

Einführung in das Projektmanagement

<https://vfhpmp.eduloop.de>

Stand 2021-01-04 11:24:36

Inhalt

Einführung in das Projektmanagement	6
1 Einführung	7
1.1 Motivation	7
1.2 Begriffe	10
1.2.1 Aufgaben eines Unternehmens	10
1.2.2 Der Projektbegriff	11
1.2.3 Der Begriff Projektmanagement	12
1.2.4 Aufgaben zum Abschnitt Begriffe	14
1.3 Projektphasen und Prozessmodelle	15
1.3.1 Projektphasen	15
1.3.2 Softwareentwicklungsprozess	17
1.3.3 Wasserfallmodell	19
1.3.4 Spiralmodell	20
1.3.5 V-Modell	21
1.3.6 Rational Unified Process (RUP)	22
1.3.7 EXtreme Programming (XP)	23
1.3.8 Schlussfolgerungen	25
1.3.9 Aufgaben zu Projektphasen und Prozessmodellen	27
2 Projektstart	29
2.1 Projektziele	29
2.1.1 Projektziele beschreiben	29
2.1.2 Wirtschaftlichkeitsrechnung	34
2.1.3 Konfliktmanagement	36
2.1.4 Verträge	39
2.1.5 Aufgaben zu Projektziele	40
2.2 Risiken in Softwareprojekten	41
2.2.1 Risikoanalyse	41
2.2.2 Risikomanagement	45
2.2.3 Aufgaben zu Risiken in Softwareprojekten	45
2.3 Projektorganisation	46
2.3.1 Motivation der Projektorganisation	47
2.3.2 Organisationsstrukturen	47
2.3.3 Projektgremien	52
2.3.4 Aufgaben zur Projektorganisation	55
3 Projektplanung	57
3.1 Grundlagen der Projektplanung	57
3.1.1 Definition	57

3.1.2 Meilensteine	59
3.1.3 Personal	60
3.1.4 Aufgaben zu den Grundlagen der Projektplanung	61
3.2 Planungsreihenfolge	62
3.2.1 Vorgehen und Motivation	63
3.2.2 Strukturplanung	64
3.2.3 Einsatzmittelplanung	69
3.2.4 Terminplanung	71
3.2.5 Kostenplanung	73
3.2.6 Aufgaben zur Planungsreihenfolge	75
3.3 Planungstechniken	76
3.3.1 Probleme der Aufwandsschätzung	77
3.3.2 FP-Analyse	78
3.3.3 CoCoMo-II	83
3.3.4 Netzplantechnik	86
3.3.5 Balkendiagramme	91
3.3.6 Aufgaben zu den Planungstechniken	92
4 Projektkontrolle	95
4.1 Voraussetzungen der Projektkontrolle	95
4.1.1 Erhebung der Ist-Daten	97
4.1.1.1 Tätigkeitsbericht	97
4.1.1.2 Reviews	100
4.1.2 Aufgaben zu den Voraussetzungen der Projektkontrolle	103
4.2 Kontrollgrößen und Metriken	104
4.2.1 Termine	104
4.2.2 Kosten und Aufwand	106
4.2.3 Plan-Ist-Vergleich	107
4.2.4 Sachfortschritt	109
4.2.5 Qualität	113
4.2.6 Aufgaben zu Kontrollgrößen und Metriken	117
5 Projektabschluss	119
5.1 Produktübergabe	119
5.1.1 Produktabnahme	119
5.1.2 Produktbetreuung	123
5.1.3 Aufgaben zur Produktübergabe	124
5.2 Projektanalyse	125
5.2.1 Abweichungs- und Wirtschaftlichkeitsanalyse	125
5.2.2 Erfahrungssicherung	128

5.2.3 Aufgaben zur Projektanalyse	130
6 Teamführung	132
6.1 Motivationstheorien	132
6.1.1 Wichtige Begriffe	132
6.1.2 Menschentypen	135
6.1.3 Einige Beispiele für Motivationstheorien	138
6.1.4 Aufgaben zu Motivationstheorien	142
6.2 Führungshinweise	142
6.2.1 Führungstheorien	143
6.2.2 Teamentwicklung	146
6.2.3 Praktische Hinweise	153
6.2.4 Aufgaben zu Führungshinweise	153
7 Anhang	155
7.1 Beispiel zu Kostenvergleichs-, Gewinnvergleichs- und Amortisationsrechnung	155
7.2 Beispiel zur Nutzwertanalyse	156
7.3 Beispiel für einen Projektvertrag zur Erstellung einer Animation	157
7.4 Formular Projektauftrag (manuelle Bearbeitung)	159
7.5 Formular Projektauftrag (computergestützt)	160
7.6 Use-Case Schablone	161
7.7 Beispiel zur Projektstrukturplanung	162
7.8 Planungshinweise für das XP-Prozessmodell	166
7.9 Beispiel zur Netzplantechnik	168
7.10 Beispiele für Checklisten	170
8 Zusatzmaterialien	174
8.1 Lernspiel	174
8.1.1 Das erste Meeting	174
8.1.2 Analyse des Meetings	175
8.1.3 Weitere Szenarien eines Meetings	178
8.2 Videos	180
8.2.1 Video - Interview mit einem Projektleiter	180
8.2.2 Video - Projektorganisation	182
8.2.3 Video - Erhebung der Projektanforderungen	185
8.2.4 Video - Führungsstile	186

Anhang

I Literaturverzeichnis	188
II Abbildungsverzeichnis	193
III Tabellenverzeichnis	199

IV Aufgabenverzeichnis	200
------------------------------	-----



Gliederung

Einführung in das Projektmanagement

[Einführung in das Projektmanagement](#)

[1 Einführung](#)

[2 Projektstart](#)

[3 Projektplanung](#)

[4 Projektkontrolle](#)

[5 Projektabschluss](#)

[6 Teamführung](#)

[7 Anhang](#)

[8 Zusatzmaterialien](#)

Diese Seite enthält die Vorlesungsunterlagen des Kurses "Einführung in das Projektmanagement" des Online-Studiengangs Medieninformatik (B.Sc.)

Die Unterlagen können über die **rechte Navigation** der Seite eingesehen werden.

1 Einführung

Gliederung
1
1.5
1.1.2
1.1.5
2

- 1 Einführung
- 1.1 Motivation
- 1.2 Begriffe
- 1.3 Projektphasen und Prozessmodelle



Lernziele

Was sollten Sie nach diesem Kapitel wissen?

- Was ist ein Projekt?
- Was/wer gehört zu einem Projekt?
- Was ist Projektmanagement?
- Welche Schlussfolgerungen lassen sich ziehen?

1.1 Motivation

In diesem Lehrbuch stehen **Softwareprojekte** mit ihren Brancheneigenheiten im Mittelpunkt. Das Internet prägt die Softwarebranche und erzeugt einen hohen Innovationsdruck. Die technologische Komplexität, ein ständiger Zuwachs der Anwendungsbereiche und individuellen Lösungen, rasante Entwicklungen bei Plattformen und Entwicklungswerkzeugen prägen heutige Softwareprojekte. "Time to Market" ist ein entscheidendes Kriterium für erfolgreiche Projekte. Drei Faktoren haben dabei großen Einfluss:



Wichtig

- der gewählte Entwicklungsprozess
- die MitarbeiterInnen und
- die verwendete Technologie.

Der **Entwicklungsprozess** ist abhängig vom jeweiligen Projekt. Softwareprojekte, bei denen die Zeit im Vordergrund steht, werden am besten evolutionär mit **iterativen Prozessen** realisiert. Mehr dazu erfahren Sie im Abschnitt "Projektphasen und Prozessmodelle".

Neben dem richtigen Entwicklungsprozess spielen die **Mitarbeiter** eine entscheidende Rolle. Auch mit dem besten Prozessmodell werden Sie scheitern, wenn Sie schlecht motivierte und unfähige Mitarbeiter eingestellt haben. Die Leistungsbereitschaft von Mitarbeitern lässt sich bekanntlich schlecht einschätzen, unterliegt großen

Schwankungen und ist von vielen Faktoren abhängig. Diese Faktoren liegen teilweise außerhalb des Unternehmensumfeldes und können durch den Projektleiter nur begrenzt beeinflusst werden. Das Kapitel "Teamführung" enthält viele Hinweise für Sie, um Mitarbeiter auszuwählen, zu motivieren und deren eventuelles Fehlverhalten richtig einzuordnen.

Moderne Technologien für die System-Modellierung und -Programmierung helfen, Fehler zu vermeiden und effektiv zu arbeiten. Sie sind damit ein wichtiger Faktor für die erfolgreiche Abwicklung von Projekten. In der Softwarebranche hat sich die **objektorientierte Vorgehensweise** durchgesetzt. Die Objektorientierung hat sich seit 1985 erfolgreich entwickelt und ist heute die vorherrschende Technologie. Hier dazu einige Daten:



Hinweis

- Start mit der ersten praktisch einsetzbaren objektorientierten Programmiersprache Smalltalk-80 von 1970–1980
- seit Beginn der 90iger C++ und ab 1996 Java
- erste Bücher zur Objektorientierten Analyse (OOA) 1990, Booch, Coad, Yourdon, Jacobson
- 1989 wurde die OMG gegründet (Object Management Group)
- 1991 wurde die ODMG gegründet (Object Database Management Group)

Seit 1997 gibt es eine einheitliche **Notation für die Modellierung** – die **UML** (Unified Modelling Language). Gründungsväter der UML sind GRADY BOOCH und JIM RUMBAUGH. Damit ist eine wesentliche technologische Voraussetzung für erfolgreiche Softwareprojekte gegeben. Es wird in vielen Abschnitten darauf Bezug genommen. Mehr dazu können Sie in den beiden Büchern von BOOCH [[Boo95](#)] und FOWLER [[Fow98](#)] nachlesen.

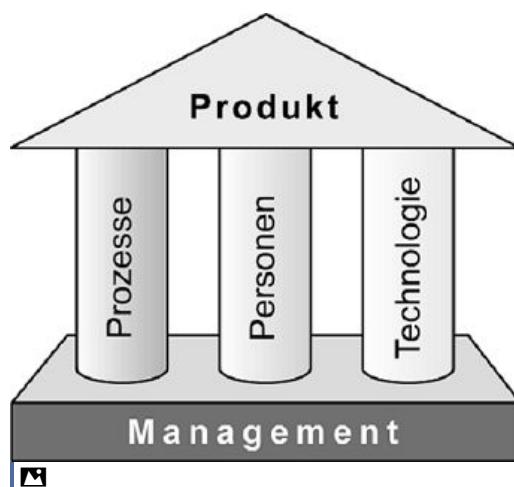
Das **Projektmanagement** mit den verschiedenen Teilaufgaben, wie



Wichtig

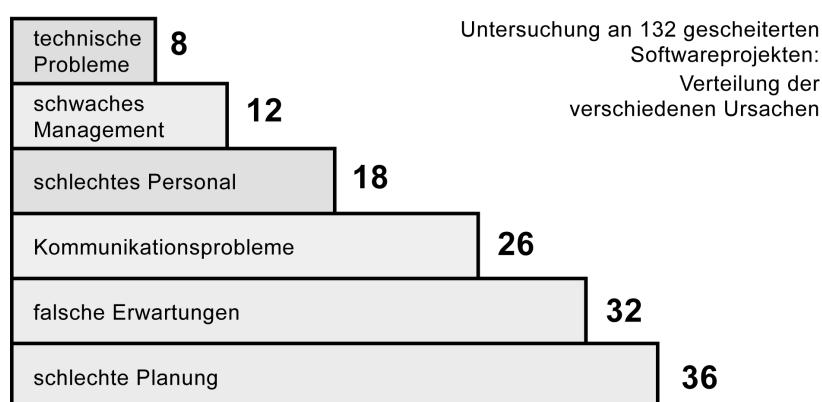
- Termin-, Kosten- und Qualitätsplanung
- Personalmanagement
- Risiko-, Change-, Konfigurationsmanagement
- Projektkontrolle und -steuerung

stellt praktisch die Plattform für die Faktoren Prozesse, Personen und Technologien bereit. Es bildet das Fundament und ist verantwortlich für die richtige Auswahl der Mitarbeiter, des Prozesses und der Technologie.



In den vier zentralen Kapiteln "Projektstart", "Projektplanung", "Projektsteuerung" und "Projektabchluss" werden moderne Managementmethoden und -techniken beschrieben.

In der Softwarebranche stellten MUMMERT+PARTNER [Mum00] Untersuchungen an und trafen folgende Feststellung: 75% der Projekte erreichen nicht das geplante Ziel und 40% aller IT-Projekte scheitern. Diese Zahlen sind erschreckend. Das folgende Bild zeigt eine Analyse der Ursachen: Es sind nicht technische, sondern – wie ersichtlich – in den meisten Fällen Managementprobleme.



Neben den traditionellen Wettbewerbsfaktoren "Kosten" und "Zeit" spielt die Wettbewerbskomponente "**Qualität**" eine entscheidende Rolle, da davon auszugehen ist, dass auf lange Sicht der Erfolg eines Unternehmens aus der überlegenen Qualität seiner Produkte resultiert. Aus diesem Grund haben die meisten Unternehmen ein **Qualitäts-**

managementsystem eingeführt, das unter anderem das Vertrauen der Kunden in die Qualitätsfähigkeit des Unternehmens stärken soll.

Dieses Online-Modul enthält viele Informationen, die Sie brauchen, damit Ihr nächstes Softwareprojekt erfolgreich verläuft. Neben zahlreichen Erfahrungen aus der Praxis sind darin aktuelle Veröffentlichungen und natürlich auch das gesicherte Wissen aus der Standardliteratur eingeflossen.

Das Modul basiert in großen Teilen auf dem Lehrbuch [Buh04], das Grundwissen zum Projektmanagement vermittelt, wobei der Fokus auf Software-Projekten liegt.

1.2 Begriffe



Gliederung

1.2 Begriffe

- [1.2.1 Aufgaben eines Unternehmens](#)
- [1.2.2 Der Projektbegriff](#)
- [1.2.3 Der Begriff Projektmanagement](#)
- [1.2.4 Aufgaben zum Abschnitt Begriffe](#)



Lernziele

Was sollten Sie nach diesem Abschnitt wissen?

- Welche Arten von Aufgaben gibt es in einem Unternehmen?
- Was/wer gehört zu einem Projekt?
- Was ist Projektmanagement?

1.2.1 Aufgaben eines Unternehmens

Die Gründung und das Bestehen eines Unternehmens sind mit der Realisierung einer Leistung oder der Herstellung eines Produkts verbunden, wobei ein Gewinn erzielt und dieser maximiert werden soll.

Das Tagesgeschäft eines Unternehmens wird in sogenannten Daueraufgaben realisiert. Um die **Daueraufgaben** erfüllen zu können, gliedert sich ein Unternehmen z.B. in Abteilungen und Gruppen. Die Gliederung eines Unternehmens wird unter dem Begriff **Aufbauorganisation** zusammengefasst. Beispiele für Abteilungen mit typischen Daueraufgaben sind:



Wichtig

- der Vertrieb,
- das Rechnungswesen,

- die Fertigung eines bereits eingeführten Produkts und
- der Kundensupport.

Die Entwicklung neuer Produkte, z.B. neuer Informationssysteme, die Anpassung von Maschinen u.ä. bezeichnet man als **Sonderaufgaben**, da sie sich nicht wie eine Daueraufgabe im Tagesgeschäft eines Unternehmens wiederholen. Eine Sonderaufgabe ist also insbesondere durch ihre Einmaligkeit und besondere Bedeutung für das Unternehmen gekennzeichnet. Zu den Sonderaufgaben zählen:



Wichtig

- Forschungs- und Entwicklungsaufgaben,
- Softwareerstellung,
- Schaffung neuer Vertriebswege,
- Schiffsbau (keine Serienproduktion).

In Softwareunternehmen finden sich Daueraufgaben hauptsächlich für die betriebswirtschaftlichen Prozesse, wie z.B. Marketing oder Rechnungswesen. Auch der Support installierter Einheiten kann als Daueraufgabe realisiert werden.

Der Hauptprozess – die Softwarereproduktion – läuft in Form von Sonderaufgaben ab. Dies wird notwendig, da die Produktion von Software durch folgende Besonderheiten gekennzeichnet ist:



Wichtig

- Software ist nicht sichtbar und sehr komplex. Die Komplexität erlaubt kein durchweg standardisiertes Vorgehen.
- Software wird in einem interaktiven, komplexen und innovativen Herstellungsprozess erstellt.

1.2.2 Der Projektbegriff

Eine Sonderaufgabe kann unter bestimmten Bedingungen als **Projekt** bezeichnet werden. Für die Erläuterung des Projektbegriffes findet sich in der DIN 69901 folgende Definition:



Definition

Ein Vorhaben, was im Wesentlichen durch seine Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wird als Projekt bezeichnet. Ein Projekt ist eine Sonderaufgabe, die mit der vom Auftraggeber geforderten Qualität in einer bestimmten Zeit mit vorgegebenen Mitteln zu lösen ist.

Qualität ist die Gesamtheit von Eigenschaften und Merkmalen eines Produktes oder einer Tätigkeit, die sich auf die Eignung zur Erfüllung gegebener Erfordernisse beziehen. Nach KUPPER [Kup96] kann man ein Projekt anhand der folgenden Punkte erkennen (bei ihm als Grundregeln für die Durchführung von Projekten bezeichnet):

! Wichtig

- Es existiert ein Auftraggeber.
- Es gibt eine Zielformulierung.
- Es gibt eine Projektgruppe.
- Es gibt einen Projektleiter.

Nach LUDEWIG [Lud91] gibt es im DV-Bereich folgende Projektarten:

! Wichtig

- Auftragsprojekte,
- interne Projekte und
- Entwicklungsprojekte.

Von **Auftragsprojekten** spricht man, wenn die Software für einen externen Auftraggeber entwickelt wird. "Die Rollen sind klar verteilt, zwischen dem Hersteller und Auftraggeber besteht ein Vertrag, in dem im Allgemeinen Lieferungen und Zahlungen verknüpft sind."

Bei **internen Projekten** wird Software im Hause für den eigenen Bedarf entwickelt, z.B. mit dem Ziel der Rationalisierung interner Abläufe. "Hersteller und Auftraggeber sind in derselben Organisation, die Bezahlung erfolgt mit 'Papiergegeld'. Konflikte werden durch den gemeinsamen Vorgesetzten gelöst."

Bei **Entwicklungsprojekten** wird ein Softwareprodukt im Auftrag z.B. der Marketing-Abteilung entwickelt, welches auf dem Markt angeboten werden soll. "Das Geld kommt aus dem Entwicklungsbudget" [Lud91].

1.2.3 Der Begriff Projektmanagement

Vom Begriff Management, das die Leitung sozio-technischer Systeme in personen- und sachbezogener Hinsicht mit professionellen Mitteln und Methoden umfasst, wird der Begriff **Projektmanagement** abgeleitet.

Der Begriff Projektmanagement soll hier wie folgt verwendet werden:



Definition

Projektmanagement ist ein Leitungs- und Führungskonzept für Projekte, welches

- den Entwicklungsprozess definiert,
- die notwendigen Aufgaben definiert,
- die Methoden für die Lösung der Aufgaben vorschlägt,
- Institutionen schafft und nutzt, von denen diese Aufgaben realisiert werden können und
- abteilungsübergreifend arbeitet.



Wichtig

Nach KUPPER [*Kup96*] bezeichnet die Kunst der **Projektsteuerung**:

- die Kunst, zu einem fest vereinbarten Zeitpunkt ein Projekt abzuschließen
- die Kunst, viele Menschen zu leiten und zu organisieren
- die Kunst, in die Zukunft zu schauen

und unter dem Aspekt der Leitung



Wichtig

- zu wissen, was man erreichen will
- den Weg zu kennen, wie man das erreicht
- Erfahrungen richtig zu verwerten
- eine Lage richtig zu beurteilen
- zu überzeugen und sich durchzusetzen
- das Ganze zu sehen, ohne das Einzelne zu übersehen

Nach BECK [*Bec00*] ist Projektmanagement die Kunst, mit den Ängsten und Rechten sowie Verantwortlichkeiten der Kunden und Entwickler richtig umzugehen:

"If we are going to develop well, we must create a culture that makes it possible for programmers and customers to acknowledge their fears and accept their rights and responsibilities."

Das Projektmanagement hat immer in erster Linie mit Menschen zu tun, ob Kunde, Projektleiter oder Mitarbeiter. Die mit den Personen zusammenhängenden Faktoren, die Einfluss auf den Erfolg haben, bezeichnet man als weiche Faktoren des Projektmanagements. Dagegen lassen sich die harten Faktoren technisch klar abgrenzen und im Gegensatz zu den weichen Faktoren leichter beherrschen.

Tabelle 1.2.3.1 nennt einige weiche und harte Faktoren des Projektmanagements:

Weiche Faktoren	Harte Faktoren
<ul style="list-style-type: none"> • Qualifizierung des Projektleiters • Mitarbeiter auswählen, leiten und motivieren • gute Instinkte und Menschenkenntnisse • Erfahrung • Ausgestaltung des Entwicklungsprozesses 	<ul style="list-style-type: none"> • Techniken • Tools • Versionskontrolle • Formalisierte Schätzverfahren

| ■■■ Tabelle 1.2.3.1: Weiche und harte Faktoren des Projektmanagements

Aus Abbildung 1.1.2 im Abschnitt 1.1 geht hervor, dass Softwareprojekte häufig aufgrund der weichen Faktoren scheitern. Sie werden ganz einfach unterschätzt. Erfolgreiche Softwareprojekte sind durch die Beherrschung beider Gruppen (weiche und harte Faktoren) gekennzeichnet. Ausführliche Informationen zu den weichen Faktoren und Fähigkeiten des Projektmanagements finden Sie in Kapitel 6 Teamführung.

1.2.4 Aufgaben zum Abschnitt Begriffe



Aufgabe

| ■■■ Aufgabe 1.2.4-1

Welche Faktoren haben nach Kent Beck wesentlichen Einfluss auf den Erfolg eines Softwareprojekts?

