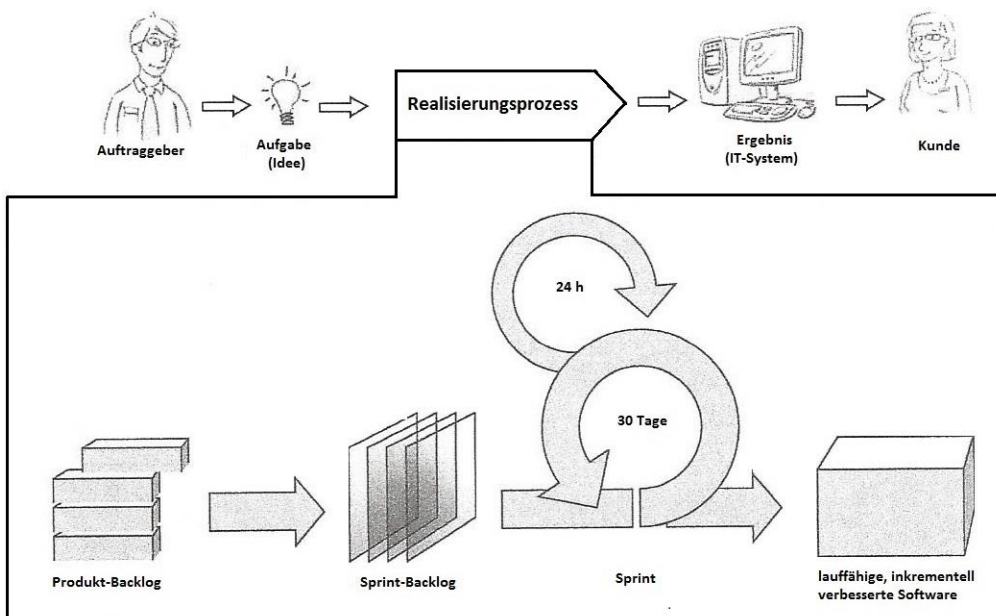
Sequenzielle Vorgehensmodelle setzen voraus, dass ein Projekt von Anfang bis Ende planbar ist und die Projektergebnisse frühzeitig und weitgehend vollständig spezifiziert werden können. Agile Vorgehensmodelle wie **Scrum** dagegen gehen davon aus, dass bei Beginn eines Projekts vieles noch unsicher ist und erst während des Projekts geklärt werden kann. Sie eignen sich daher besonders gut für komplexe IT-Projekte.

Eine Besonderheit des Scrum-Verfahrens besteht darin, dass sich die **Entwicklerteams selber organisieren** und kein Projektleiter vorgesehen ist. Entsprechend hoch sind die Anforderungen an alle Projektbeteiligten.

Bei Scrum wird der Projektverlauf in feste Zeitabschnitte, sogenannte «Sprints» unterteilt. In jedem Sprint wird das Endprodukt inkrementell weiterentwickelt. Ein Sprint kann erst abgeschlossen werden, wenn die für diesen Zeitabschnitt geplante Funktionalität fertig entwickelt und abgenommen worden ist. Gleichzeitig gelten bei Scrum die in der Softwareentwicklung bewährten Schritte «Anforderungsspezifikation - Design - Implementierung - Test». Diese stellen aber keine eigenen Phasen dar, sondern finden jeweils innerhalb eines Sprints statt.

Die spezifizierten Anforderungen werden im **«Product Backlog»** festgehalten. Dieser umfasst somit die komplette Funktionalität des erwarteten Endprodukts. Funktionen oder Teilfunktionen, die in einem bestimmten Sprint zu realisieren sind, werden in den **«Sprint Backlog»** übernommen. Folgende Grafik verdeutlicht dieses Arbeitsprinzip:

Abbildung 5: Arbeitsprinzip bei Scrum



Ein Sprint-Zyklus dauert typischerweise **30 Arbeitstage**, wobei täglich (im «Daily Scrum») die Aktivitäten der nächsten **24 Stunden** geplant werden.

Ein Projekt, das nach Scrum abgewickelt wird, muss folgende **Rollen** besetzen:

- Der **Product Owner** trägt die Verantwortung für das Projektergebnis, definiert dessen Eigenschaften und füllt den Sprint-Backlog entsprechend seinen Prioritäten.
- Das **Scrum-Entwickerteam** realisiert das Projektergebnis unter Einhaltung der vereinbarten Qualitätsstandards.
- Der **Scrum Master** stellt sicher, dass die Regeln des Scrum-Verfahrens eingehalten werden und räumt allfällige Hindernisse aus dem Weg.

Ohne Planung und Koordination geht es auch bei einem Scrum-Projekt nicht. An die Stelle von Projektsitzungen treten hier folgende **«Zeremonien»**:

- Beim **Sprint Planning** wird der nächste Sprint geplant. Der Planungsprozess findet in zwei Sitzungen statt. Im ersten Meeting werden die Anforderungen aus dem Product Backlog vorgestellt. Dabei stehen **«User Stories»** im Vordergrund, die die gewünschte Funktionalität aus Kundensicht beschreiben. Im zweiten Meeting bespricht das Team Aspekte der Umsetzung und zerlegt die Entwicklungsarbeiten in Einheiten, die nicht länger als einen Tag dauern sollten.
- Der **Daily Scrum** ist ein kurzes, tägliches Meeting des Projektteams, um Informationen auszutauschen, anstehende Arbeiten zu koordinieren und ggf. bestehende Probleme zu lösen. Innerhalb eines Sprints müssen bestimmte Tasks gelöst und abgeschlossen werden. Für das Projektteam bedeutet dies, dass man einander unterstützen muss und jedem hilft, der in Verzug gerät, bevor man den nächsten Task angeht.
- Am Ende eines Sprints findet das **Sprint Review** statt. Hier wird überprüft, ob die Ziele des Sprints erreicht wurden. Dabei ist es die Aufgabe des Product Owners, zusammen mit den Endbenutzern neu entwickelte Funktionen zu testen. Unfertige Funktionen wandern zurück in das Product Backlog und werden neu geplant.
- In der **Retrospektive** zwischen einzelnen Sprint werden Erfahrungen ausgetauscht und Verbesserungsmöglichkeiten identifiziert.