- A) weiche Faktoren
- B) harte Faktoren



Aufgabe

| ■■■ Aufgabe 1.2.4-2

Nennen Sie drei Hauptursachen für das Scheitern von Softwareprojekten!

**Aufgabe****Aufgabe 1.2.4-3**

Geben Sie eine Definition für den Begriff Projektmanagement!

**Gliederung**

1.3 Projektphasen und Prozessmodelle

1.3 Projektphasen und Prozessmodelle

1.3.1 Projektphasen

1.3.2 Softwareentwicklungsprozess

1.3.3 Wasserfallmodell

1.3.4 Spiralmodell

1.3.5 V-Modell

1.3.6 Rational Unified Process (RUP)

1.3.7 EXtreme Programming (XP)

1.3.8 Schlussfolgerungen

1.3.9 Aufgaben zu Projektphasen und Prozessmodellen

**Lernziele**

Was sollten Sie nach diesem Abschnitt wissen?

- In welche Phasen lässt sich der Projektlauf untergliedern?
- Welche Prozessmodelle für die Softwareentwicklung gibt es und was sind deren charakteristische Eigenschaften?
- Wie sieht das Wasserfallmodell aus?
- Was sind die wesentlichen Merkmale des Spiralmodells?
- Welche charakteristischen Eigenschaften hat das V-Modell 97?
- Was versteht man unter dem Begriff "Rational Unified Process" (RUP)?
- Was sind die Besonderheiten des EXtreme Programmings (XP)?

1.3.1 Projektphasen

Von der Initiierung eines Projekts bis zu seinem erfolgreichen Abschluss durchläuft es verschiedene Projektphasen. Die Phasen unterteilen den **zeitlichen Ablauf** für das Projektmanagement. Im Allgemeinen wird mit der Ausarbeitung der Projektziele und -inhalte begonnen und daran anschließend muss die Planung von Arbeitspaketen mit

Terminen, Kosten u.a. vorgenommen werden. Nach dieser Planungsphase sind die Aufgaben zu kontrollieren und zu steuern sowie je nach Notwendigkeit Plananpassungen vorzunehmen. Ob erfolgreich oder nicht erfolgreich – eine Auswertung zum Abschluss eines Projekts sollte die Ursachen aufzeigen und den Projektablauf dokumentieren.

Die folgenden Phasen können somit spezifiziert werden:



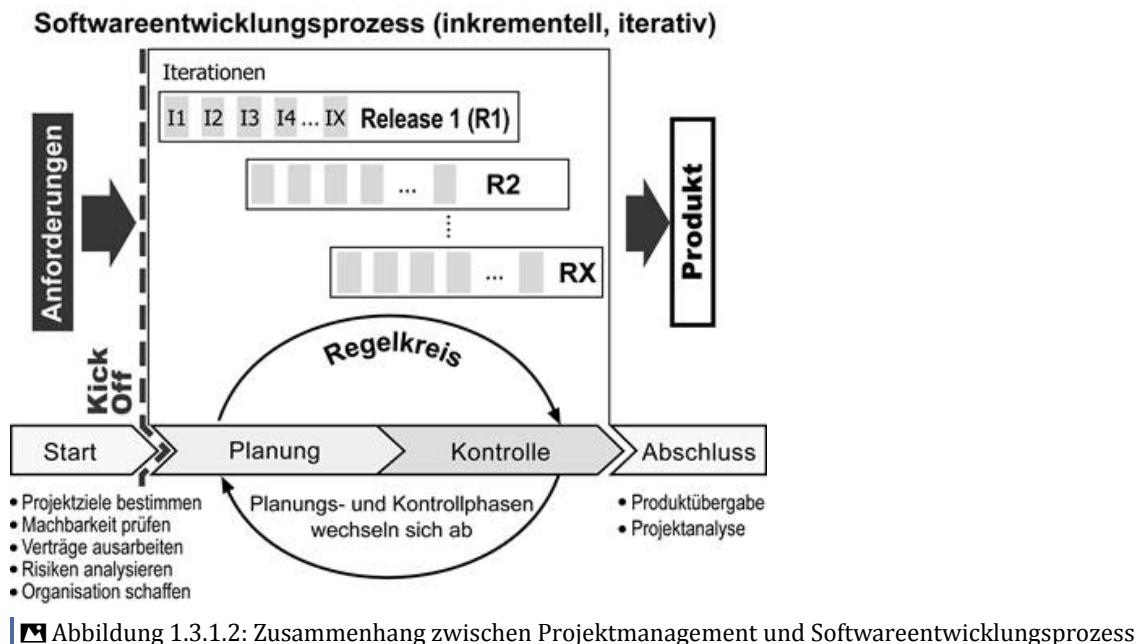
| Abbildung 1.3.1.1: Projektphasen

Innerhalb dieser Projektphasen können mit Hilfe von Methoden **Modelle** oder **Artefakte** erzeugt werden. Ein Projekt hat also neben der oben beschriebenen zeitlichen Dimension auch eine **Methodenachse**. Solche Artefakte sind beispielsweise: die Projektzielanalyse, die Risikoanalyse, Netzpläne, Tätigkeitsberichte, Reviews oder eine Meilensteintrendanalyse.

Projekte können sehr unterschiedliche Eigenheiten haben. Im Gegensatz zu Bauprojekten oder anderen Projekten aus den Ingenieurwissenschaften lassen sich die meisten Softwareprojekte unmöglich vom Start bis zum Abschluss planen und abarbeiten. Das liegt vor allem daran, dass sich die Anforderungen nicht vollständig festhalten lassen. PHILIPPE KRUCHTEN hat in einem Artikel des ObjektSpektrum 4/2001 [Kru01] folgende Gründe genannt:

- Software bietet von Natur aus fast unbegrenzte Flexibilität.
- Psychologische und organisatorische Zwänge erschweren bzw. verhindern es, ein Softwaresystem allein auf der Grundlage von Vermutungen vollständig, korrekt und zutreffend zu definieren (ohne entsprechendes Feedback in den Bearbeitungszyklen).
- Die Spezifikationssprachen sind häufig ungenau.
- Entwürfe und Implementierungen sind oft mangelhaft.
- Sich schnell ändernde Marktanforderungen machen häufig späte Änderungen erforderlich.

Um dem Rechnung zu tragen, muss ein geeigneter Prozess definiert werden – auch **Softwareentwicklungsprozess** genannt. Die nachfolgenden Abschnitte geben einen Überblick über die heute existierenden Prozessmodelle für die Softwareentwicklung. Diese Prozessmodelle werden durch das Projektmanagement begleitet. Abbildung 1.3.1.2 zeigt den Zusammenhang zwischen Projektmanagement und Softwareentwicklungsprozess.



1.3.2 Softwareentwicklungsprozess

Die Geschichte der Prozessmodelle für Software beginnt schon in den sechziger Jahren und hält bis zum heutigen Tag an. Demzufolge hat sich eine Reihe von Modellen eingebürgert, auf die weiter unten eingegangen wird. Zunächst werden jedoch einige allgemeine Aussagen gemacht, die auf alle Modelle zutreffen.

Die Entwicklung der Software beginnt im Allgemeinen mit der Formulierung der **Anforderungen**, die aus Aufgabenstellungen im Geschäftsbereich resultieren. In vielen Fällen müssen dazu die Geschäftsprozesse erst analysiert werden, um zu ermitteln, welche Aufgaben durch die neue Software automatisiert werden können. Die in abstrakten Begriffen dokumentierten Anforderungen lassen sich nicht in einem Schritt auf die konkrete Ebene eines Programms abbilden, ohne dass sich Fehlinterpretationen und Missverständnisse einschleichen – es werden **Modelle** benötigt, Zwischenschritte für eine richtige Lösung. Die **Modelle** dienen dazu, durch geeignete Darstellungen (Notationen) den Problembereich zu formalisieren und zu konkretisieren, sowie durch eine zweckmäßige Zerlegung des Gesamtproblems die Komplexität abzubauen. Modelle werden durch **Methoden** beschrieben. Der Lösungsweg muss also die Festlegung geeigneter Methoden enthalten. Es gibt eine reiche Auswahl von Ansätzen, deren Einsatz von verschiedenen Kriterien abhängt:

- Umfang des Problems,
- Art des Problems (technischer, kaufmännischer, militärischer Art),
- Verflechtung mit anderen Systemen,
- möglicher technischer, finanzieller und personeller Aufwand,

- spätere Benutzer.

Die richtigen Methoden für den Softwareentwicklungsprozess sind nur ein Aspekt, der betrachtet werden sollte. Ein zweiter Aspekt ist der zeitliche Ablauf: Wann und wie oft werden diese Methoden eingesetzt? Reicht es, jedes Modell nur einmal zu erstellen oder müssen Modelle mehrfach erstellt und überarbeitet werden? Die modernen Ansätze der Softwareentwicklung fassen diese als **iterativen Prozess** auf, bei dem die Entwicklungsschritte mehrmals durchlaufen werden. Ein iterativer Softwareentwicklungsprozess erfordert auch einen iterativen Ansatz im Projektmanagement. Es wird ein Grobplan entworfen, eine Detailplanung erfolgt nur für die jeweils folgende Iteration – Planung und Kontrolle wechseln sich ständig ab. Im Allgemeinen werden die folgenden Methoden angewendet:

Die **Geschäftsprozessanalyse** (engl. Business Engineering) dient dazu, Informationen über die Geschäftsprozesse im Anwenderbereich zu erlangen und auszuwerten, um daraus dann Anforderungen für das zu entwickelnde Programm abzuleiten.

Die **Definition der Anforderungen** (engl. Requirements Engineering) soll garantieren, dass die Ziele auch tatsächlich realisiert werden. Es werden die Programmanforderungen für die zu automatisierenden Vorgänge möglichst detailliert und widerspruchsfrei beschrieben. Dazu gehören die zu erbringenden Funktionalitäten, aber auch qualitative Eigenschaften wie eine gute Wartbarkeit des Programms oder eine leicht zu bedienende Benutzeroberfläche. Es müssen also Methoden verwendet werden, die diese Beschreibung in einer sowohl für den Anwender als auch für den Entwickler verständlichen Form liefern, um Missverständnisse zu vermeiden.

Die **Analysemethoden** (engl. Analysis) dienen dazu, die Anforderungen auf Einheiten zu verteilen, die einen Teil dieser Anforderungen dann repräsentieren. Diese Einheiten müssen mit Namen versehen werden und können später durch ein oder mehrere Programmteile realisiert werden – doch darüber wird im Design entschieden.

Die **Designmethoden** (engl. Design) sind erforderlich, um die Programmteile möglichst übersichtlich zu entwerfen, die für eine automatische Abwicklung der Arbeitsvorgänge erforderlich sind.

Die **Implementierungsmethoden** (engl. Implementation) sollen einen fehlerfreien und verständlichen Programmcode erbringen.

Testmethoden (engl. Testing) sollen mögliche Fehler ermitteln.

In den folgenden Abschnitten werden einige spezielle Vorgehensmodelle entsprechend ihrer Entstehung vorgestellt und diskutiert. Jedes Projekt ist anders. Es wird deshalb nur selten gelingen, dass ein Projekt genau nach einem Vorgehensmodell abgewickelt werden kann.

1.3.3 Wasserfallmodell

Das Wasserfallmodell, das in den frühen sechziger Jahren die Ad-hoc-Ansätze für die Softwareentwicklung ersetzte, war das erste systematische Prozessmodell für Softwareentwicklungsprozesse. Es wurde aus den Ingenieurwissenschaften entlehnt und stellte erstmalig die Forderung, sich vor der eigentlichen Implementation auch mit der Analyse und dem Entwurf der Strukturen zu beschäftigen und diese dann zu dokumentieren. Insofern stellt dieses Modell einen ungeheuren Fortschritt in der Softwareentwicklung dar. Es war für die damaligen kleineren Projekte auch gut geeignet.



Abbildung 1.3.3.1: Wasserfalllebenszyklus nach BARRY BOEHM

Wie die Abbildung 1.3.3.1 zeigt, werden in den Phasen Problemanalyse und Systemspezifikation erst alle Anforderungen spezifiziert, bevor die anderen Schritte nacheinander abgearbeitet werden.

Dieses Modell hat sich in der Softwareentwicklung bis auf einige Ausnahmen nicht behauptet. Ein wesentlicher Grund dafür ist die zunehmende Größe der Systeme und die zu berücksichtigenden Anforderungen. Im Wasserfallmodell wird von dem Irrglauben ausgegangen, dass die meisten Anforderungen in einer frühen Projektphase festgelegt werden können. Die auf dem Wasserfall beruhenden Prozessmodelle werden auch als **schwere Prozesse** bezeichnet. PHILIPPE KRUCHTEN hat in seinem Artikel des Objekt-Spektrum 4/2001 [Kru01] diese Prozesse folgendermaßen gekennzeichnet:

- starr und zentral kontrolliert,
- viele Aktivitäten und Artefakte werden in einer bürokratischen Atmosphäre durchgeführt bzw. erzeugt,
- Unmengen an Dokumenten,

- komplizierte, langfristige Planung bis ins Detail,
- signifikanter Verwaltungsoverhead neben der eigentlichen Arbeit,
- stärker am Prozess als an den Menschen orientiert, d.h. Personen werden als austauschbare Teile in einem mechanischen Vorgehen betrachtet,
- nicht adaptiv, sondern vorhersagbar.

Der Wasserfalllebenszyklus hat nur noch für Projekte aus den Ingenieurwissenschaften größere Bedeutung. So wird man den Bau einer Brücke immer noch penibel vorplanen. Realisierte Bauabschnitte lassen sich nur schwer ändern.

Es gibt einige ausgewählte Projekte im Bereich der Softwareentwicklung, die sich nach dem Wasserfallmodell realisieren lassen. Hierzu zählen z.B. die Übernahme von alten Datenbeständen und deren Konvertierung auf ein neues System sowie die Einführung und Anpassung von Standardsoftware für betriebliche Abläufe (Einführungsprojekte). Diese Projekte zeichnen sich dadurch aus, dass sie von kurzer Dauer und wenig komplex sind.

1.3.4 Spiralmodell

Als Gegensatz dazu haben sich in den letzten Jahren die **iterativen Prozessmodelle** durchgesetzt. Der Prozess wird als eine Reihe von Iterationen mit festgelegter Dauer, die mit einer Version (einer Untermenge des zu entwickelnden Systems) enden, verstanden. Die Dauer ist festgelegt (Timeboxing) – falls Ziele nicht erreicht werden, müssen Anforderungen entfernt werden. Der schnellste Weg zu einer Lösung ist in der Regel, eine Reihe von Möglichkeiten auszuprobieren und dann zwischen diesen Alternativen zu entscheiden. Ein iterativer Ansatz berücksichtigt diese Art von Kreativität und direktem Experimentieren. Einer der ersten, die theoretische Arbeit auf diesem Gebiet geleistet haben, war BARRY BOEHM mit seinem Spiralmodell [Boe88].



Abbildung 1.3.4.1: Spirallebenszyklus nach BARRY BOEHM

Das Spiralmodell ist ein Metamodell, das sowohl durch den **Evaluationsaspekt** als auch durch das **Risikomanagement** geprägt wird. Es definiert 4 Schritte, wobei der dritte Schritt "Entwicklung und Validierung des Produkts der nächsten Stufe" ein weiteres Prozessmodell, z.B. das Wasserfallmodell, einschließt. In diesem Schritt findet die eigentliche Softwareentwicklung statt. Das oberste Ziel dieses Modells ist die Risikominimierung durch die Bildung von engen Spiralen mit kleinen Teilentwicklungen. Das Modell gestattet separate Zyklen für verschiedene Komponenten und ist damit ein **echtes iteratives Modell**. Es findet eine periodische Überprüfung der Entwicklung statt und es gestattet deshalb ein frühzeitiges Eliminieren von Fehlern. Es ist sehr flexibel, da es die Einlagerung anderer Prozessmodelle erlaubt. Das Spiralmodell erfordert einen relativ hohen Managementaufwand und ist daher für kleine und mittlere Projekte nicht so gut geeignet.

1.3.5 V-Modell

Das V-Modell [Ver02] ist eine Weiterentwicklung des Wasserfallmodells. Dieses Vorgehensmodell wurde im Auftrag der Bundeswehr entwickelt und auch von den zivilen Behörden übernommen.

Das V-Modell ist komplex. Die zugehörigen Handbuchsammlungen umfassen mehr als 300 Seiten und enthalten

- Entwicklungsstandards,
- Methodenstandards,
- Werkzeuganforderungen.

Die anzufertigenden Dokumente hängen vom Projektzuschnitt ab. Der Einarbeitungs- und Dokumentationsaufwand ist aber erheblich – dafür können sich Wartungs- und Änderungsdienst reduzieren.

Das V-Modell setzt sich aus vier verschiedenen **Submodellen** zusammen, zwischen denen ein enger Zusammenhang besteht. Ergebnisse eines Submodells fließen in ein anderes Submodell ein (Produktfluss).

- System-/Softwareerstellung (SE)
- Projektmanagement (PM)
- Qualitätssicherung (QS)
- Konfigurationsmanagement (KM)

Das V-Modell unterstützt die iterative und objektorientierte Softwareentwicklung. Die Objektorientierung war aber nicht der Ausgangspunkt für die Entwicklung dieses Modells. Sie findet in sogenannten Methodenzuordnungstabellen ihren Ausdruck. Für die im Submodell SE definierten Aktivitäten werden UML-Techniken vorgeschrieben. Für die Durchführung der Ist-Aufnahme und Analyse eignen sich beispielsweise Anwendungsfall- und Aktivitätsdiagramme.

Das V-Modell wird von den öffentlichen Behörden und militärischen Institutionen in Deutschland eingesetzt. International findet es keine Verwendung.

1.3.6 Rational Unified Process (RUP)

Ein weiteres iteratives Prozessmodell ist der **Rational Unified Process (RUP)**, eine Entwicklung der Firma Rational, speziell der Process Development Abteilung unter ihrem Chef PHILIPP KRUCHTEN.

Dieses Prozessmodell zeigt zwei orthogonale Strukturen:

- senkrecht die statische Struktur: **Method Engineering** oder Methodenachse,
- waagerecht die dynamische Struktur: den **Lifecycle** oder Lebenszyklus eines Projekts oder auch Zeitachse.

In der folgenden Abbildung ist die Methodenachse nur grob in Business Engineering usw. unterteilt. Innerhalb dieser groben Struktur lassen sich weitere feinere Methoden definieren, im Business Engineering z.B. die Business Use Case Analyse und die Aktivitätsdiagramme. Bei der Beschreibung dieser Methoden geht dieses Prozessmodell mit den in der Unified Modelling Language (UML) beschriebenen Methoden konform ([Fow98]).

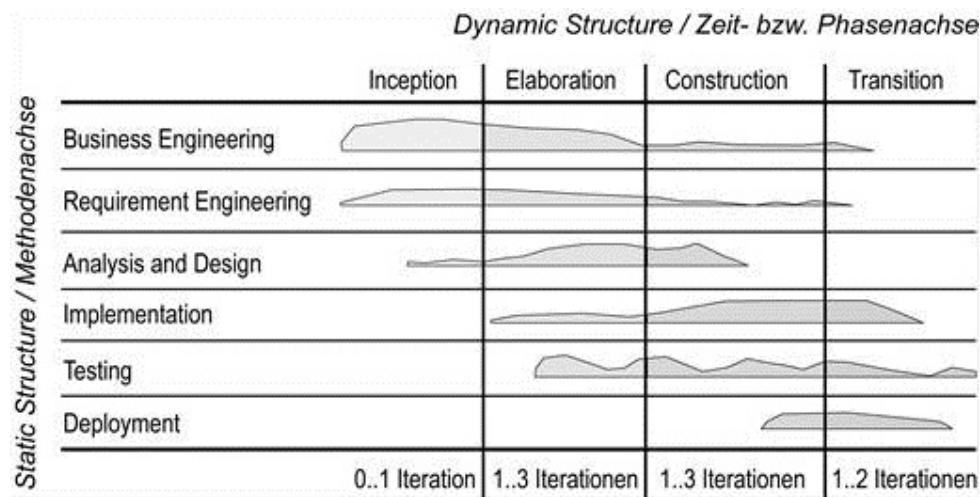


Abbildung 1.3.6.1: Rational Unified Process (RUP), von der Rational Co.

Die einzelnen Phasen auf der zeitlichen Ebene (vgl. Abbildung 1.3.6.1) lassen sich wie folgt definieren:

- **Inception:** Definition der Ziele (Einstieg)
- **Elaboration:** Projekt planen, spezifizieren und Architekturen festlegen (Ausarbeitung)
- **Construction:** Realisierung (Konstruktion)
- **Transition:** Einführung beim Anwender (Überleitung)

Jede dieser Phasen wird in mehrere Iterationen zerlegt. Eine **Iteration** ist eine **Sequenz von Aktivitäten**, die geplant werden, definierten Evaluationskriterien entsprechen und ein Ergebnis in Form eines Releases (intern, extern) erzeugen. Wie aus Abbildung 1.3.6.1 weiter hervorgeht, werden in jeder Iteration mehrere Methoden angewendet (Mini-Wasserfall), und man beginnt sehr früh mit der Implementation.

Die Verfasser des RUP legen Wert darauf, dass ihr Modell nicht einfach statisch angewendet und übernommen wird, sondern immer auf die jeweiligen Bedingungen hin betrachtet und verändert wird.

Nur eine kleine Menge der im RUP vorgesehenen Aktivitäten und Artefakte, die echten Wert beisteuern, sollte durchgeführt bzw. erstellt werden. Im Projektverlauf sollte jede Prozesstätigkeit, die dessen Wert nicht verbessert, aufgegeben werden.

Der Rational Unified Process ist also wie eine Apotheke: man entnimmt nur die Methoden und Techniken, die für die konkrete Problemstellung sinnvoll sind – wie man aus einer Apotheke nur die Medikamente entnimmt, die für eine bestimmte Krankheit hilfreich sind.

1.3.7 EXtreme Programming (XP)

eXtreme Programming ist ein Prozessmodell für die iterative, objektorientierte Softwareentwicklung. Es wurde von KENT BECK, WARD CUNNINGHAM und RON JEFFRIES entworfen und anhand von Projekten weiterentwickelt [Bec00] (weiterführende Informationen finden Sie in der Literaturliste zu diesem Abschnitt). Keine der eingesetzten Praktiken ist wirklich neu. Neu sind die Kombination und der extreme Grad, in dem die Praktiken eingesetzt werden (vgl. [Rei00]).

eXtreme Programming ist durch **kleine Releases**, unterteilt in Iterationen und Arbeitsspakete, gekennzeichnet. Releases sollten für Zeiträume bis zu 3 Monaten geplant werden. Ein Release ist eine Zusammenstellung von fertigen Softwareeinheiten, die in ihrer Gesamtheit eine bestimmte Funktionalität abdecken. Es ist lauffähig und testbar. Die Iterationen sind nicht länger als 3 Wochen zu planen und die Arbeitsspakete sollen in bis zu 3 Tagen erledigt sein. Für die Planung des Release wird ein Planungsspiel zur Feinabstimmung der Aufgaben mit dem Kunden durchgeführt. Die Aufgaben oder Anforderungen werden in Form von **User Storys** niedergeschrieben. Diese sind Grundlage für eine Diskussion mit dem Kunden und werden so lange verändert, bis es eine Einigung gibt. Sie können sowohl durch den Kunden als auch durch den Entwickler entworfen werden und sind innerhalb einer Iteration abzuarbeiten. Für jede Iteration wird eine **Iterationsplanung** vorgenommen. Dazu wird die anstehende Story in **Tasks** (Aufgaben) zerlegt.

Ein zentraler Punkt in diesem Modell sind **Tests**. Es werden Tests für Storys geschrieben (**Functional Tests**) und Tests für Klassen (**Unit Tests**) entwickelt. Die Testfälle für Klassen werden so implementiert, dass sie jederzeit ausgeführt werden können. Tests werden vor der Implementation geschrieben. Auf eine komplette Architektur wird verzichtet. Sie wird ersetzt durch die (System)-**Metapher**, von der sich jeder Entwickler leiten lässt.

Ein weiterer wichtiger Punkt ist die **Refaktorisierung**. Sie dient zur Vereinfachung des Programmcodes. Der Vorgang soll die Verständlichkeit erhöhen, ohne das Verhalten zu verändern. Der neue Code wird dann mit den dokumentierten Tests geprüft. Es ist besonders wichtig, einfachen und leicht verständlichen Code zu erzeugen.

eXtreme Programming baut auf das Programmieren in Paaren. Es gibt keinen privaten Code. Jeder kann seinen Platz wechseln, wenn festgestellt wird, dass er für eine andere Aufgabe oder zu einem anderen Partner besser passt.

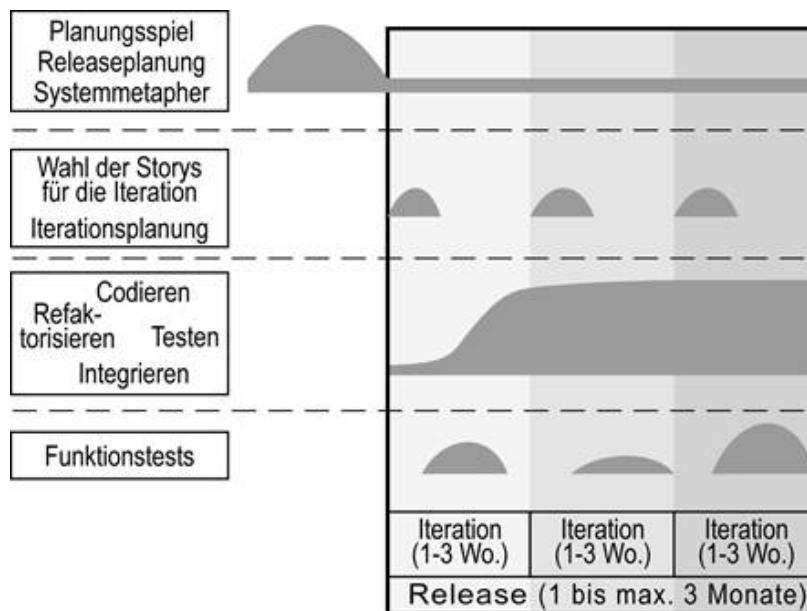


Abbildung 1.3.7.1: eXtreme Programming (XP)

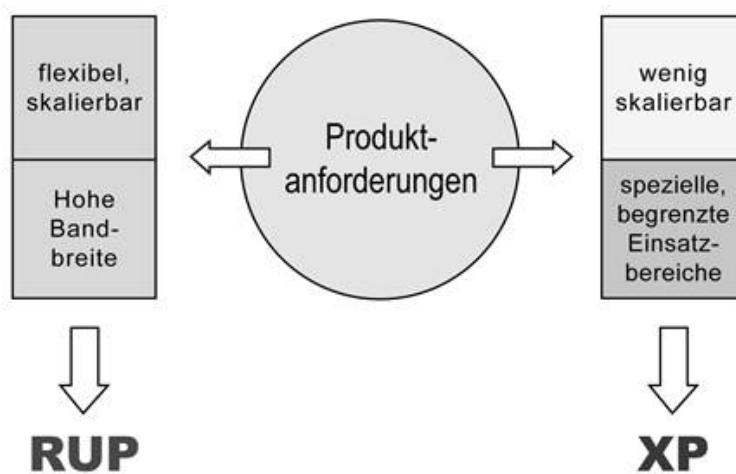
Dieses Modell verzichtet auf umfangreiche Dokumentationen. Man verlässt sich in erster Linie auf gut kommentierten Quelltext. Dazu werden eindeutige Programmierrichtlinien gebraucht. XP ist damit ein sogenanntes leichtes Prozessmodell und ein wichtiger Vertreter der **agilen** Vorgehensmodelle.

1.3.8 Schlussfolgerungen

Aus der **Sicht des Projektmanagements** sind die iterativen Modelle wesentlich schwerer zu planen und zu kontrollieren, da die Dokumente in vielen Versionen erstellt werden und damit die Schnittstellen zwischen den Arbeitspaketen sehr genau abgestimmt und kontrolliert werden müssen. Auf der anderen Seite kann die Genauigkeit der Planungen wesentlich erhöht werden, da ständig Ist-Werte für die Korrektur bereitstehen und die Planungszeiträume kürzer sind. Die Mitarbeiter müssen sehr diszipliniert und verantwortungsbewusst arbeiten, innerhalb der Entwicklungsteams muss sehr viel kommuniziert werden. Besonders geschulte Mitarbeiter, die sich voll mit dem Projekt identifizieren, erfordert das eXtreme Programming. Jeder Mitarbeiter muss an unterschiedlichen Plätzen die gesamten vorgeschriebenen Methoden beherrschen.

Aus der **Sicht der Qualität** hat die iterative Vorgehensweise viele Vorteile. Sie ermöglicht eine Anpassung an Kundenwünsche und berücksichtigt das Auftreten von Fehlern und Missverständnissen, da Dokumente wieder und wieder kontrolliert und überarbeitet werden.

Die Entscheidung zwischen dem RUP oder dem XP-Prozess kann auf Grundlage von Einsatzkriterien gefällt werden. Soll ein Produkt ein breites Aufgabenspektrum abdecken und sehr flexibel und anpassbar sein, sollte der RUP verwendet werden, da auch das Projekt komplex ist. Soll das Produkt nur spezielle, begrenzte Einsatzbereiche unterstützen, kann mit dem XP-Prozess gearbeitet werden.



| Abbildung 1.3.8.1: Einsatz RUP oder XP

Multimedia-Projekte weisen eine Reihe von Besonderheiten auf, die eine separate Behandlung erfordern. Es sind keine reinen Programmentwicklungen. Es werden HTML- und PDF-Dateien erzeugt, es entstehen Flash-Dateien und Videoformate – also Dokumente, Multimedia-Elemente und auch Quellcode. Diese unterschiedlichen Komponenten verlangen einen sehr flexiblen Entwicklungsprozess.

Die Videoproduktion erfordert eigentlich ein Wasserfallmodell. Bevor Schauspieler und Sprecher engagiert werden, müssen die Drehbücher erstellt und jeder Dialog genau ausformuliert werden. Prototyping schließt sich aus. Eine Wiederholung ist nur in seltenen Fällen möglich. Die Navigation und die Textgestaltung in HTML-Dokumenten erfordern dagegen meistens etliche Varianten, bevor man ein befriedigendes Ergebnis vorzuliegen hat. Auch das Erstellen von komplexen interaktiven Animationen erfordert die Ausarbeitung detaillierter Storyboards für den Handlungsablauf – insbesondere dann, wenn der Auftrag an einen externen Auftragnehmer geht und Kosten sowie Entwicklungszeit festzulegen sind. Auf der anderen Seite wird eine Animation auch ganz wesentlich durch das GUI geprägt, das durch Vorstellungen der späteren Anwender bestimmt wird. Diese Vorstellungen konkretisieren sich aber erst nach der Erstellung von Szenen. Ein frühes Prototyping ist daher sehr empfehlenswert.

Das Spiralmodell erlaubt die Verwendung unterschiedlicher Vorgehensmodelle für den eigentlichen Entwicklungsschritt und kommt damit Multimediaprojekten mit den unterschiedlichen Charakteren der einzelnen Medien entgegen. Es ist auf die Beachtung von Risiken ausgelegt.

Auch der Rational Unified Process (RUP) hat eine hohe Flexibilität. Sein Vorteil besteht darin, dass er ständig aktualisiert wird, aber durch seine starke Anlehnung an die UML mehr für die objektorientierte Softwareentwicklung verwendet wird. Das Spiralmodell scheint durch die vielen laufenden großen Multimediaprojekte eine Renaissance zu erleben (z.B. im E-Learning-Bereich oder für Geo-Informationssysteme), wie sich aus aktuellen Recherchen im Internet ablesen lässt.

1.3.9 Aufgaben zu Projektphasen und Prozessmodellen



Aufgabe

Aufgabe 1.3.9-1

Softwareprojekte weisen einige Besonderheiten gegenüber anderen Projekten (z.B. Bauprojekten) auf. Welche Besonderheiten sehen Sie? Nennen Sie drei!



Aufgabe

Aufgabe 1.3.9-2

Welche Rückschlüsse für die Gestaltung der Entwicklungsprozesse lassen sich ziehen?



Aufgabe

Aufgabe 1.3.9-3

Erläutern Sie kurz das eXtreme Programming (XP) und ziehen Sie Rückschlüsse für die Projektplanung bei Verwendung dieses Prozessmodells!



Aufgabe

Aufgabe 1.3.9-4

Herr Oldenburg, der Geschäftsführer einer Internet-Firma, will ein neues Entwicklungsprojekt auflegen. Er wird dazu die 6 Mitarbeiter einer Entwicklungsgruppe des Unternehmens dem Projekt zuordnen. Gruppenleiter ist Herr Markwart, der damit auch die Projektleitung übernimmt. Die Gesamtdauer des Projekts wurde auf 9 Monate festgelegt. Die Zielbeschreibung ist auf Grund des innovativen Charakters noch unscharf und soll anhand der erbrachten Ergebnisse vervollständigt werden.

Schlagen Sie ein geeignetes Prozessmodell vor!

- A) Wasserfallmodell
- B) Spiralmodell
- C) Rational Unified Process
- D) Extreme Programming

**Aufgabe**

Aufgabe 1.3.9-5

Recherchieren Sie, was man unter agiler Softwareentwicklung versteht!

2 Projektstart	
	<u>2 Projektstart</u>
	<u>2.1 Projektziele</u>
	<u>2.2 Risiken in Softwareprojekten</u>
	<u>2.3 Projektorganisation</u>

**Lernziele**

Was sollten Sie nach diesem Kapitel wissen?

- Welche Aufgaben gehören in die Projektstartphase und wie werden die Projektziele definiert?
- Wie kann die Wirtschaftlichkeit nachgewiesen werden?
- Wie wird eine Risikoanalyse aufgebaut?
- Welche Organisationsform soll gewählt werden?

2.1 Projektziele	
	<u>2.1 Projektziele</u>
	<u>2.1.1 Projektziele beschreiben</u>
	<u>2.1.2 Wirtschaftlichkeitsrechnung</u>
	<u>2.1.3 Konfliktmanagement</u>
	<u>2.1.4 Verträge</u>
	<u>2.1.5 Aufgaben zu Projektziele</u>

Die Projektstartphase ist ein ganz wichtiger Abschnitt eines Projektes. Hier werden die Ziele und das dazugehörige Budget zwischen dem Auftragnehmer und dem Auftraggeber vereinbart. Es wird damit die Grundlage für eine gewinnbringende Projektabwicklung geschaffen. Neben den Projektzielen mit den Terminzielen, Kostenzielen und den funktionellen (qualitativen) Zielen müssen dazu auch die Risiken analysiert (vgl. [Abschnitt 2.2](#)) und eine geeignete Struktur (Organisation) im Unternehmen geschaffen werden (siehe [Abschnitt 2.3](#)). Schließlich muss es einen verbindlichen Vertrag zwischen dem Auftraggeber und dem Auftragnehmer geben. Die Startphase schließt mit der Entscheidung zur Projektannahme und dem Kick-Off ab.

2.1.1 Projektziele beschreiben

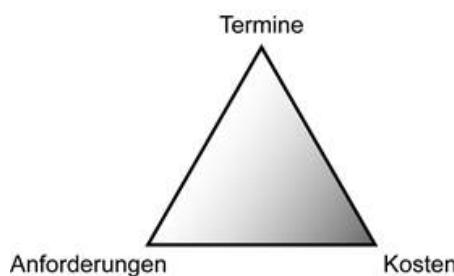
Ein Softwareprojekt beginnt wie jedes andere Projekt mit einer Vision. Auslöser solcher Visionen können schlechte Bedingungen am Arbeitsplatz, ökonomische Zwänge, Anregungen von Zeitschriften, Messen und Kongressen sein oder auch einfach aus der Beobachtung der Konkurrenz entstehen. Mit Hilfe von Softwareprojekten versuchen also Unternehmen, ihre Produktivität zu verbessern und ihre Marktstellung zu behaupten oder auszubauen. Die in der Vision erzeugten Wünsche müssen in konkrete Ziele eines aufzulegenden Projekts umgearbeitet werden. Diese müssen in einer Projektzielbeschreibung zusammengefasst werden. Die Zielbeschreibung enthält die Projektidee (Nutzen) sowie eine Lösungsidee und ist Voraussetzung für die Spezifikation der Softwareanforderungen nach der Entscheidung für die Durchführung des Projekts. Für diese Entscheidung werden mit der Projektzielbeschreibung die Grundlagen geschaffen. Darüber hinaus muss eine Einschätzung der Grundparameter des Projekts vorgenommen werden:



Wichtig

- die Ecktermine,
- das Budget (Kosten, Einsatzmittel) und
- die Softwareanforderungen.

Diese Grundparameter stehen in einer gegenseitigen Wechselwirkung, im Projektmanagement auch unter dem Begriff "Magisches Dreieck" bekannt.



■ Abbildung 2.1.1.1: Das magische Dreieck des Projektmanagements

Will man beispielsweise in einem Softwareprojekt die Zuverlässigkeit des Produkts erhöhen – also die Qualität verbessern – werden zusätzliche Tests erforderlich. Diese Tests erfordern Zeit und damit verschiebt sich der Fertigstellungstermin des Projekts. Da Personal länger beschäftigt werden muss, erhöhen sich auch die Kosten.

Eine Schätzung der Termine und Kosten zu einem so frühen Zeitpunkt ist schwer. Bei vielen Softwareprojekten steht der Fertigstellungstermin im Vordergrund – "Erster am Markt" ist aus Marketinggründen wichtig. Ausgangspunkt für jede Termin- oder Kos-

tenplanung ist der zu realisierende Funktionsumfang (qualitative Projektziele), der seinen Ausdruck in einer groben Aufgaben- oder Projektstruktur findet. Die Arbeitspakete erfordern wiederum einen bestimmten Aufwand (Arbeitsstunden oder Mann-Stunden). Mit dem Aufwand lassen sich unter Berücksichtigung des einsetzbaren Personals die Termine und auch Kosten schätzen. Will man die Projektzeit verkürzen, um dem Kriterium "Erster am Markt" zu entsprechen, müssen qualitative Abstriche gemacht werden oder es muss mehr Personal eingesetzt werden. Die Problematik der Aufwandsschätzung sowie der Termin- und Kostenplanung wird im Kapitel "Projektplanung", besonders aber im Abschnitt "Planungstechniken" behandelt. Alle Termin- und Kostenschätzungen sind mit großen Unsicherheiten behaftet. Sie beruhen auf Erfahrungen und dem Wissen über die Leistungsfähigkeit der Mitarbeiter (falls Sie diese kennen). Sorgen Sie also für ausreichend Spielraum bei den qualitativen Projektzielen.

Softwareanforderungen

Wie die Ausführungen im Abschnitt Projektphasen und Prozessmodelle gezeigt haben, ist es unmöglich, die Anforderungen vollständig am Anfang eines Projektes zu bestimmen. Die Präzisierung und Ermittlung detaillierter Anforderungen ist eine prozessbegleitende Aufgabe, die in den Softwareentwicklungsmodellen einen festen Platz hat (Requirements Engineering). Im Hinblick auf die Vertragsgestaltung steckt man in einem echten Dilemma. Konflikte sind sehr oft vorprogrammiert, denn wenn es keine detaillierten Beschreibungen der Anforderungen gibt, kann je nach Standpunkt mehr oder weniger in ein Projekt hineininterpretiert werden. Diese Tatsache ist einer der Hauptgründe, weshalb das Wasserfallmodell so zählebig ist. In der Praxis hat sich gezeigt, dass inkrementelle, iterative Modelle nur mit Kunden funktionieren, zu denen ein vertrauenswürdiges Verhältnis besteht.

Ein weiterer Aspekt muss unbedingt beachtet werden: Die qualitativen Projektziele beinhalten die funktionalen Anforderungen, aber auch die sogenannten Attribute. Attribute sind Leistungsanforderungen (Verarbeitungszeiten, Datenmenge), besondere Qualitäten (Benutzerfreundlichkeit) und Randbedingungen (Normen und Gesetze). Gerade diese Attribute können ein Projekt verteuern.

Die Anforderungen aus Anwendersicht (auch Anwenderanforderungen im Gegensatz zu den Systemanforderungen, die daraus abgeleitet werden) können je nach der Art des Projektes unterschiedlich ermittelt werden. Bei **Auftragsprojekten** bieten sich **Gespräche** und **Interviews** mit dem Kunden oder Auftraggeber an. Die Gespräche mit dem Kunden müssen gut vorbereitet werden. Sie sollten dabei mit folgenden Problemen rechnen:



Wichtig

- Der Kunde weiß nicht genau, was er will.
- Der Kunde weiß nicht genau, was möglich ist.
- Der Kunde verlangt das Unmögliche.
- Der Kunde ist betriebsblind (er macht Dinge, deren Zusammenhänge er nicht kennt).
- Der Kunde kennt keine Analysetechniken.
- Der Kunde kennt keine Darstellungstechniken.

Da Missverständnisse und daraus abgeleitete falsche Anforderungen eine häufige Ursache für das Scheitern von Softwareprojekten darstellen, sollten Sie diese Probleme sehr ernst nehmen.

Auch **interne Projekte** oder **Organisationsprojekte** erfordern **Gespräche** und **Interviews** mit den Vorgesetzten und Mitarbeitern aus den betroffenen Abteilungen.

Die in beiden Fällen durchzuführenden Interviews müssen am Arbeitsort des Befragten stattfinden. Es sollten möglichst nicht mehr als insgesamt 2 Personen teilnehmen, und beide sollten gut auf das Interview vorbereitet sein. Zu dieser Vorbereitung gehört die Ausarbeitung von Fragen.

Wichtig

Stellen Sie Ihre Fragen

- kurz,
- redundanzfrei,
- offen,
- konkret,
- nicht suggestiv und

verwenden Sie keine Unterfragen. Dokumentieren Sie die Interviews möglichst:

- mit kurzen, schriftlichen Notizen (Lücken im Fragenkatalog),
- verbal und bildlich,
- ohne Tonband oder ähnlichem.

Die Protokollierung erfolgt mit vorbereiteten Formularen.

Entwicklungsprojekte erfordern eine **klassische Marktanalyse** und **Bedarfsanalyse** für ein Produkt. Dazu können Instrumentarien wie Portfolio-Darstellungen verwendet werden (z.B. Geschäftsfeldportfolio). Das gehört aber in den Bereich des Marketings und wird hier nicht weiter verfolgt.

Ziele haben nur dann Bestand, wenn Sie auch machbar sind. Jede Projektzielbeschreibung sollte deshalb auch eine **Machbarkeitsstudie** enthalten. Die Machbarkeit der qualitativen Ziele kann folgendermaßen bestimmt werden:



Wichtig

- Recherchen und Analysen gleich gearteter Projekte.
- Entwurf einer technischen Lösung durch die Formulierung einer **Metapher**.
- Untersuchung der Möglichkeiten von Grundtechniken (z.B. XML, XSL, Java).
- Gegenüberstellung der Vor- und Nachteile von Lösungen.

In der Machbarkeitsstudie werden also die aufgestellten Anforderungen analysiert. Die Analyse der Anforderungen sollte von Mitarbeitern vorgenommen werden, die später auch für die Kodierung verantwortlich sind. Gerade diese Mitarbeiter haben ein hohes Interesse an machbaren Anforderungen und verfügen über das notwendige Wissen dies einzuschätzen. Es gibt eine Reihe von sehr brauchbaren Techniken, die durch die UML beschrieben werden.

Die Ausarbeitung von Lösungsansätzen ist auch deshalb so wichtig, weil Anforderungen meistens hierarchisch aufgebaut sind. Die spezielle Lösung (aus einer Reihe von Lösungsvarianten) einer übergeordneten Anforderung erzeugt dann abhängige Anforderungen.

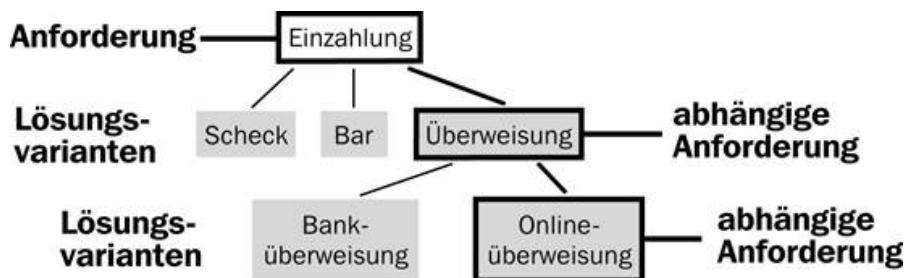


Abbildung 2.1.1.2: Abhängigkeit der Anforderungen von den Lösungen

Die Anforderungen müssen in einem Dokument zusammengefasst werden, das als Vertragsgrundlage für Leistungen, Termine und Budget zwischen Auftraggeber und Auftragnehmer dient. Es gibt verschiedene Begriffe für dieses Dokument:

- Das Institute of Electrical and Electronic Engineers (IEEE) spricht von der **Software Requirements Specification**.
- Die ISO 9001, eine internationale Qualitätsnorm, nennt dieses Dokument in seiner deutschsprachigen Fassung einfach **Anforderungen**.
- Im deutschen Sprachraum weit verbreitet ist der Ausdruck **Pflichtenheft** bzw. **Lastenheft**.



1. Einleitung

1.1 Anlass

1.2 Ziele

1.3 Einsatzbereich

2. Überblick

2.1 Kontext

2.2 Struktur der Problemstellung

2.3 Globale Attribute

2.4 Annahmen

2.5 Perspektiven

3. Einzelanforderungen

[hierarchisch gegliederte Auflistung der Einzelanforderungen]

4. Anhänge

▣ Abbildung 2.1.1.3: Gliederung der Software Requirements Specification nach der IEEE 830-1993

Um Missverständnisse bei der Dokumentation auszuschließen, muss eine präzise, klare und deutliche Formulierung gewählt werden. Anforderungen ändern sich. Das Anforderungsmanagement ist eine wichtige Aufgabe im Projekt.

Auf der Auftraggeberseite spielt die Wirtschaftlichkeit eine große Rolle. Jedes Projekt stellt eine Investition dar, die sich für den Kunden lohnen soll. Das trifft für alle Arten von Projekten zu, also sowohl für Auftragsprojekte als auch für interne und Entwicklungsprojekte – ein Unternehmen muss Gewinn erwirtschaften. Mit der dazu nötigen Wirtschaftlichkeitsrechnung beschäftigt sich der nächste Abschnitt.

2.1.2 Wirtschaftlichkeitsrechnung

Projektverantwortliche beim Auftraggeber (IT-Verantwortliche) müssen Projekte auch finanziell durchsetzen. In der Argumentation muss der **Return on Investment (ROI)** betont werden. Der Projektleiter beim Auftragnehmer leistet seinen Beitrag durch die Schätzung der Projektkosten. Letztendlich entscheidet sich ein Projekt danach, ob der Finanzrückfluss, z.B. durch Einsparungen, zu den Ausgaben in einem positiven Verhältnis steht.

Die folgenden Zitate sind einem Artikel der Computer Zeitung 08/2001 entnommen und charakterisieren die harten ökonomischen Zwänge an neue Softwareprojekte [CoZ01]:

"Für IT-Verantwortliche vollzieht sich derzeit ein Paradigmenwechsel: Statt den neuesten Trends hinterherzulaufen, heißt es für sie jetzt, dem Management knallharte **Kennzahlen** über Kosten und Nutzen von IT-Projekten darzulegen. 'Die Periode, in der Unternehmen Investitionen getätigt haben, um bloß nicht hinter der Konkurrenz zurückzubleiben, ist definitiv vorbei (...). Um das Management oder die Geschäftsführung von den finanziellen Vorteilen eines IT-Vorhabens zu überzeugen, lassen sich einfache Berechnungsmethoden wie **Amortisation** oder der **Return on Investment** heranziehen. Dabei handelt es sich um den Vergleich des Finanzrückflusses mit den Ausgaben. Er drückt sich daher in Prozent aus: Wer das Zehnfache an Geld herausholt, als er hineingesteckt hat, erzielt einen ROI von 900 Prozent."

Weiter wird darauf hingewiesen, dass neben den Kosten für Hard- und Software, Infrastrukturaufbau sowie Testen, Installation, Validierung und Wartung noch andere Kosten entstehen – nämlich für die Unterbrechung der Geschäftsprozesse bei der Umstellung (oder beim Parallelbetrieb) der Systeme, für Training, Change-Management-Maßnahmen oder den potenziellen Verlust von Mitarbeitern.

"Auf der Seite der Benefits sollten die direkten Kosteneinsparungen im Vordergrund stehen. Hier gilt es gezielt darzulegen und auf einzelne Bereiche herunterzubrechen, wo die Ersparnisse auftauchen, beispielsweise in Form von einer gesteigerten Produktivität: Wenn pro Mitarbeiter ein größerer Output erzielt werden kann, kann man etwa auf Neueinstellungen verzichten" [CoZ01].

Die folgende Aufzählung nennt drei mögliche einfache **Verfahren der statischen Investitionsrechnung**:



- Kostenvergleichsrechnung,
- Gewinnvergleichsrechnung,
- Amortisationsrechnung.

Ein Beispiel, in dem aufgezeigt wird, wie man mit Hilfe dieser Berechnungsverfahren aus ökonomischer Sicht zu einer Projektentscheidung (Einführung der neuen Technologie) gelangt, finden Sie im Anhang 7.1.

Neben den berechenbaren Größen werden oft nicht quantifizierbare Größen für die Beschreibung des Nutzens von Softwareprojekten herangezogen. Das sind z.B. solche Größen wie "Verbesserung der Arbeitsbedingungen", "Erhöhung der Transparenz",

usw. Um solche qualitativen Größen einer mathematischen Betrachtung zuzuführen, kann die **Nutzwertanalyse** verwendet werden. Man schreibt alle Größen auf, die im Hinblick auf das Projekt relevant erscheinen. Falls es Unterschiede in der Wertigkeit zwischen den Größen gibt, kann dieser Unterschied durch die Einführung eines Wichtungsfaktors berücksichtigt werden. Der Nutzwert der einzelnen Faktoren kann mit Punkten bewertet werden. Dazu wird für jeden Faktor eine kurze Beschreibung hinterlegt, unter welchen Umständen der Nutzwert sehr gut (5 Punkte) oder eben sehr schlecht (1 Punkt) einzuschätzen ist. In der Summe aller Faktoren sollte dann ein positives Ergebnis erzielt werden. Ein Beispiel für eine Nutzwertanalyse finden Sie im Anhang 7.2.

Die Bestimmung der Ziele beinhaltet eine Menge Streitpotenzial zwischen den Interessengruppen – Kunden, Auftragnehmer und Entwickler. Wie diese Konflikte gelöst werden können, damit beschäftigt sich der nächste Abschnitt.

2.1.3 Konfliktmanagement

Gerade die Zielbestimmung erfordert die enge Zusammenarbeit der beteiligten Gruppen – Auftraggeber, Benutzer, Entwickler und Auftragnehmer. Dabei stoßen ganz unterschiedliche Interessen aufeinander. Der Auftraggeber möchte für wenig Geld eine komfortable Software. Der Auftragnehmer möchte möglichst viel Gewinn mit dem Projekt realisieren. Der Benutzer möchte ein leicht bedienbares Programm. Die Entwickler schließlich legen Wert darauf, so zu programmieren, dass sie Teile auch in anderen Projekten verwenden können. BARRY BOEHM [Boe96] hat **Qualitätskriterien** aus diesen unterschiedlichen Sichten zusammengefasst.

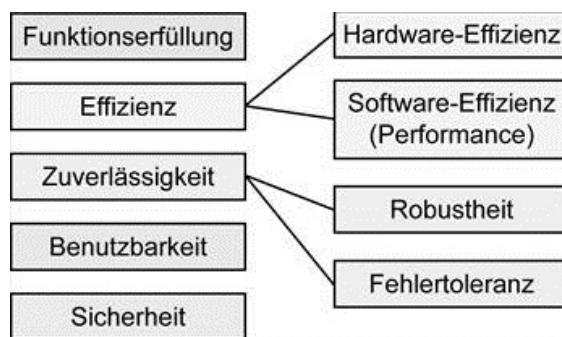


Abbildung 2.1.3.1: Qualitätsmerkmale für Software aus Benutzersicht nach BARRY BOEHM

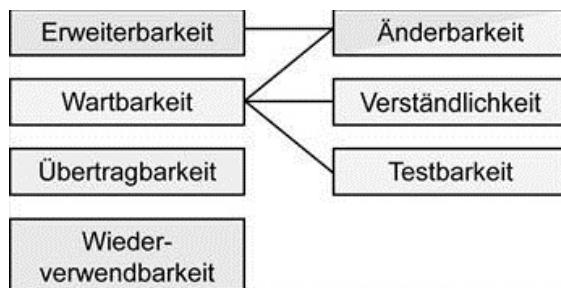


Abbildung 2.1.3.2: Qualitätsmerkmale aus Entwicklersicht nach BARRY BOEHM

Die Projektziele finden sich in den Vertragsunterlagen zwischen dem Auftraggeber und dem Auftragnehmer z.B. in der Requirements Specification wieder. Die **Projektzielbeschreibung** ist deshalb **Verhandlungsgegenstand** zwischen diesen beiden Parteien. Wie oben beschrieben, verfolgen sie unterschiedliche Ziele. Konflikte werden in der Startphase von Projekten besonders zwischen diesen Gruppen auftreten. In der Praxis findet man aber auch oft Meinungsverschiedenheiten zwischen dem Projektleiter und dem eigenen Management. Das Management möchte den Auftrag und neigt dazu, dem Auftraggeber entgegenzukommen. Natürlich soll der Gewinn möglichst hoch bleiben. Der Projektleiter wird unter Druck gesetzt, seine Schätzungen zu überarbeiten und mehr zu leisten.

Zwischen den Konfliktpartnern muss verhandelt werden. Dabei ist zu beachten, dass man nie jemanden gegen seinen Willen überzeugen kann. Man kann:



- latente Überzeugungsbereitschaft wecken
- gegen Überzeugungen zu etwas überreden
- aus einer Machtposition heraus erpressen
- Mitleid erregend um etwas bitten
- Geschäft/Tausch anbieten [*Kel01*]

Welche dieser Methoden Sie anwenden, hängt von der Position Ihres Gesprächspartners ab. Natürlich auch von den eigenen Ambitionen: Was bedeutet dieses Projekt für mich persönlich? Öffnet mir dessen Erfolg neue Tätigkeitsbereiche und Chancen im Unternehmen oder sichere ich damit meinen Arbeitsplatz?

Oft kommt es also darauf an, geschickt zu verhandeln. **Verhandlungen** sollten gut vorbereitet werden:



- meine Ziele:
 - strategische Ziele

- Gesprächsziele
- Rückzugsziele
- meine Argumente:
 - Sachargumente
 - „Macht-Argumente“
 - Nutzargumente
- meine Strategie:
 - vor der Verhandlung
 - Phasen der Verhandlung
 - Taktik
 - Psychologie berücksichtigen [*Kel01*]

Der letzte Punkt spricht den Umgang mit Ihrem Verhandlungspartner an: Eine erfolgreiche Verhandlung wird vielfach auch davon abhängen, wie Sie sich in die Situation Ihres Verhandlungspartners hinein versetzen können. Beachten Sie die folgenden Punkte:



Wichtig

- Was sind "Überzeugungshilfen" für die Gegenseite?
- Wie wahrt die Gegenseite ihr "Gesicht"?
- Was bedeuten Sieg oder Niederlage für den anderen?

In vielen Fällen können Konflikte schon vermieden werden, indem Sie Risiken für mögliche Konflikte früh erkennen und mindern.



Wichtig

- Vereinbaren Sie die Ziele, Teilziele, Abnahmen und Ressourcen schriftlich.
- Planen Sie Puffer ein und veröffentlichen Sie diese nicht.
- Binden Sie die Betroffenen immer früh ein und vermitteln Sie dadurch, dass deren Argumente und Wünsche berücksichtigt werden.
- Klären Sie Ihre Kompetenzen, die sich aus Ihrer Stellung und Ihrem Sachverstand ergeben.
- Notfalls müssen Sie konsequent sein und Verhandlungen abbrechen.

So bitter sich der letzte Punkt anhört – ist es doch besser, ein Projekt abzulehnen, als mit der Überzeugung zu starten: "Es ist nicht zu schaffen, aber es wird sich schon richten". Mit Projekten betritt man immer Neuland, und bei aller Erfahrung, die Sie für die Vorausplanung eingesetzt haben, werden sich doch viele unerwartete Probleme einstellen, die eine "geplante" Überlastung einfach nicht zulassen.

2.1.4 Verträge

Es muss einen Projektvertrag geben, der die Pflichten zwischen dem Kunden und dem Auftragnehmer regelt. Der Kunde verpflichtet sich beispielsweise, fällige Zahlungen in einer bestimmten Höhe zu leisten. Der Auftragnehmer verpflichtet sich, festgelegte Leistungen pünktlich zu erbringen. In Softwareprojekten sollte immer ein Punkt enthalten sein, der bei Nichteinhaltung einer bestimmten Verpflichtung oder bei gewünschten Änderungen von Leistungen ein Verfahren zur Konfliktbewältigung vorschreibt. Ein Beispiel für einen Projektvertrag finden Sie im [Anhang 7.3](#).

Ein internes Dokument ist der sogenannte **Projektantrag**. Er enthält die Zusammenfassung der oben beschriebenen Elemente der Projektzielbestimmung. Letztendlich handelt es sich hierbei um ein Formular, das auf einen Blick wichtige Kennwerte wie Kosten und Termine präsentiert:



Aufgabe

- wichtige Projektziele
- Aufgabenbeschreibung
- grober Kostenplan
- grober Terminplan
- Verantwortlichkeiten
- Konfliktlösungsstrategie

Der Projektantrag wird für die Entscheidungsfindung beim Auftragnehmer selbst verwendet und später in das Projektcontrolling überführt. Wird der Projektantrag vom Projektleiter und dem übergeordneten Management (interner Auftraggeber) unterzeichnet, so wird daraus ein **Projektauftrag** und damit ein wichtiger Vertrag für den Projektleiter. Im [Anhang 7.4](#) und [Anhang 7.5](#) finden Sie Formulare, die Sie für Ihre Arbeit verwenden können.

Wichtige Dokumente bilden auch die notwendigen **Arbeitsverträge** mit den Mitarbeitern. Da der Erfolg von Softwareprojekten sehr stark vom Personal abhängt, ist die Gestaltung der Arbeitsverträge besonders wichtig und hier wiederum die Gestaltung der

Arbeitszeiten und der Bezahlung, da diese Punkte die Motivation beeinflussen (vgl. Ab schnitt 6.1 "Motivationstheorien").

2.1.5 Aufgaben zu Projektziele

**Aufgabe**

■ Aufgabe 2.1.5-1

Die folgenden Punkte gehören in eine Projektzielbeschreibung. Wie können die notwendigen Aussagen ermittelt werden ...

- A) ... zur Beschreibung der Aufgabe und Einschätzung ihrer Machbarkeit?
- B) ... zur Festlegung von Terminen?
- C) ... zur Einschätzung des Budgets?

**Aufgabe**

■ Aufgabe 2.1.5-2

Was versteht man unter dem "Magischen Dreieck" im Projektmanagement? Nennen Sie die drei zugehörigen Begriffe!

**Aufgabe**

■ Aufgabe 2.1.5-3

Warum ist dieses Dreieck "magisch"?

**Aufgabe**

■ Aufgabe 2.1.5-4

Welche Möglichkeiten haben Sie, um nicht quantifizierbare Nutzensgrößen eines Softwareprojekts (z.B. leichtere Entscheidungsfindung) einer mathematischen Auswertung zuzuführen?

- A) Gewinnvergleichsrechnung
- B) Nutzwertanalyse
- C) Amortisationsrechnung

**Aufgabe**

■ Aufgabe 2.1.5-5

Welcher Vertrag wird zur internen Entscheidungsfindung im Projekt verwendet?

- A) Requirements Specification

- B) Arbeitsvertrag
- C) Projektauftrag



Aufgabe

Aufgabe 2.1.5-6

Ordnen Sie die folgenden Qualitätskriterien von Software jeweils der Benutzersicht bzw. Entwicklersicht zu!

Qualitätsmerkmal	Entwicklersicht	Benutzersicht
Wiederverwendbarkeit	?	?
Funktionserfüllung	?	?
Erweiterbarkeit	?	?
Sicherheit	?	?
Benutzbarkeit	?	?
Wartbarkeit	?	?

2.2 Risiken in Softwareprojekten



Gliederung

2.2 Risiken in Softwareprojekten

2.2.1 Risikoanalyse

2.2.2 Risikomanagement

2.2.3 Aufgaben zu Risiken in Softwareprojekten

Jedes Projekt ist mit Risiken verbunden, sonst hätte es die Bezeichnung Projekt nicht verdient. In diesem Abschnitt wird auf **Risikoanalyse** und **Risikomanagement** eingegangen. Diese Methoden dienen zur Absicherung und Schadensminimierung im Projekt.

2.2.1 Risikoanalyse

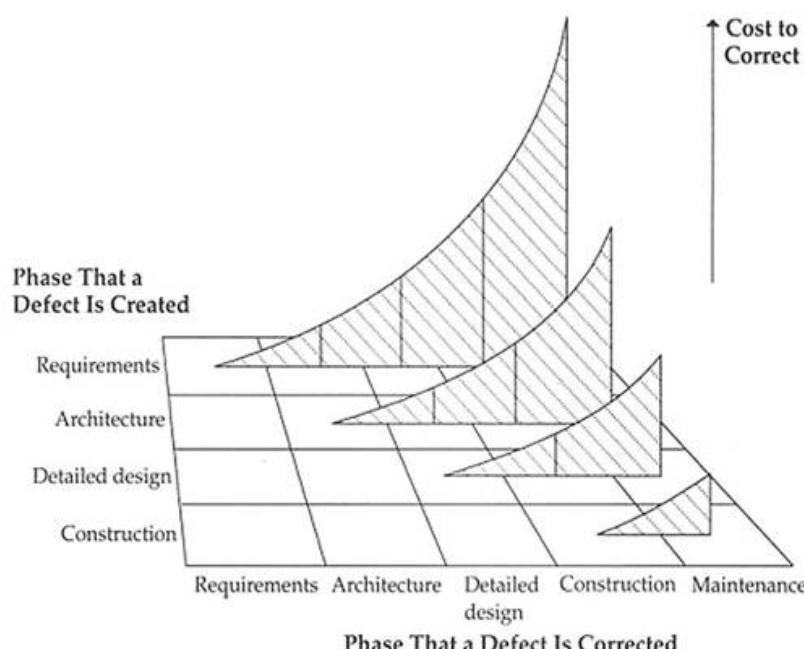
Die Risikoanalyse ist eine der vordringlichsten Aufgaben des Projektmanagements und gehört in jede solide Projektausarbeitung. Sie ist eine wesentliche Voraussetzung für den Abschluss von Verträgen. Der Projektleiter kann auf die mit dem Projekt verbun-

denen Risiken eingehen. Für die Risikoanalyse müssen die auftretenden **Risikotypen** näher untersucht werden. DEMARCO nennt in seinem Buch Bärentango [DeM03] fünf Kernrisiken:



- Fehlerhafter Zeitplan
- Inflation der Anforderungen
- Mitarbeiterfluktuation
- Spezifikationskollaps (Scheitern des Projekts durch das Scheitern des Verhandlungsprozesses zu den Anforderungen)
- Mangelnde Arbeitsleistung

Ein Hauptrisiko liegt bei den **Anforderungen**. Anforderungen werden oft textlich dokumentiert und sind deshalb interpretierbar und unvollständig. Häufig treten Missverständnisse mit dem Kunden auf (vgl. [Abschnitt 2.1 Projektziele](#)). Im schlimmsten Fall, der gar nicht so selten vorkommt, sind sie undokumentiert. Ein großes Problem stellt das **Änderungsmanagement** (Change Management) und die Wartung von Abhängigkeiten dar. Oft fehlt die nötige Koordination bzw. die vollständige Überarbeitung der Dokumente. Die Folge sind sehr hohe Kosten, die durch fehlerhafte Anforderungen verursacht werden. Man schätzt, dass bis zu 90% der Kosten, die aus der Fehlerbeseitigung resultieren, aus der Korrektur von Anforderungen erwachsen.



■ Abbildung 2.2.1.1: Verteilung der Kosten für die Fehlerkorrektur
Quelle: [McC98]

Abbildung 2.2.1.1 verdeutlicht, dass für die Höhe der Kosten der Zeitpunkt der Korrektur entscheidend ist. Eine falsche Anforderung, die sofort korrigiert wird, verursacht wenig Kosten. Wird diese mit in die Konstruktionsphase geschleppt, entstehen natürlich wesentlich höhere Kosten, da mehr sinnlose Arbeit vergeudet wurde und eventuell Änderungen in weiteren Komponenten erforderlich sind.

Als Projektleiter sollten Sie deshalb dem **Anforderungsmanagement** Ihre besondere Aufmerksamkeit schenken. Die UML (Unified Modelling Language) stellt zu diesem Zweck eine sehr hilfreiche Technik zur Verfügung – die Anwendungsfallanalyse (**Use Case Analysis**).

Ein **Anwendungsfall** (Use Case) ist ein typisches Zusammenspiel vom Benutzer mit dem System, um ein bestimmtes Ziel zu erreichen. Anwendungsfälle können sich in ihrer Größe unterscheiden. Sie bieten die Basis für die Kommunikation mit dem Auftraggeber bei der weiteren Planung des Projekts. Eine der wichtigsten Aufgaben in der Startphase besteht darin, alle möglichen Anwendungsfälle zu entdecken. Die Anwendungsfälle müssen in dieser Phase noch nicht in jeder Einzelheit beschrieben werden. Die Beschreibung sollte genau genug sein, damit die Benutzer die grundlegende Idee verstehen und die Entwickler ein grobes Verständnis dafür erhalten, was sich dahinter verbirgt.

Anwendungsfälle werden textlich und graphisch beschrieben. Die graphische Beschreibung verdeutlicht die Beziehungen zwischen Anwendungsfällen und visualisiert auch den Einfluss der Umgebung. Für die textliche Beschreibung können Schablonen verwendet werden [*Bal00*] [*Fow98*]. Ein Beispiel finden Sie im Anhang 7.6.

Falls iterative Prozessmodelle verwendet werden, wird der oben genannte Umstand berücksichtigt, dass die Anforderungen nicht vollständig und fehlerfrei erfasst werden können. So gilt für den RUP (vgl. Abschnitt Phasen und Prozesse), dass nur 10% der Anforderungen in der Inception-Phase zum Projektanfang vorliegen sollten.

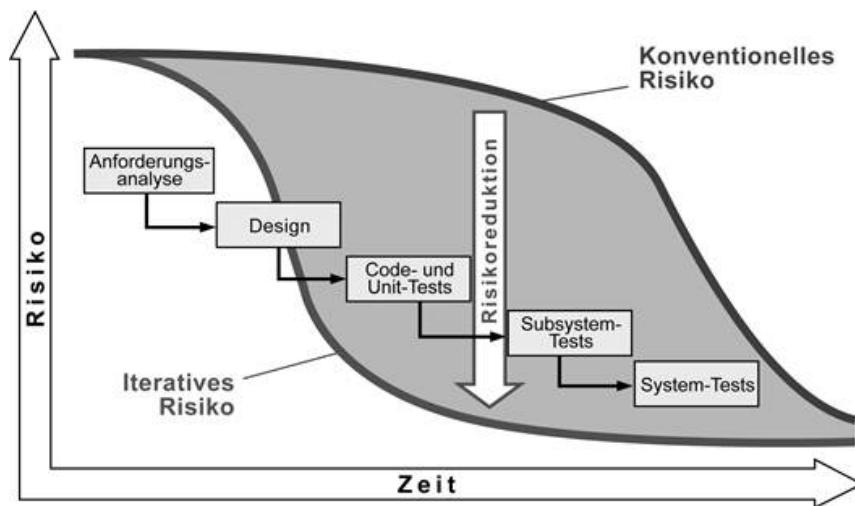


Abbildung 2.2.1.2: Risikoverteilung in Abhängigkeit vom Prozessmodell

Wie aus der Abbildung 2.2.1.2 ersichtlich ist, kann durch ein iteratives Prozessmodell das Risiko wesentlich vermindert werden. Der Kunde sieht schon sehr frühzeitig Ergebnisse. Fehlerhafte Anforderungen werden deshalb schon sehr früh erkannt und die Kosten für ihre Korrektur niedrig gehalten, man lernt, den Partner besser zu verstehen.

Solche Prozessmodelle sind auch gut geeignet, um personelle Risiken schnell in den Griff zu bekommen, denn Sie lernen Ihre Mitarbeiter sehr schnell kennen. Die Mitarbeiter sind angehalten, eine Aufgabe zügig und anforderungsgerecht zu einem Ergebnis zu führen. Da sich viele Projektprobleme auf Mitarbeiterprobleme zurückführen lassen, sollte die Arbeit mit dem Team für den Projektleiter ein Hauptbetätigungsfeld sein. Versuchen Sie, frühzeitig Konflikte zu vermeiden und Fehlverhalten zu beeinflussen.

Erstellen Sie Prototypen, um die technologischen Risiken zu minimieren. Vergessen Sie nicht, dass die größten technologischen Risiken sich daraus ergeben, wie Komponenten eines Entwurfs verknüpft sind und nicht daraus, wie die Komponenten selbst beschaffen sind. Sie kennen Java und Sie mögen auch mit XML Erfahrungen haben – die Verbindung von beiden Komponenten muss deshalb längst nicht trivial sein. Wählen Sie für den Prototyp einen einfachen Teil, der aber gerade die Verbindung der Komponenten dokumentiert. Kommentieren Sie den Quelltext gut aus und machen Sie dieses Beispiel dann allen Mitarbeitern bekannt.

Organisieren Sie Schulungen für Ihre Mitarbeiter, um die personellen Risiken zu minimieren. Das größte Manko bei den Mitarbeitern liegt heute immer noch im Einsatz der "Objektorientierung". Alle beherrschen eine objektorientierte Programmiersprache, aber es gibt nur wenige, die einen objektorientierten Entwurf hinbekommen. Entwickeln Sie ein flexibles Schulungsprogramm am Projekt selbst. Wenn ein Mitarbeiter einen

guten Entwurf erstellt hat, soll er ihn vorstellen. Schulen Sie Entwurfsmuster nach GAMMA [*Gam95*].

2.2.2 Risikomanagement

Risikozentriertes Vorgehen bedeutet, dass man sich der Risiken bewusst ist, diese analysiert und auf das eigene Vorgehen bezieht. Des Weiteren sollte man eine eigene Strategie für die Bewältigung der Risiken erarbeiten.

Risikomanagement wird meist in folgende Teilbereiche gegliedert:



Risikobewertung

- Risiken identifizieren
- Risiken analysieren
- Prioritäten festsetzen

Kontrolle von Risiken

- Indikatoren zum frühzeitigen Erkennen des Eintretens (Materialisieren) eines Risikos festlegen
- Maßnahmen zur Minimierung der Folgen eines materialisierten Risikos festlegen
- Indikatoren überwachen
- Gegebenenfalls Maßnahmen einleiten

2.2.3 Aufgaben zu Risiken in Softwareprojekten



Aufgabe 2.2.3-1

Nennen Sie 2 technologische Risiken eines IT-Projekts und beschreiben Sie geeignete Maßnahmen, um sie zu minimieren!



Aufgabe 2.2.3-2

Charakterisieren Sie die Use Case Analysis! Ist sie ...

- A) ... ein prototypischer Entwurf der Benutzeroberfläche?
- B) ... eine Analysemethode zur Spezifikation der Anforderungen?

**Aufgabe****Aufgabe 2.2.3-3**

Beschreibt ein Use Case ...

- A) ... das Zusammenspiel zwischen Benutzer und System
- B) ... zwischen Klassen des Systems
- C) ... zwischen Komponenten des Systems

**Aufgabe****Aufgabe 2.2.3-4**

Wo sehen Sie ein großes Risiko?

- A) Implementation einer Java Klasse ohne Zugriff auf andere Komponenten (DBs oder XML-Dokumente)
- B) Entwurf eines XML-Dokuments
- C) Implementation einer Java-Klasse zur Veränderung eines XML-Dokuments

**Aufgabe****Aufgabe 2.2.3-5**

Ziehen Sie aus der Antwort zu Aufgabe 2.2.3-4 eine allgemeine Schlussfolgerung!

**Aufgabe****Aufgabe 2.2.3-6**

Schätzen Sie die folgende Aussage als falsch oder richtig ein: "Risiken sollten möglichst nicht untersucht werden. Erkannte Risiken sollten ganz zum Schluss berücksichtigt werden."

**Gliederung**

2.3 Projektorganisation

2.3 Projektorganisation

2.3.1 Motivation der Projektorganisation

2.3.2 Organisationsstrukturen

2.3.3 Projektgremien

2.3.4 Aufgaben zur Projektorganisation

Heute existieren zahlreiche Formen der Projektorganisation. Dieser Abschnitt stellt die wichtigsten Alternativen zur Einbettung eines Projekts in die dauerhafte Organisationsstruktur eines Unternehmens vor.

2.3.1 Motivation der Projektorganisation

Unternehmen besitzen für die Durchführung ihrer Daueraufgaben Führungsstrukturen und Regelungen, sogenannte **Ablauf- und Aufbauorganisationen**.

Das Projektmanagement muss erfolgreich in die vorhandenen Organisationsstrukturen eingearbeitet werden. Da sich Projekte von den Daueraufgaben abgrenzen (zeitliche Begrenzung, einmaliger Inhalt usw.), muss es auch besondere Organisationsformen für ein Projekt geben. Die Bildung von Projektorganisationen trägt dem Rechnung.

In der DIN 69901 wird **Projektorganisation** folgendermaßen definiert:



Gesamtheit der Organisationseinheiten und der aufbau- und ablauforganisatorischen Regelungen zur Abwicklung eines Projekts.

Das erfordert, dass alle beteiligten Stellen in einen temporären Organisationsplan eingebunden sind. Es werden geeignete Strukturen und Gremien geschaffen, die für die Projektaufgabe zweckmäßig sind. Besondere Beachtung müssen die vorhandenen Strukturen finden, damit Konflikte innerhalb des Unternehmens möglichst vermieden werden.

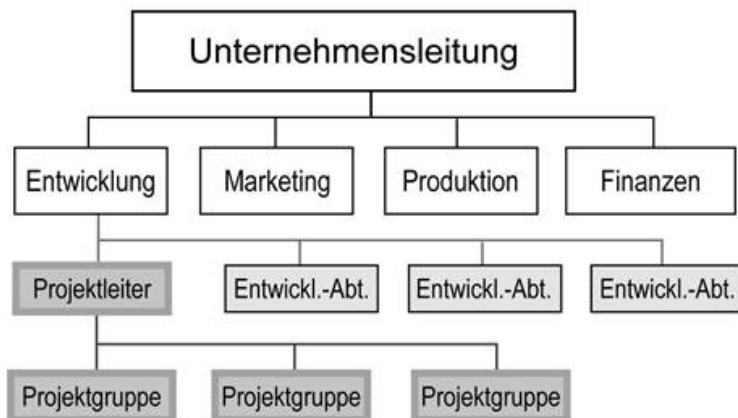
Der Charakter von Projekten erfordert eine Organisation mit einem ausgeglichenen Verhältnis von Stabilität und Flexibilität. Zuständigkeiten, Vollmachten und Verantwortlichkeiten müssen eindeutig geregelt sein. Ein Zuviel an Organisation bewirkt eine starre und träge Entscheidungsfindung, aber ohne Organisation lässt sich die Zusammenarbeit mehrerer am Team beteiligter Personen nicht realisieren. Man unterscheidet aufbauorganisatorisch zwischen der internen und externen Projektorganisation: **externe Projektorganisation** ist die Einbindung in die Firmenhierarchie und dokumentiert sich in den **Projektstrukturen**, **interne Projektorganisation** ist die eigentliche Organisation des Projektteams und dokumentiert sich in den **Projektgremien**.

2.3.2 Organisationsstrukturen

Die Organisationsstrukturen des Projektes, die dann Teil der Aufbauorganisation des Unternehmens werden, definieren die Kompetenzen und Zuständigkeiten innerhalb eines Projekts. Nur durch eine zweckmäßige Aufbauorganisation wird eine vernünftige Aufgabenteilung in einem Projekt, an dem mehrere Personen arbeiten, überhaupt möglich. Es gibt mehrere Möglichkeiten, sein Team im Gesamtunternehmen zu organisieren – eine falsche Struktur kann den Erfolg eines Projekts verhindern.

Autonome oder reine Projektorganisation

Diese Projektorganisationsform ist in [Abbildung 2.3.2.1](#) dargestellt. Hierbei arbeiten alle beteiligten Mitarbeiter ausschließlich am Projekt – es wird sozusagen eine "Firma in der Firma" gegründet.



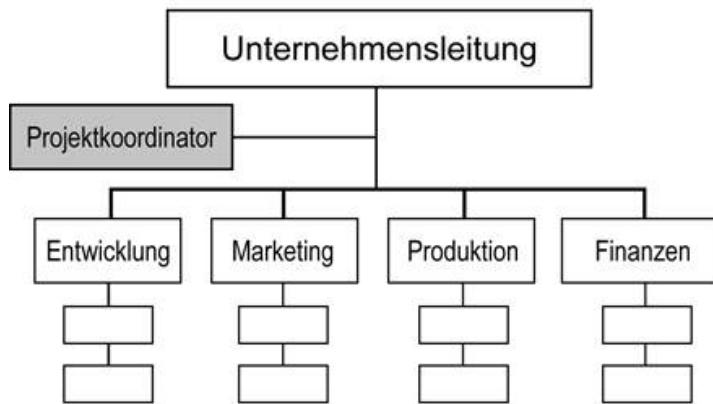
Vorteile: Die Projektmitarbeiter sind voll in das Projekt integriert. Die Identifikation mit dem Projekt ist hoch. Es gibt eindeutige Weisungsbefugnisse des Projektleiters in fachlicher und disziplinarischer Hinsicht.

Nachteile: Die Bereitstellung der Ressourcen und deren Verbleib nach dem Projekt müssen geregelt werden. Falls Ressourcen aus den Fachabteilungen herausgenommen werden, muss ein Ersatz gefunden werden. Projektmitarbeiter müssen nach dem Projekt wieder in ihre Abteilungen integriert werden.

Aufgaben des Projektleiters: ganzheitliche Zuständigkeit und Verantwortung für das Steuerungs-, Planungs- und Fachmanagement, hohe Anforderungen an die Führungsqualifikation und hohe Fachkompetenz.

Einfluss-Projektorganisation

In Abbildung 2.3.2.2 ist die Einfluss-Projektorganisation dargestellt. Hier muss der Projektkoordinator seinen Einfluss auf die Fachabteilungen geltend machen, in denen die Projektmitarbeiter weiterhin integriert bleiben und lediglich einen Teil ihrer Arbeitszeit für das Projekt "abzweigen".



Vorteile: Diese Form kommt mit einer Minimalausstattung aus. Es findet keine Ressourcenbindung statt.

Nachteile: Die Mitarbeiter haben nur eine geringe Identifikation mit dem Projekt. Der Projektleiter ist auf eine starke Linienstelle angewiesen, da er keine Entscheidungs- und Weisungsbefugnisse hat – weder fachliche noch disziplinarische.

Aufgaben des Projektleiters: projektzielbezogene Koordination der Fachabteilungen, koordiniert die Planung der Fachabteilungen, kontrolliert und überwacht den Projektfortschritt, stellt Hilfsmittel zur Verfügung, erstattet Meldung und fertigt Berichte an. Sie ist typisch für kleine und mittlere Unternehmen.

Matrix-Projektorganisation

Diese in Abbildung 2.3.2.3 gezeigte Organisationsform versucht, die Vorteile der autonomen und der Einfluss-Projektorganisation zu vereinen:

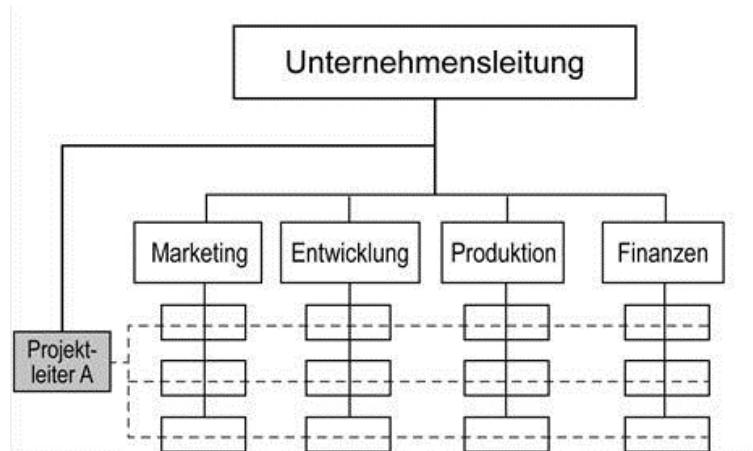


Abbildung 2.3.2.3: Matrix-Projektorganisation

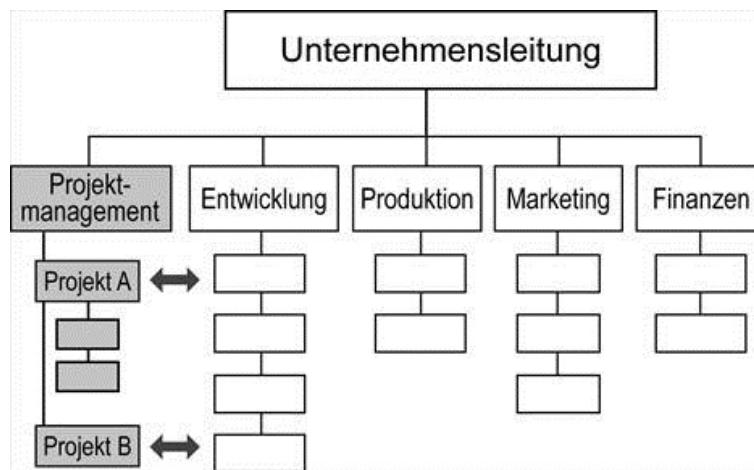
Vorteile: Diese Organisationsform ermöglicht einen flexiblen Personaleinsatz, denn die Mitarbeiter bleiben disziplinarisch den Leitern der Fachabteilungen, aus denen sie gewonnen wurden, unterstellt. Es ist Kontinuität in der fachlichen Weiterbildung und in der beruflichen Entwicklung der Mitarbeiter gegeben. Spezialwissen kann genutzt werden. Die Mitarbeiter haben ein großes Sicherheitsgefühl.

Nachteile: Es ist ein hoher Kommunikationsaufwand notwendig, da zwischen den Abteilungen und dem Projektleiter die Kompetenzen abgestimmt werden müssen. Der Projektleiter hat ja nur die fachliche Weisungsbefugnis. Dadurch können leicht Kompetenzkonflikte auftreten. Diese Machtteilung kann zur Verunsicherung der Mitarbeiter beitragen.

Aufgaben des Projektleiters: Projekt strukturieren, Rahmenplanung durchführen, Aufträge auslösen, Berichtswesen aufbauen, Unterschriften leisten und Änderungen in der Planung realisieren. Diese Organisationsform findet sich ebenfalls in kleinen und mittleren Unternehmen.

Auftragsprojektorganisation

Die Strukturen dieser Organisationsform gehen aus Abbildung 2.3.2.4 hervor. Der Projektleiter mit seiner Mannschaft, die ihm vollständig untersteht, tritt als Auftraggeber gegenüber den Abteilungen auf. Diese Mannschaft macht die Planungen und erarbeitet die fachlichen Spezifikationen für die Komponenten, die dann als Aufträge vergeben werden.



Vorteile: Diese Organisationsform ist matrixorientiert, beinhaltet aber keine Doppelunterstellungen der Projektmitarbeiter. Sie erhalten ihre Aufgaben im Projekt ebenso wie die Aufgaben im Tagesgeschäft von ihrem Abteilungsleiter.

Nachteile: Es wird eine Institution Projektmanagement im Unternehmen installiert, die dann auch ein Eigenleben entwickeln kann und in Konkurrenz zu den anderen Abteilungen tritt.

Aufgaben des Projektleiters: Projekt strukturieren, alle Planungen durchführen, Aufträge auslösen, die Auftragnehmer kontrollieren.

Linienprojektorganisation

Die letzte Organisationsform ist die Linienprojektorganisation. Es ist eigentlich keine neue Organisationsform, denn das Projekt wird ganz einfach von einer im Unternehmen vorhandenen Struktur übernommen: Der Leiter einer Einheit wird zum Projektleiter, die Projektmitarbeiter rekrutieren sich vollständig aus dieser Einheit. Solche Lösungen bieten sich besonders in Unternehmen mit Entwicklungs- oder Forschungsabteilungen an. Die Verhältnisse sind dann vergleichbar mit der reinen Projektorganisation.

Auswahl einer geeigneten Organisationsform

Die Auswahl einer geeigneten Organisationsform hängt nach BURGHARDT [Bur95] in erster Linie von der **Projektgröße** und der Einbeziehung anderer Bereiche oder Unternehmen (**Überbetrieblichkeit**) ab. Für sehr große Projekte (über 30 eigene Mitarbeiter) kommt eigentlich nur die reine Projektorganisation in Frage. Für Projekte mittlerer Größe und/oder hoher Überbetrieblichkeit bieten sich die Matrix- oder die Auftragsprojektorganisation an. Die Linienprojektorganisation ist nur für kleine Projekte (bis 10 Mitarbeiter) bei sehr geringer Überbetrieblichkeit geeignet. Die Einflus-

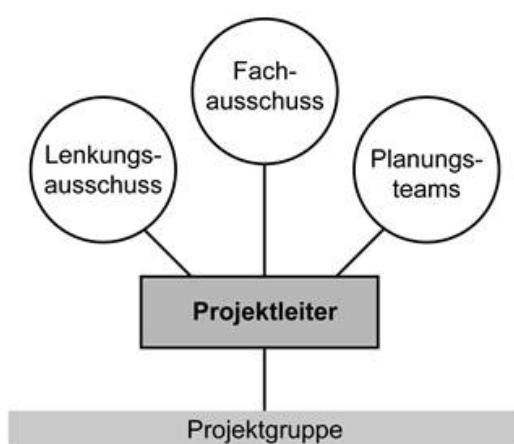
sprojektorganisation ist für kleine Projekte mit hoher Überbetrieblichkeit empfehlenswert.

HUBER-JAHN hat in einer Dissertation [Hub93] die Projektorganisationsformen von kleinen und mittleren Unternehmen untersucht und kommt dort zu der Aussage, dass in diesen Unternehmen fast die Hälfte der Projekte mit der Matrixprojektorganisation realisiert werden.

Ein weiteres Kriterium für die Auswahl einer geeigneten Organisationsform kann der gewählte **Entwicklungsprozess** sein. Projekte, die nach dem klassischen Wasserfalllebenszyklus abgearbeitet werden, sind oft durch eine hohe Überbetrieblichkeit gekennzeichnet und damit für die Auftrags- oder Matrixprojektorganisation geeignet. Der Rational Unified Process ist ein iteratives Prozessmodell für Softwareprojekte unterschiedlicher Größe – hier ist die reine oder die Matrixprojektorganisation empfehlenswert. Das eXtreme Programming ist für kleine Projekte mit eingearbeiteten, eigenen Mitarbeitern konzipiert und damit für die Linienprojektorganisation zu empfehlen.

2.3.3 Projektgremien

Analysiert man die interne Projektorganisation, so stellt man fest, dass es unterschiedliche Rollen oder bestimmte Aufgabenträger in einem Projekt gibt, die sich in den unterschiedlichen Gremien wiederfinden. Die folgende Abbildung zeigt einen Überblick von möglichen Gremien.



Ein entscheidender Punkt ist die Einrichtung eines Kontroll- bzw. Leitungsgremiums, das die für die Projektdurchführung notwendigen Entscheidungen vorbereitet und trifft. Bei kleineren Projekten in kleinen Unternehmen übernimmt diese Aufgaben die Geschäftsleitung. Bei größeren Projekten mit bereichsübergreifenden Aufgaben wird ein **Lenkungsausschuss** oder **Projektausschuss** [Kup96] gebildet. Er definiert die

Projektziele und Aufgaben mit dem Projektleiter, bestätigt die Ergebnisse der einzelnen Phasen und deren terminliche Planung, hilft bei auftretenden Problemen sowie bei der Abstimmung unterschiedlicher Bereiche. Der Lenkungsausschuss ist ein temporäres, projektbegleitendes Gremium, dessen Mitglieder sich aus den unterschiedlichsten Bereichen wie

- Management,
- Anwender und
- betroffene Produktionsabteilungen

zusammensetzen können. Der Lenkungsausschuss trifft sich bei Bedarf, mindestens aber monatlich. Er kann einen **Fachausschuss** berufen, der das Projekt bzw. die Projektgruppe fachlich berät.

Der Fachausschuss ist natürlich ebenfalls ein temporäres Organ, das sich aus Fachleuten der unterschiedlichsten Abteilungen zusammensetzen kann. Er hat nur beratende Funktionen und ist nicht weisungsbefugt.

Planungsteams sind nach BURGHARDT "Arbeitsgruppen, die mit Fachleuten aus unterschiedlichen Bereichen (Entwicklung, Fertigung, Vertrieb, Kaufmannschaft etc.) besetzt werden und über längere Zeit gemeinsam die Erstplanung eines Projekts (fachliches und technisches Konzept) vorantreiben. Hierbei tagen sie häufig nach der (4+1)-Tagungsregel, d.h. vier Tage der Woche im Team und einen Tag in der Heimdienststelle. Demgegenüber sind die Mitglieder eines Planungsausschusses nur in unregelmäßiger und auch geringerer Folge beisammen" [Bur95].

Einfluss auf ein Projekt kann auch der in fast allen Unternehmen vorhandene EDV-Ausschuss erhalten. Weitere Stellen, die versuchen, Ihr Projekt zu beeinflussen (oft natürlich gewollt), können neben dem Auftraggeber sein:

- Externe Berater (z.B. Unternehmensberatungen),
- Datenadministration,
- Datenschutzbeauftragter,
- Revisionskommissionen,
- Programmierer (aus anderen Projekten oder Abteilungen),
- Rechenzentrum.

Aufgabenträger ist aber in erster Linie das **Projektteam** mit dem **Projektleiter**. Er hat natürlich die größte Verantwortung und ist ab einer Gruppengröße von 7 Personen voll mit der Organisation ausgelastet. Seine Aufgaben umfassen die folgenden Punkte:

- schätzen (Aufwand, Zeitdauer),
- planen (Personal, Tätigkeiten, Ressourcen, Termine, Tests, Dokumentation),

- organisieren (Zuordnung von Tätigkeiten an Personen),
- kontrollieren (Qualität, Kosten und Termine),
- steuern und korrigieren,
- koordinieren,
- informieren,
- motivieren.

Dazu muss er – wie bereits unter dem Abschnitt Organisationsstrukturen erläutert – Weisungs- und Entscheidungsbefugnisse bekommen, insbesondere auch für Personalentscheidungen. Ein Projektleiter muss Erfahrungen besitzen, Führungsqualitäten aufweisen und die notwendigen Instrumentarien des Projektmanagements beherrschen. Er ist letztendlich verantwortlich für die Erfüllung der Projektziele!

Der **Auftraggeber** spielt eine entscheidende Rolle. Er muss sich an der Festlegung der nächsten Aufgaben beteiligen (welche Ziele sind aus betriebswirtschaftlicher Sicht jetzt wichtig und bringen sofort Geld?). Er soll Prioritäten setzen. Ein guter 'Customer' ist eine wesentliche Voraussetzung für ein erfolgreiches Projekt!

Mitarbeiter haben den größten Einfluss auf das Projekt: Zwischen den Mitarbeitern gibt es große Produktivitätsschwankungen. Es gibt nur sehr wenige hyperproduktive Entwickler. Teams sind der Schlüssel zum Erfolg: Gute Teams sind um den Faktor 3 produktiver als normale Teams. Das Ziel ist es, mit durchschnittlichen Entwicklern hyperproduktive Teams aufzubauen. Ein wichtiger Punkt ist die Motivation und Kommunikation im Team. Zu große Teams können kontraproduktiv sein – Overhead und Kommunikationsbedarf steigen an. Bestimmte Vorgänge können nicht beschleunigt werden. Mit vielen Personen dauert es oft länger!

Die Betroffenen beanspruchen oft gleiche Rechte – deshalb sollte man die "**Bill of Rights**" beim Projektstart festlegen. Folgende Rechte und Wünsche sollten berücksichtigt werden:

- Definition der Projektziele
- Festlegung der Budgets
- Auswahl des Projektleiters
- Auswahl der Projektmitarbeiter
- Methoden und Verfahren festlegen
- Zielvorgaben für die Mitarbeiter geben
- klare Aufgabenstellungen
- klare Berichtswege
- klare Leistungskriterien

"Der Schlüssel zum Projektmanagement ist ein Gleichgewicht der Kräfte zwischen 'business people' und 'technical people' [Bec00]. Geschäftsleute fällen betriebswirtschaftliche Entscheidungen, setzen Termine, Ziele und Prioritäten. Die Entwickler fällen technische Entscheidungen und geben Schätzungen ab (Aufwand).

"Bestehen Sie auf der Einrichtung eines Lenkungsausschusses. Erlauben Sie Kontrollen und berichten Sie regelmäßig. Schildern Sie Probleme und formulieren Sie Änderungen nur in Absprache mit dem Ausschuss! Nehmen Sie Einfluss auf die Zusammenstellung Ihres Teams. Sorgen Sie dafür, dass Sie für die Durchführung Ihrer Aufgaben ausreichende Befugnisse haben."

2.3.4 Aufgaben zur Projektorganisation



Aufgabe

Aufgabe 2.3.4-1

Projektorganisationen (PO) lassen sich anhand des Unterstellungsverhältnisses der Mitarbeiter und der Weisungsbefugnis (WB) des Projektleiters (PL) gegenüber den Mitarbeitern unterscheiden. Bitte kreuzen Sie die richtige Zuordnung an!

	Einfachunter-stellung	Mehrfachunter-stellung	Fachliche WB	Disziplinarische WB
Matrix-PO	?	?	?	?
Autonome-PO	?	?	?	?
Einfluss-PO	?	?	?	?



Aufgabe

Aufgabe 2.3.4-2

Herr Oldenburg, der Geschäftsführer einer Internet-Firma, will ein neues Produkt im Auftrag eines Kunden erstellen. Er wird dazu Mitarbeiter aus den Abteilungen Vertrieb, Service und Rechenzentrum für die Projektlaufzeit dem Projekt zuordnen. Projektleiter wird Herr Rubusch. Herr Rubusch bekommt die fachliche Weisungsbefugnis. Welche Organisationsform kann gewählt werden?



Aufgabe

Aufgabe 2.3.4-3

Welche Aufgaben übernimmt der Lenkungsausschuss?



Aufgabe

Aufgabe 2.3.4-4

Wie setzt sich der Lenkungsausschuss zusammen?



Aufgabe

Aufgabe 2.3.4-5

Herr Oldenburg, der Geschäftsführer einer Internet-Firma, will ein neues Entwicklungsprojekt auflegen. Er wird dazu die 6 Mitarbeiter einer Entwicklungsgruppe des Unternehmens dem Projekt zuordnen. Gruppenleiter ist Herr Markwart, der damit auch die Projektleitung übernimmt. Welche Organisationsform liegt hier vor?



Gliederung

3 Projektplanung

3 Projektplanung

- 3.1 Grundlagen der Projektplanung
- 3.2 Planungsreihenfolge
- 3.3 Planungstechniken



Lernziele

Was sollten Sie nach diesem Kapitel wissen?

- Was versteht man unter Planen?
- Was wird im Projekt alles geplant?
- Was wird zuerst geplant?
- Welche Planungstechniken können angewandt werden?



Gliederung

3.1 Grundlagen der Projektplanung

3.1 Grundlagen der Projektplanung

- 3.1.1 Definition
- 3.1.2 Meilensteine
- 3.1.3 Personal
- 3.1.4 Aufgaben zu den Grundlagen der Projektplanung

Dieser Abschnitt führt in die Grundlagen der Projektplanung ein. Die Begriffe **Planung** und **Projekt** gehörten schon seit jeher eng zusammen. So bedeutet "Pro" nach vorne und "jekt" werfen, also den Blick nach vorne werfen, was der wohlbekannten Definition von Planung entspricht. **Planung wiederum ist das gedankliche Vorwegnehmen zukünftigen, zielgerichteten Handelns.**

3.1.1 Definition

Unter **Projektplanung** werden die systematische Informationsgewinnung über den zukünftigen Ablauf des Projektes und die gedankliche Vorwegnahme des notwendigen Handelns im Projekt verstanden, z.B.:

- Arbeiten und Tätigkeiten,
- Ressourcen (Einsatzmittel),
- Kosten,

- Termine,
- Qualität.

Mit anderen Worten: Man legt Ziele fest und versucht die notwendigen Tätigkeiten und Ressourcen für diese Ziele zu bestimmen. Grundlage dieser Voraussagen sind Erfahrungen aus früheren Projekten oder aus dem laufenden Projekt, verbunden mit einigen Techniken, die man dazu einsetzen kann (vgl. Abschnitt 3.3 Planungstechniken).

"In einem Projekt spielen wir alle ein Spiel – wir stellen Vermutungen an, wir haben eine Ahnung von dem, was wir machen sollen und wie die Lösung aussehen sollte, und wir tun so, als seien wir uns dessen, was wir sagen, sicher. Aber in Wirklichkeit sind wir uns über viele Dinge unsicher. Wir bluffen ziemlich viel und hoffen das Beste. Wir agieren auf der Grundlage mangelhafter Informationen, und die Ergebnisse zeigen das auch häufig. Ein kluger Projektmanager stellt diese Tatsache in Rechnung, indem er in sein Projekt 'Kontrollstellen'" einbaut, an denen die Annahmen validiert werden können, und indem er zusätzliche Maßnahmen einplant, um an bessere Informationen zu kommen. Der iterative Ansatz ist hierfür hervorragend geeignet: Er erlaubt es uns, Informationen zu sammeln, während wir gleichzeitig unsere Pläne verbessern und die Richtung ändern können" [Kru01].

Einem Sprichwort zufolge überlebt keine Planung den ersten Kontakt mit dem Feind. Ein kluger Projektmanager weiß das und bietet Möglichkeiten an, Pläne aufgrund neuer Informationen zu überarbeiten.

Die Planung ist in einen **Regelkreis** der gesamten Projektdurchführung eingebunden (vgl. Abbildung 3.1.1.1) und damit also keine einmalige Aufgabe, sondern eine Aufgabe, der sich der Projektleiter ständig unterziehen muss. Durch Maßnahmen der Projektüberwachung oder -kontrolle werden Ist-Daten erhoben. Diese Daten werden für die Projektsteuerung verwendet und führen zur Überarbeitung der Planungsunterlagen. Auf keinen Fall darf man die Korrektur der Planungsunterlagen dabei unterschätzen, damit das Gesamtprojekt überhaupt steuerbar bleibt und Pläne vernünftige Soll-Vorgaben liefern können.

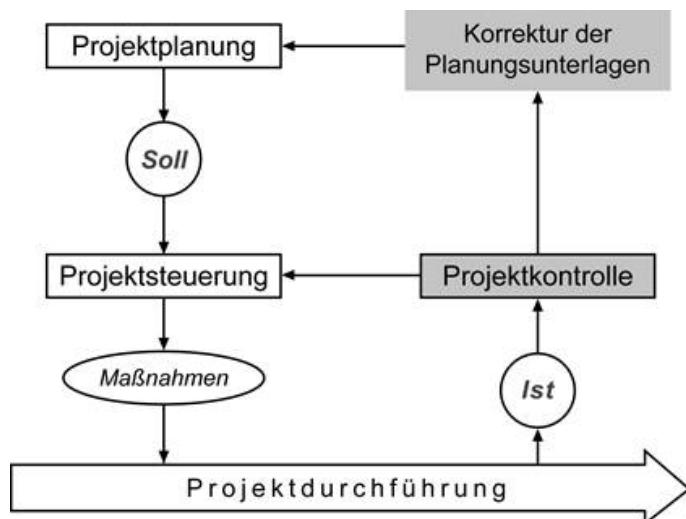


Abbildung 3.1.1.1: Planung als Regelkreis

Kein Mensch ist in der Lage, einen vollständigen detaillierten Plan für ein größeres Softwareprojekt zu erarbeiten. Vielmehr wird man bei der Übernahme eines iterativen Prozessmodells **die Planung anpassen** und einen übergeordneten Plan, den sogenannten **Phasenplan**, erarbeiten und dort eine grobe Aufteilung des Budgets und der Termine vornehmen. Eine detaillierte Planung erfolgt nur für die nächste oder übernächste Iteration im sogenannten **Iterationsplan** (vgl. [Abschnitt 1.3 Projektphasen und Prozessmodelle](#)).

Schlechte Planungen sind der Hauptgrund für das Scheitern von EDV-Projekten (vgl. [Abbildung 1.1.2 im Abschnitt 1.1 Motivation](#)). Deshalb müssen die Planungen mit großer Sorgfalt und mit ausreichend Zeit durchgeführt werden. Für größere Softwareprojekte bieten sich **Planungsteams** an. Dessen Mitglieder sollten möglichst "alte Hasen" sein, die über eine reiche Projekterfahrung verfügen.

3.1.2 Meilensteine

Die oben genannten Kontrollpunkte werden im Software-Projektmanagement als Meilensteine bezeichnet. **Meilensteine** sind Ereignisse, die das Eintreten eines bestimmten Zustandes beschreiben. An Meilensteinen sollten immer konkrete Projektergebnisse vorliegen, die überprüft werden können, z.B. die Layoutvorlage oder das Objektmodell.

Definition: Ein Meilenstein ist ein Ereignis von besonderer Bedeutung oder Auslöser von vordefinierten Funktionen innerhalb eines Projektes. Er ist eine geschlossene Ergebnismenge an einem "präzisen Prüfpunkt", die "überprüfbar und versionsfähig" ist.

Der Meilenstein hat ein definiertes Sachergebnis (Meilensteininhalt) und einen Fertigstellungstermin (Meilensteintermin).

Meilensteininhalte sind

- wesentlich
- überprüfbar
- übergebbar
- eindeutig festgelegt
- im Voraus definiert (Phasenorganisation)

Meilensteintermine werden in der Projektplanung ermittelt.

Meilensteine haben zusätzlich den schönen Effekt der Motivationssteigerung. Durch die häufigen Zwischenziele sind die Mitarbeiter wahrscheinlich immer recht froh, wieder mal einen Teilbereich abgeschlossen zu haben und nicht dem Frust des noch so weit entfernten Projektziels nachgeben zu müssen.

Zur Definition eines Meilensteins gehören:

- Name des Meilensteins,
- Verantwortlicher für den Meilenstein,
- Termin für die Erbringung der Meilensteinergebnisse.

3.1.3 Personal

Die gesamte Planung beruht auf Erfahrungen und Prognosen. Der größte Unsicherheitsfaktor ist der Mensch mit seinen Fähigkeiten und Fertigkeiten. Die wichtigste Tätigkeit des Projektleiters ist deshalb die **Personalführung** und in der Planungsphase die **Mitarbeiterauswahl** – ein Recht, das sich der Projektleiter niemals nehmen lassen darf. Versuchen Sie, den Mitarbeitern, die Sie einstellen, **Sicherheit** zu geben. Tom DeMarco schreibt dazu in seinem Buch "Der Termin" [DeM98]:



Anmerkung

- Menschen können Veränderungen nur in Angriff nehmen, wenn sie sich sicher fühlen.
- Veränderung ist eine entscheidende Voraussetzung für den Erfolg jeder Projektarbeit (und wahrscheinlich auch für die meisten anderen lohnenden Unternehmungen).
- Fehlende Sicherheit bewirkt fehlende Risikobereitschaft.

- Risikovermeidung ist fatal – sie führt dazu, dass die mit einem Risiko verbundenen Chancen ungenutzt bleiben.
- Unsicherheit entsteht, wenn Menschen sich direkt bedroht fühlen oder Angst vor Machtmisbrauch haben.

Zur **Personalbeschaffung** schreibt DeMarco an einer anderen Stelle seines Buches:



Anmerkung

- Bei Beginn der Schlacht (oder des Projekts) ist die eigentliche Arbeit des Managers bereits getan.
- Wer Personal einstellt, braucht dafür alle für das Management relevanten Körperteile: Herz, Seele, Nase und Bauch (vor allem das Gefühl aus dem Bauch heraus).
- Versuchen Sie nicht, alles allein zu tun – zwei Bäuche sind besser als einer.
- Bitten Sie neu eingestellte Mitarbeiter, ein Projekt zu leiten, dessen Anforderungen exakt denen entsprechen, denen sie bereits früher gerecht wurden, und herausfordernde Ziele dieses Mal zurückzustellen.
- Fragen Sie nach Empfehlungen: Vielleicht kann Ihnen die Person, die Ihnen am meisten zusagt, andere gute Kandidaten nennen.
- Reden ist Silber, Zuhören ist Gold.

3.1.4 Aufgaben zu den Grundlagen der Projektplanung



Aufgabe

Aufgabe 3.1.4-1

Ist ein Meilenstein:

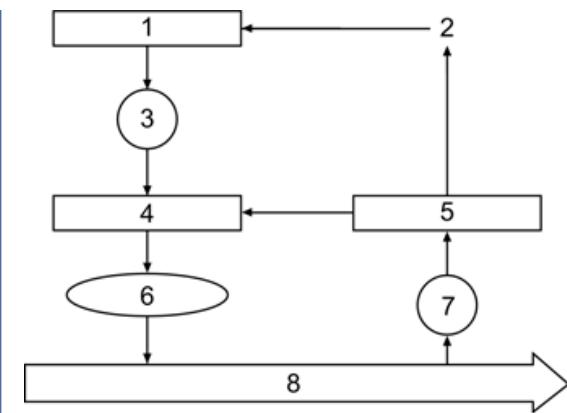
- A) Ein Vorgang
- B) Ein Arbeitspaket
- C) Ein Ereignis



Aufgabe

Aufgabe 3.1.4-2

Ordnen Sie die Zahlen in der folgenden Abbildung den richtigen Begriffen zu!



Projektdurchführung, Projektsteuerung, Projektkontrolle, Soll, Ist, Projektplanung, Maßnahmen, Korrektur der Planungsunterlagen



Aufgabe

Aufgabe 3.1.4-3

Eine Grundvoraussetzung für die Projektplanung ist die Auswahl des richtigen Personals. Ordnen Sie die folgenden Aussagen in "richtig" oder "falsch" ein!

- a) Keiner im Team darf sich sicher sein, damit man möglichst viel Druck ausüben kann.
- b) Einstellungsgespräche müssen immer alleine geführt werden. Argumente von Kollegen verwirren nur.
- c) Hören Sie beim Einstellungsgespräch Ihrem Gegenüber zu.
- d) Reden Sie möglichst viel, damit der neue Kollege einen positiven Eindruck von Ihren Führungsqualitäten bekommt.
- e) Sagen Sie nie direkt, wenn Ihnen etwas nicht gefällt. Sparen Sie alle Fehler eines Mitarbeiters für ein großes "Donnerwetter" auf. Sie können so Ihre Machtposition stärken.
- f) Versuchen Sie einen Mitarbeiter mit einer Aufgabe zu betrauen, bei der er auf Erfahrungen zurückgreifen kann, um Sicherheit in der neuen Umgebung zu erlangen.

3.2 Planungsreihenfolge



Gliederung

3.2 Planungsreihenfolge

- 3.2.1 Vorgehen und Motivation
- 3.2.2 Strukturplanung
- 3.2.3 Einsatzmittelplanung

- [3.2.4 Terminplanung](#)
- [3.2.5 Kostenplanung](#)
- [3.2.6 Aufgaben zur Planungsreihenfolge](#)

Dieser Abschnitt befasst sich mit der Reihenfolge, in der die unterschiedlichen Aufgaben und Arbeitspakete, die bei einem Projekt anfallen, abgearbeitet werden. In diesem Zusammenhang wird auf folgende Problemfelder eingegangen:

- Strukturplanung,
- Einsatzmittelplanung,
- Terminplanung und
- Kostenplanung.

3.2.1 Vorgehen und Motivation

Pläne bauen aufeinander auf. Nach Kupper [Kup96] hat sich die folgende Reihenfolge in der Praxis bewährt.



Aus der Projektzielbeschreibung (vgl. Abschnitt 2.1 Projektziele), die die Plattform bildet, muss abgeleitet werden können, welche konkreten Arbeiten/Tätigkeiten zur Realisierung notwendig sind. Die "große" Aufgabe wird in **Arbeitspakete** zerlegt. In der Literatur wird dieser Schritt im Allgemeinen als **Strukturplanung** bezeichnet. Die Strukturplanung enthält auch die Kostenstruktur, die dann eine effiziente Kostenkontrolle gestattet. Sind die Aufgaben abgesteckt, kann überlegt werden, in welcher Reihenfolge diese bzw. unter welchen **Ablaufbedingungen** die Arbeitspakete realisiert werden können: Gibt es Abhängigkeiten oder können Arbeitspakete nebeneinander bearbeitet werden? Ablaufbedingungen spiegeln sich im Netzplan wieder. Der Netzplan ist eine

grafische Übersicht der Arbeitspakete und deren Abhängigkeiten und erhöht wesentlich die Transparenz eines Projekts, vgl. dazu den Abschnitt 3.3 Planungstechniken.

Arbeitspakete erfordern einen bestimmten **Aufwand**, dieser muss geschätzt werden. Der Aufwand wird im Allgemeinen in der Maßeinheit **Personen-** oder **MannMonate (MM)** angegeben. Die Aufwandsschätzung ist sehr kompliziert und mit großen Unsicherheiten belastet. Hier bietet sich die Anwendung von Planungstechniken an, die im gleichnamigen Abschnitt (3.3) beschrieben wird. Die Ergebnisse der Aufwandschätzung fließen auch in den Netzplan ein, der dann die Ermittlung der Gesamtdauer und der Pufferzeiten einzelner Arbeitspakete erlaubt. Erst danach kann auf die Einsatzmittel, die Kosten und die Termine geschlossen werden.

Daraus ergibt sich folgende **Planungsreihenfolge**:



Wichtig

1. Teilen Sie den Projektablauf in kleine Schritte (Arbeitspakete und Vorgänge finden)!
2. Benennen Sie Meilensteine (Ecktermine)!
3. Überprüfen Sie zeitliche und inhaltliche Abhängigkeiten und Bedingungen zwischen den Arbeitspaketen!
4. Ordnen Sie die Ressourcen und Einsatzmittel zu!
5. Aus der Ressourcenplanung resultieren Kosten. Erfassen Sie diese!
6. Terminieren Sie!

3.2.2 Strukturplanung

Der **Projektstrukturplan** ist der wichtigste Plan überhaupt. Er bildet die Grundlage aller weiteren Überlegungen. Die Strukturplanung enthält die Elemente Produktstruktur, Projektstruktur und Kontenstruktur (Burghardt [Bur95]).

Strukturplanung		
Produktstruktur	Projektstruktur	Kontenstruktur

■■■ Tabelle 3.2.2.1: Strukturplanung

Produktstruktur (Bildung von Arbeitspaketen)

Die Produktstruktur beinhaltet die Arbeitspakete, die realisiert werden müssen, damit das Gesamtprodukt entstehen kann. Ein **Arbeitspaket** ist unter der Betrachtung der Projektplanung die kleinste nicht teilbare Einheit mit einem definierten Anfangs- und

Endzeitpunkt, also einer gewissen Dauer, dem Ressourcen zugeordnet werden können. An dieser Stelle soll der schon oft erwähnte Grundsatz wiederholt werden, dass der Detailliertheitsgrad der Planungen immer dem Wissensstand angepasst werden muss. Eine detaillierte Feinplanung ist bei Softwareprojekten nur für kurze Zeitabstände machbar. Projekte beschäftigen sich im Allgemeinen mit komplexen Aufgabenstellungen – also solchen, die sich in viele Arbeitspakete zerlegen lassen. Speziell bei Softwareprojekten resultiert die Komplexität aus den folgenden drei Punkten:

- aus der Problemstellung,
- aus dem Entwicklungsprozess,
- aus der Flexibilität, die geboten werden soll.

Die in Projekten zu bearbeitenden Problemstellungen sind komplex – heute wird Software dazu genutzt, um Flugzeuge zu steuern oder Weltraumsonden zu navigieren. Über den Entwicklungsprozess haben wir in verschiedenen Abschnitten ausreichend diskutiert. Heutige Software soll flexibel sein – am augenscheinlichsten wird dies bei Standardsoftware, die nicht nur eine, sondern zahlreiche Schriftarten bereitstellt und neben Schriftzeichen auch mit Zeichnungselementen und Bildern umgehen kann.

Es ist ein großer Vorteil für Ihr Projekt, wenn die Anforderungen an das System (aus der Projektzielbeschreibung) in Form von **Anwendungsfällen** vorliegen (vgl. Abschnitt Risiken, wo eine kurze Beschreibung dieser Technik vorgenommen wird). Das Problem der Strukturplanung reduziert sich dann auf die Zerlegung der Anwendungsfälle in Arbeitspakete, und zwar immer nur für den Anwendungsfall, der zur Realisierung ansteht. Da auch Anwendungsfälle noch eine hohe Komplexität aufweisen, kann man für deren Bewältigung ganz allgemeine Eigenschaften von komplexen Systemen nutzen.

Komplexe Systeme weisen im Allgemeinen einen **hierarchischen Aufbau** auf, der für die Produktstrukturplanung genutzt werden kann, denn **Hierarchieebenen** sind immer auch Abstraktionsebenen.

In der Abbildung 3.2.2.2 wird diese Eigenschaft anhand eines Baumes verdeutlicht.

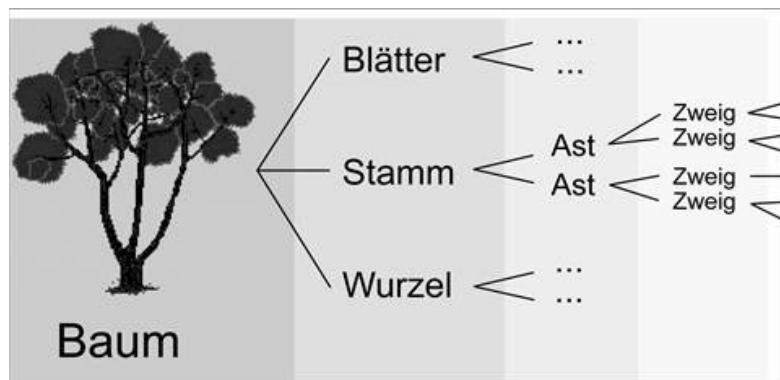


Abbildung 3.2.2.2: Hierarchieebenen am Beispiel eines Baumes

Erste Schlussfolgerungen für die Produktstrukturplanung könnten also die folgenden sein:

- Finde Abstraktionsebenen, Hierarchieebenen in einem System bzw. in einem komplexen Anwendungsfall.
- Verfeinere die Ebenen soweit, bis Arbeitspakete gefunden wurden, die überschaubar, programmierbar und kontrollierbar sind.
- Definiere Beziehungen und Abhängigkeiten zwischen den Komponenten, die wichtig für deren Funktion sind.

Neben der Untersuchung von Hierarchien liefert auch der zugrunde liegende Softwareentwicklungsprozess Hinweise auf Arbeitspakete. Er schreibt eine Reihe von Methoden vor, die in einer zeitlichen Reihenfolge abzuarbeiten sind; daraus müssen Arbeitspakete abgeleitet werden. Im Anhang 7.7 wird das Vorgehen anhand eines Beispiels verdeutlicht.

Konfigurationsmanagement

Artefakte müssen verwaltet und für den Test und die Auslieferung konfiguriert werden. Eine wesentliche Voraussetzung dafür ist die **Versionskontrolle**. Dabei sollten Sie die beiden folgenden Grundsätze beherzigen:

- Bestimmte Version eines Dokuments (z.B. Use Case Analysis) – nicht jede neue Version muss in eine neue Datei.
- Bestimmte Version eines Quellcodes – jede neue Version muss in eine neue Datei gespeichert werden, um zu verhindern, dass lauffähige Versionen verloren gehen.

In der Tabelle 3.2.2.3 ist ein Vorschlag für die Versionierung von Dokumenten gegeben. Die Versionskontrolle erfolgt mit einer Tabelle innerhalb des Dokuments.

Bearbeitungsgeschichte

Datum	Version	Beschreibung	Autor
<tt/mm/jjjj>	<x.x>	<kurzer Text mit den wesentlichen Änderungen>	<Name>

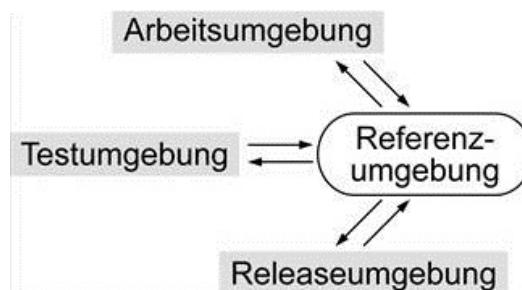
■ Tabelle 3.2.2.3: Tabelle für die Versionierung von Dokumenten

Die Verwaltung von Softwareeinheiten ist etwas aufwendiger und meiner Meinung nach ohne Softwaretools nicht mehr zu realisieren (auf jeden Fall ist es eine Dummheit, diese nicht zu verwenden). In Bezug auf die Verwaltung von Software müssen einige Begriffe geklärt werden.

Softwarekonfiguration ist die Gesamtheit zusammenpassender Softwareeinheiten. Eine **Softwareeinheit** ist der kleinste für diesen Zweck als unteilbar behandelte Bestandteil eines Softwareprodukts. Eine Einheit wird als Ganzes betrachtet und geändert. Eine konsistente Menge von Softwareeinheiten, die als Ganzes bestimmten Anforderungen genügen, bezeichnet man als **Release**. [Lud91]

Der Zweck der Versionskontrolle bei der Softwareentwicklung ist es also, erstens ein schon lauffähiges Release nicht zu zerstören und zweitens zu ermöglichen, dass Entwickler veraltete Versionen einer Softwareeinheit für die Weiterentwicklung nutzen können. Das hört sich trivial an, ist aber für viele ein scheinbar unlösbares Problem.

Grundvoraussetzung für eine sichere Versionskontrolle ist eine Referenzumgebung, die Einheiten entgegennimmt, versioniert und wieder verteilt.

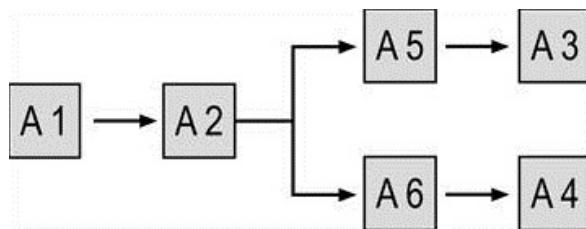


■ Abbildung 3.2.2.4: Zentrale Rolle der Referenzumgebung

Automatische Versionskontrolle und -verteilung in der Referenzumgebung ist unabdingt zu empfehlen, wenn mehrere Mitarbeiter vernetzt arbeiten, damit garantiert wird, dass jeweils die aktuellen Versionen genutzt werden. Ein Beispiel ist das Programm CVS – Concurrent Versions System – eine Client-Server-Anwendung, die den Entwicklern einen Zugriff auf die aktuellsten Versionen garantiert, sobald ein Internetanschluss vorhanden ist. Sie ist für viele Plattformen erhältlich.

Ablaufplanung

Voraussetzung für die Zeitplanung ist die Festlegung der Abhängigkeiten zwischen den Arbeitspaketen. Das heißt, Sie müssen Vorgänger und Nachfolger der Arbeitspakete bestimmen. Das kann tabellarisch oder grafisch passieren. Dazu wird zweckmäßigerweise die Netzplantechnik verwendet, die im Abschnitt Planungstechniken ausführlich behandelt wird. Zur Verdeutlichung soll an dieser Stelle eine kleine Abbildung reichen, die die Abhängigkeiten von sechs Arbeitspaketen darstellt.



Arbeitspaket	Vorgänger	Nachfolger
A1		A2
A2	A1	A5, A6
A3	A5	
A4	A6	
A5	A2	A3
A6	A2	A4

| Tabelle 3.2.2.5: Ablaufplan der Arbeitspakete A1 bis A6

Projektstruktur (Mitarbeiterzuordnung)

Hier können wir uns relativ kurz halten. Es geht um die Zuordnung von Arbeitsgruppen Ihrer Projektorganisation zu den Aufgabenpaketen – wer macht also was. Sie müssen dabei den Ablaufplan der Arbeitspakete beachten, damit Sie Ihre Mitarbeiter und weiteren Ressourcen nicht überlasten. Alle haben Kostensätze, die eine Kostenkontrolle ermöglichen. Auch hier ist eine Anbindung an den Netzplan (siehe Planungstechniken) zu empfehlen und der Einsatz von Planungstools, z.B. das Programm Microsoft Project. Der Projektstrukturplan ist eine weitere Grundlage für den Einsatzmittelplan, der weiter unten in diesem Abschnitt behandelt wird.

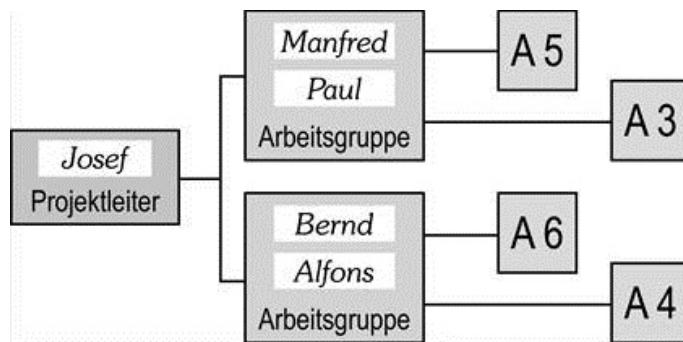


Abbildung 3.2.2.6: Projektstruktur

In der Abbildung ist eine kleine Projektgruppe dargestellt. Josef, der Projektleiter, hat zwei Gruppen gebildet, die jeweils die Arbeitspakete parallel abarbeiten.

Kontenstruktur

Eine geeignete Kontenstruktur ermöglicht dem Projektmanagement, Projekte und Teilprojekte kostentransparent zu machen. Die mögliche Aufteilung und Gliederung der Konten hängt in vielen Unternehmen von den Möglichkeiten der Buchhaltung und der verwendeten Software ab. Die Kontenstruktur sollte nach Burghardt [Bur95] die folgenden Ziele abdecken:

- Kostenverursacher erkennbar machen (Kostenherkunft),
- Kostenkomponenten darstellen (Kostenschwerpunkte),
- Konten auf den Terminplan ausrichten (zeitliche Synchronisierung),
- Beziehung zur Projekt- und Produktstruktur herstellen (technische Synchronisierung),
- Kostenkalkulation und -kontrolle unterstützen (Kostenmanagement).

Im Anschluss an die klassische Kostenrechnung können weiterhin **Kostenstellen** und **Kostenarten** betrachtet werden. Weitere Hinweise finden Sie in Kapitel 4 Projektkontrolle.

3.2.3 Einsatzmittelplanung

Die Einsatzmittelplanung (Planung der für das Projekt notwendigen Ressourcen) umfasst neben dem oben schon erwähnten Personal auch Geld- und Betriebsmittel (Maschinen, Server, Materialien). Allerdings unterscheiden sich Softwareprojekte hier wieder erheblich von anderen Projekten wie Bauprojekten: Alles hängt bei Softwareprojekten am richtigen Personal, nur wenig ist von anderen Einsatzmitteln abhängig. Deshalb wird hier nur auf die Personalplanung ausführlich eingegangen. Die Personalplanung ist unter folgenden Kriterien aufzubauen:

- Berechnung des Personalbedarfs,

- organisatorische Zuordnung (vgl. Projektstruktur),
- Qualifikation des Personals und dessen zeitliche und örtliche Verfügbarkeit.

Der wichtigste Punkt, die Personalauswahl, wurde bereits im Abschnitt Grundlagen angesprochen, da sie tatsächlich eine wesentliche Grundlage für die erfolgreiche Abwicklung eines Projekts darstellt. Der Einsatzmittelplan erfordert die Planung weiterer Aspekte, die einen Einfluss auf die Berechnung des notwendigen Personalbedarfs haben. Dazu gehören insbesondere der Kommunikationsplan und der Personal- und Ausbildungsplan, in deren Folge Kapazitäten gebunden werden.

Der **Kommunikationsplan** enthält Aussagen zum Berichtswesen, den Arbeitsberatungen, Brainstormings und Blue Fridays. Ein wesentlicher Punkt ist das **Berichtswesen**, das von der Projektorganisation (vgl. Abschnitt Organisationsformen) geprägt wird:

- Berichte der Mitarbeiter: z.B. Tätigkeitsbericht (vgl. Kapitel 4 Projektkontrolle)
- Berichte des Projektleiters: z.B. Projektfortschrittsbericht, monatlicher Mittelverbrauch (vgl. Kapitel 4 Projektkontrolle)

Der **Personalplan** und der **Ausbildungsplan** dienen dazu, Urlaub und Ausbildung in Ihrer Kapazitätsplanung zu berücksichtigen:

- Berücksichtigen Sie, dass Bildungsurlaub im Tarifvertrag zugesichert sein kann.
- Urlaub wie gewünscht ermöglichen, das steigert die Arbeitsmoral. Urlaubssperren nur bei exponierten Abschnitten (Einführung eines Teilsystems) aussprechen.
- Weiterbildungskurse können Kosten sparen (Zeit). Die Kosten für diese Kurse sind nicht unerheblich und müssen geplant werden.
- Karriere der Mitarbeiter fördern.
- Was passiert nach dem Projekt? Aufstiegsmöglichkeiten innerhalb des Projekts organisieren.
- Veröffentlichungen sowie Teilnahme an Tagungen und Messen ermöglichen.

Wenn Sie den realistischen **Arbeitsvorrat** berechnen wollen, müssen Sie gerade Urlaub, Krankheit und sonstige Ausfallzeiten mit berücksichtigen. Als Faustregel können Sie pro Jahr mit 6 Wochen Urlaub und 2 Wochen Krankheit rechnen, sodass noch 10 volle MM der 12 vollen MM für eine Arbeitskraft übrig bleiben.

Hieraus lässt sich der **Produktivanteil (PA)** berechnen (**Formel (2-1)**):

$$\text{Produktivanteil (PA)} = \frac{\text{Netto - Stundenzahl (S}_{\text{netto}}\text{)}}{\text{Brutto - Stundenzahl (S}_{\text{brutto}}\text{)}} \cdot 100\%$$

$$\text{PA} = \frac{10 \text{ MM}}{12 \text{ MM}} \cdot 100\%$$

$$\text{PA} = 83\%$$

In der obigen Formel wird die Netto-Stundenzahl des Betrachtungszeitraumes durch die tatsächlich in einem Jahr zur Verfügung stehenden MM ausgedrückt. Die Brutto-Stundenzahl wird durch die theoretisch möglichen MM ausgedrückt. Ein Mannmonat hat, je nach Tarifgebiet, z.B. 22 Arbeitstage mit 8 Arbeitsstunden pro Tag.

Kent Beck schreibt in seinem Buch [*Bec00*], dass von 21 Arbeitstagen nur 7 Tage so genannte "ideale Programmertage" sind, an denen die Mitarbeiter wirklichen Projektfortschritt realisieren. Die restlichen Tage gehen für Arbeitsberatungen, Kommunikation, Lernen, usw. "verloren".

In der Detailplanung können Ausfallzeiten auch genauer quantifiziert werden. Dazu können entsprechende **Projektkalender** oder **Arbeitskalender** genutzt werden. In diesen Kalendern sind die absehbaren Ausfallzeiten schon berücksichtigt. Dazu zählen Feiertage oder eben Urlaubstage sowie geplante Dienstreisen.

Berücksichtigen Sie Ausfallzeiten bei der Bestimmung der notwendigen **Mitarbeiterzahl (D)** (**Formel (2-2)**), deren Ausgangspunkt die Aufwandsschätzungen sind. Probleme der Aufwandsschätzung werden im Abschnitt 3.3 Planungstechniken besprochen.

$$\text{Anzahl Mitarbeiter} = \frac{\text{Aufwand}}{\text{Dauer}} \cdot \frac{1}{\text{Produktivanteil}}$$

$$D = \frac{A}{T} \cdot \frac{1}{PA}$$

Bedenken Sie, dass der Aufwand nur geschätzt wurde, dass auch Sie in keinen Ihrer Mitarbeiter "hineinschauen" können – dann bekommen Sie eine ungefähre Vorstellung davon, warum so viele Softwareprojekte aufgrund falscher Planungen scheitern (vgl. Abschnitt 1.1 Motivation). Je kleiner das Aufgabenpaket ist, umso genauer können Sie schätzen!

Die Formel (2-2) setzt auch voraus, dass die Aufgabe tatsächlich teilbar ist und kein zusätzlicher Kommunikationsbedarf entsteht. Diese Annahme ist nur in den wenigsten Fällen realistisch (vgl. Abschnitt 3.3 Planungstechniken).

3.2.4 Terminplanung

Ohne die oben beschriebene Strukturplanung ist eine Terminplanung unmöglich. Der Strukturplan enthält auch die **Ablaufplanung**. Bei der Ablaufplanung werden die Vorgänger und Nachfolger bestimmt. Weiterhin muss für jedes Aufgabenpaket der **Aufwand** bekannt sein, denn aus dem Aufwand und den eingesetzten Einsatzmitteln ergibt sich die Dauer der Vorgänge (vgl. dazu Formel (2-2) und den Abschnitt 3.3 Planungstechniken). Zweckmäßigerweise ist für die Terminplanung die Netzplantechnik zu verwenden, bei kleineren Projekten kann auch mit Balkendiagrammen gearbeitet werden. Beide Techniken werden im Abschnitt Planungstechniken vorgestellt.

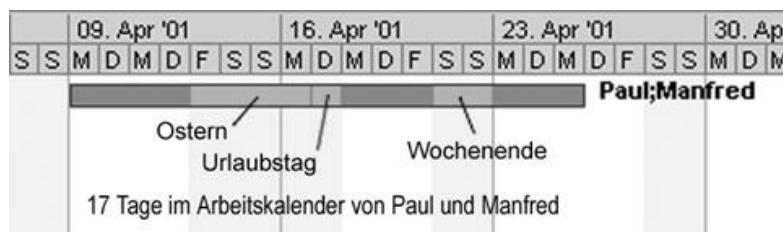
Bei der Grobplanung werden die Gesamtprojektlaufzeit und der Beginn oft vorgegeben. Meist aus Marketinggründen – weil es heute so wichtig ist, "Erster am Markt" zu sein. Für die Grobplanung müssen dann die Eckwerte für wichtige Meilensteine gesetzt werden.



■ Abbildung 3.2.4.1: Darstellung von Hauptetappen auf der Zeitachse innerhalb der Grobplanung

Die weitere Terminierung der Aufgabenpakete lt. Strukturplanung muss jetzt so erfolgen, dass alle Ziele für die Releases auch tatsächlich erfüllt werden. Für die Terminkontrolle und die Qualitätskontrolle sollte man die Aufgabenpakete ergebnisorientiert bilden (vgl. Strukturplanung). Auch sollte man die Detailplanung so anlegen, dass die Aufgabenpakete nicht zu viele Bearbeiter erfordern – bzw. es jeweils einen Verantwortlichen gibt.

Ob ein Projekt tatsächlich vom Anfang bis zum Ende im Detail durchgeplant wird, hängt vom verwendeten Prozessmodell ab. Dazu gibt es im Abschnitt Planungshinweise (weiter unten in diesem Abschnitt) für konkrete Prozessmodelle praktische Anregungen.



Das Arbeitspaket hat einen Aufwand von 20 MannTagen
 Dauer = 20 MannTage / 2 Mann = 10 Tage

Beginn: Mo., 9.4.2001, Ende: Mi., 25.4.2001

Abbildung 3.2.4.2: Auszug aus der Detailplanung unter Verwendung eines Balkendiagramms

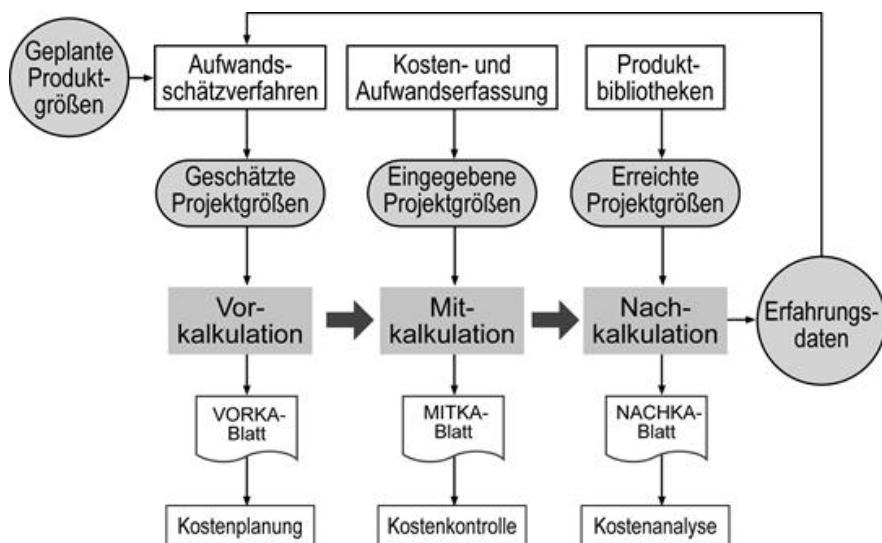
Terminplanung ist auch immer mit Terminbeschleunigung verbunden. Jeder möchte seine Ergebnisse so schnell wie möglich. Nach Burghardt [Bur95] sind folgende Möglichkeiten der **Terminbeschleunigung** betrachtenswert:

- Paralleles Durchführen von Aufgaben,
- veränderter Mitarbeitereinsatz,
- Aufstocken der vorgesehenen Personalkapazität, (zeitlich begrenztes) Ansetzen von Überstunden,
- Vergabe von Aufgaben an Unterauftragnehmer (intern, extern),
- Kaufen von Entwicklungsteilen statt Eigenentwicklung ("make or buy"),
- Verbessern der Qualifikation des einzusetzenden Personals,
- sinnvolles Beschränken der Leistung des geplanten Produkts durch Wertanalyse (WA) usw.

Beachten Sie bitte auch die Hinweise zur Aufwandsschätzung im Abschnitt 3.3 Planungstechniken.

3.2.5 Kostenplanung

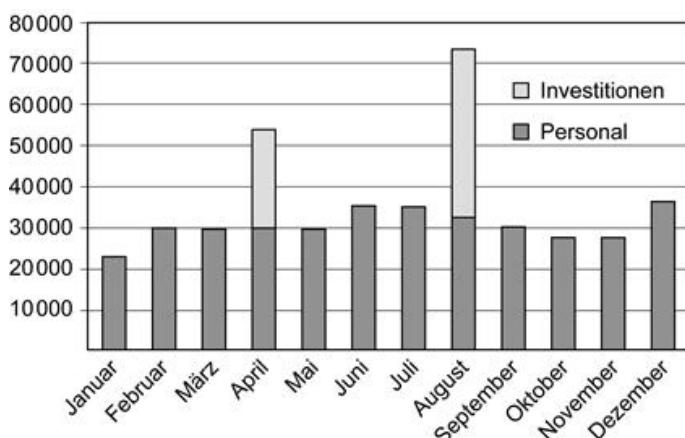
Die Kostenplanung geht Hand in Hand mit der Terminplanung. Für die Planung der Kosten müssen diese kalkuliert werden. Diesen Schritt bezeichnet man in der Kostenrechnung auch als **Vorkalkulation** (VORKA). Der Detaillierungsgrad und der betrachtete Zeitraum hängen auch wieder von der Prozessgestaltung ab. Während der Projektlaufzeit wird die **Mitkalkulation** (MITKA) durchgeführt, damit Plan- und Istwerte verglichen werden können. Ein Projekt sollte immer auch ausgewertet werden und deshalb eine **Nachkalkulation** (NACHKA) durchgeführt werden.



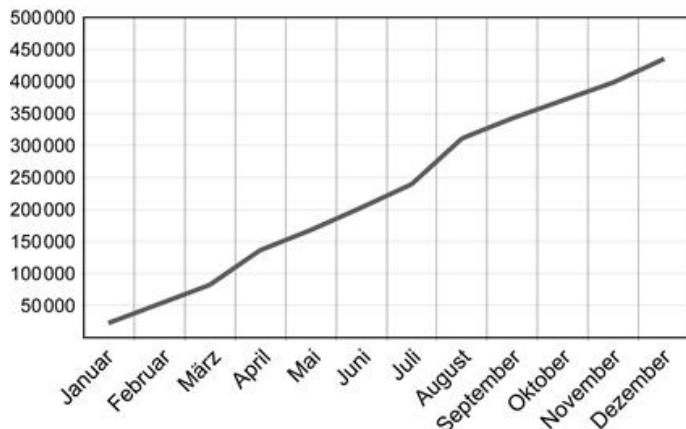
Für eine möglichst hohe Transparenz sollten mehrere **Kostenstellen** (vgl. Kostenstruktur weiter oben) eingeführt werden. Die zu berücksichtigenden **Kostenarten** hängen vom konkreten Projekt ab, umfassen aber im Allgemeinen die folgenden Positionen:

- Personalkosten = Anzahl Mitarbeiter ▪ Kostensatz ▪ Zeiteinheit,
- andere Ressourcen (externe Rechenleistungen),
- Investitionskosten,
- Verbrauchsmittel,
- Dienstreisen,
- Weiterbildung.

Mieten, Pachten usw. werden meistens über Zuschlagssätze (**Zuschlagskalkulation**) berücksichtigt. Diese kommen aus der internen Kostenrechnung (Kostenstellenrechnung). Daraus entsteht ein **Kostenplan**, der auch grafisch dargestellt werden kann (vgl. Abbildung 3.2.5.2 und Abbildung 3.2.5.3).



■ Abbildung 3.2.5.2: Monatlicher Kostenplan, grafisch



■ Abbildung 3.2.5.3: Kostensummenlinie

Im Anhang 7.8 wird ein Beispiel für einen Projektplan vorgestellt, der nach dem extreme Programming entwickelt wurde. Bei dem Projekt selbst handelt es sich um die Entwicklung einer Komponente für die Online-Recherche einer Bibliothek.

3.2.6 Aufgaben zur Planungsreihenfolge



Aufgabe

■ Aufgabe 3.2.6-1

Der Produktivanteil PA in Ihrer Abteilung beträgt 83%. Für ein neu aufzulegendes Softwareprojekt wurde der Aufwand mit 70 MannMonaten (MM) bestimmt. Ihnen stehen 4 Mitarbeiter zur Verfügung. Bestimmen Sie die Dauer des Projekts in Tagen, gehen Sie dabei von einem MM mit 22 Arbeitstagen aus!



Aufgabe

■ Aufgabe 3.2.6-2

Bringen Sie die folgenden Begriffe in die richtige Reihenfolge. Gehen Sie dabei von dem möglichen zeitlichen Nacheinander der Planungen aus!

- A) Terminplanung
- B) Einsatzmittelplanung
- C) Produktstrukturplanung
- D) Kostenplanung
- E) Ablaufplanung



Aufgabe

Aufgabe 3.2.6-3

Im Konfigurationsmanagement werden Umgebungen definiert, in denen Softwareeinheiten bearbeitet oder verwaltet werden. In welcher Umgebung werden die Softwareeinheiten registriert, versioniert und verteilt?

- A) Releaseumgebung
- B) Testumgebung
- C) Referenzumgebung
- D) Arbeitsumgebung



Aufgabe

Aufgabe 3.2.6-4

Die folgenden Begriffe gehören zu einem komplexen System. Ordnen Sie die Begriffe bestimmten Abstraktionsebenen zu! Zeichnen Sie dazu ein Hierarchiegramm!

Suche nach Titel – Bibliotheksbetrieb – Bestandsverwaltung – Suche nach Verfasser – Katalogisierung – Ausleihe – Beschaffung – Nutzerverwaltung – Personalverwaltung – Mahnungen – Aufnahme in den Bestand – Bücher aussondern – Recherche



Gliederung

3.3 Planungstechniken

3.3 Planungstechniken

3.3.1 Probleme der Aufwandsschätzung

3.3.2 FP-Analyse

3.3.3 CoCoMo-II

3.3.4 Netzplantechnik

3.3.5 Balkendiagramme

3.3.6 Aufgaben zu den Planungstechniken

Planungstechniken sind methodische Werkzeuge für den Projektleiter. Sie beruhen auf erprobten Vorgehens- und Darstellungsweisen bei der Planung eines Projekts. In diesem Abschnitt werden die häufig verwendeten Planungstechniken

- Function-Point-Analyse,
- CoCoMo-II,
- Netzpläne und

- Balkendiagramme

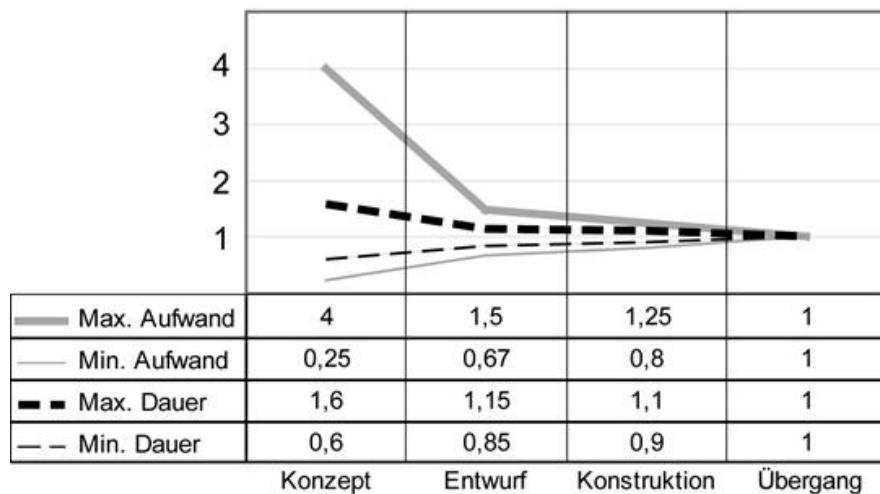
näher vorgestellt.

3.3.1 Probleme der Aufwandsschätzung

Ausgangspunkt für die Bestimmung von Terminen oder Kosten sowie der Anzahl der benötigten Mitarbeiter und anderer Ressourcen ist der Aufwand in MannMonaten (MM) für die zu realisierenden Arbeitspakete (vgl. Abschnitt Planungsreihenfolge). Der Aufwand muss geschätzt werden.

Die Aufwandsschätzung (Schätzung der Personen- oder MannMonate) stellt ein großes Problem dar. Warum?

- Zu Beginn eines Projekts kennen Sie noch nicht alle Aufgaben und die Mitarbeiter im Detail. Deshalb sind die Abweichungen des geschätzten minimalen und maximalen Aufwands in der Konzeptionsphase besonders groß (vgl. Abbildung 3.3.1.1). Erst mit zunehmender Projektlaufzeit werden die Schätzungen besser.



- Bei einem hohen Innovationsgrad ist ein Vergleich mit anderen Projekten schwierig.
- Randbedingungen der Projektentwicklung wurden vernachlässigt, z.B.: Was gehört alles zur Programmierung – nur das Programm oder auch eine Dokumentation? Welche Programmiersprache? Wer programmiert? Gibt es Kenntnisse aus dem Anwendungsgebiet? Sind Ausfallzeiten bekannt? Welche Tools und Bibliotheken werden verwendet? Ändern sich diese Randbedingungen, so ändert sich auch die Schätzung.

- Die Leistungsfähigkeit des eingesetzten Personals ist nicht bekannt. Aus Erfahrungen weiß man, dass die Unterschiede gravierend sein können.
- Die verlangten Vorgaben, z.B. hinsichtlich Laufzeit oder Robustheit, Effizienz oder Fertigstellungstermin, werden unterschätzt. Solche Vorgaben sollten gut überlegt werden. Bei manchen Programmtypen lässt sich Effektivität nur mit sehr hohem Aufwand realisieren (viele Ein- und Ausgaben). Allgemeingültige Lösungen sind teurer als spezielle Lösungen.

Aus den oben genannten Gründen gibt es kein Verfahren, das Ihnen eine genaue Schätzung des Aufwands ermöglicht. Die folgenden formalisierten Schätzverfahren beruhen im Wesentlichen auf **Erfahrungen**, die bei der Abwicklung von Softwareprojekten gemacht wurden.

Die Aufwandsschätzung sollte in zwei Schritten ablaufen. Zum einen die **Schätzung der Größe eines Arbeitspaketes** und zweitens die **Berechnung des Aufwands** in MannMonaten aus der ermittelten Größenangabe. Das Problem der Bestimmung des Aufwandes wird also vereinfacht und in eine Schätzung der "Größe" umgewandelt. Als Maßzahl (Metrik) für die Größe oder den Umfang dienen SLOC (Source Lines of Code) bzw. Funktionspunkte (FP, Function Points), vgl. folgenden Abschnitt.

Diese werden in eine parametrische Schätzgleichung eingesetzt bzw. einer grafischen Auswertung zugeführt, um daraus den Aufwand zu bestimmen. Die Parameter der Schätzgleichungen bzw. der Verlauf einer entsprechenden Kurve werden aus einer Vielzahl von ausgewählten Projekten gewonnen (Analogiemethode). Aus der Literatur sind zwei Verfahren sehr bekannt: zum einen das Function-Point-Verfahren (FP-Analyse, FPA) und das Constructive Cost Model II-Verfahren (CoCoMo II), die im Folgenden vorgestellt werden.

3.3.2 FP-Analyse

Nach dem Entwurf ISO (International Organization for Standardization)/IEC (International Electrotechnical Commission) 14143 dienen die funktionalen Anforderungen aus Sicht des Anwenders als Grundlage für die Berechnung der Function Points. Dabei ergibt sich die funktionale Größe einer Softwareeinheit aus den sogenannten Basis-Funktionskomponenten (BFC – Basic Function Component). Hierzu wird jeder Prozess, der durch einen Anwender mit der Softwareeinheit realisiert werden kann, auf die darin enthaltenen BFC untersucht. Grundsätzlich ist es nach ISO/IEC 14143 möglich, für unterschiedliche Messmethoden unterschiedliche BFC zu definieren. Die Function-Point-Analyse (FPA) nach A. Albrecht kennt 5 BFC:

- Externe Eingaben

- Externe Ausgaben
- Externe Abfragen
- Interne Geschäftsentitäten
- Externe Geschäftsentitäten

Es werden alle Elementarprozesse gezählt. Dazu werden die einzelnen Prozesse strukturell auf die vorhandenen Komponenten Eingaben, Ausgaben, Abfragen, usw. untersucht.

BFCs werden anhand einer Bewertungsmatrix in Unadjusted Function Points (UFP) umgerechnet. Für jede BFC werden dafür nach Albrecht zwei Ausprägungen betrachtet. Abbildung 3.3.2.2 zeigt diese Ausprägungen am Beispiel der externen Eingaben. Die Bewertung dieser Ausprägungen lässt dann eine Einteilung in "einfach", "mittel" oder "komplex" zu. Daraus ergibt sich die Möglichkeit, Unadjusted Function Points zu vergeben.

Abbildung 3.3.2.1: kVASy®-Maske Kaufmännische Kundendaten

In Abbildung 3.3.2.1 wird eine Maske betrachtet, die mehrere Elementarprozesse, wie zum Beispiel Anzeigen oder Ändern, enthält. Jeder dieser Prozesse muss einzeln untersucht werden. In diesem Beispiel wird nur der obere Bereich der Maske betrachtet.

Für den Elementarprozess Anzeigen wird eine externe Eingabe gezählt. Die oben hell schraffierten Felder sind Eingabefelder und somit zu zählen. Als Entitäten werden zwei Kundentypen betrachtet. Anhand der Bewertungsmatrix nach Albrecht erhält man für diese externe Eingabe 4 Unadjusted Function Points.

Externe Eingabe	Anzahl Datenelementtypen <= 4	Anzahl Datenelementtypen > 4, und <= 15	Anzahl Datenelementtypen > 15
Anzahl benutzter Entitätstypen <= 1	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Einfach</pre> <p>3 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Einfach</pre> <p>3 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:0.035cm solid #808080; padding-left:0.049cm;"> Mittel</pre> <p>4 UFP</p>
Anzahl benutzter Entitätstypen = 2	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Einfach</pre> <p>3 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Mittel</pre> <p>4 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:0.035cm solid #808080; padding-left:0.049cm;"> Komplex</pre> <p>6 UFP</p>
Anzahl benutzter Entitätstypen > 2	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Mittel</pre> <p>4 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:none; padding-left:0.049cm;"> Komplex</pre> <p>6 UFP</p>	<pre style="border-top:none; border-bottom:0.035cm solid #808080; border-left:0.035cm solid #808080; border-right:0.035cm solid #808080; padding-left:0.049cm;"> Komplex</pre> <p>6 UFP</p>

			6 UFP
--	--	--	-------

■ Tabelle 3.3.2.2: Bewertungsmatrix Externe Eingabe

Sind alle Elementarprozesse und deren BFCs bewertet, können die Unadjusted Function Points für die Funktionalität "Kaufmännische Kundendaten" berechnet werden. Die UFP sind unabhängig von äußeren Einflüssen wie dem Entwicklungsumfeld, den Qualitätsanforderungen oder der Technik. Der Vergleich verschiedener Projekte erfordert die Betrachtung dieser Einflussfaktoren, da sie den Aufwand maßgeblich verändern können.

Dazu werden die UFP in sogenannte Adjusted Function Points (AFP) umgerechnet. Auch hierfür gibt es verschiedene Ansätze. Nach Albrecht werden diese Einflüsse durch den Value Added Factor (VAF) berücksichtigt, sie werden mit Hilfe von 14 Merkmalen berechnet. Jedes Merkmal kann 0 bis 5 Punkte in die Bewertung einbringen. Die Summe dieser Merkmale ergibt den Wert für TDI – Total Degree of Influence. VAF bestimmt sich dann zu:

$$VAF = (TDI \cdot 0,01) + 0,65$$

AFP ergibt sich dann aus:

$$AFP = UFP \cdot VAF$$

Die Größenangabe FP zum Arbeitspaket kann mit dem **Aufwand in MannMonaten** in Verbindung gebracht werden. Das erfolgt mit Hilfe von alten Projekten (**Analogiemethode**). Man braucht dazu die Ist-Werte von Projekten, die mit dem Function-Point-Verfahren geschätzt wurden.

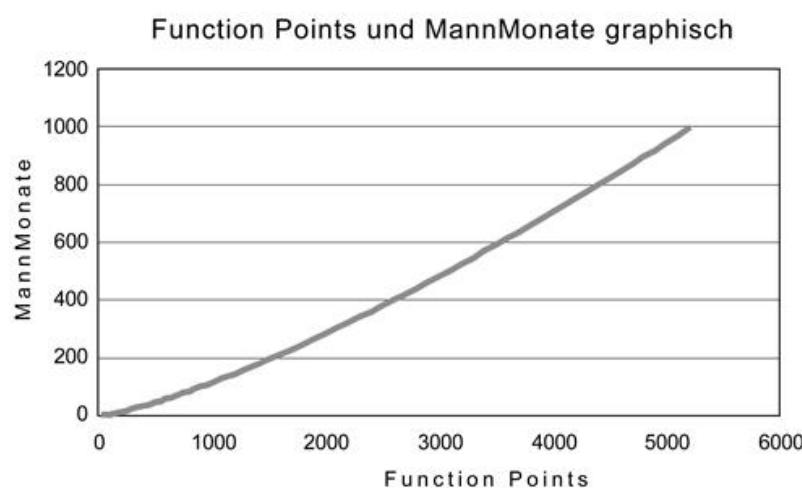
Nach Hürten [Hür99] benötigt man mindestens 20-25 Projekte, um eine hinreichend abgesicherte Aufwandskurve zu erhalten. Es ist darauf zu achten, dass die Projekte dabei aus dem gleichen Entwicklungsumfeld kommen. Entsprechend groß sind die Abweichungen, wenn fremde Kurven verwendet werden. Will man solche Kurven überhaupt nutzen, sollte man die Quelle ausreichend kennen.

Die folgende Tabelle zeigt ein Beispiel mit einem exponentiellen Trend.

FP	MM	FP	MM
50	2,3	1100	129,6
100	5,6	1200	145,2
150	9,5	1300	161,3

200	13,9	1400	177,7
250	18,6	1500	194,6
300	23,6	1600	211,7
350	28,9	1700	229,3
400	34,4	1800	247,1
450	40,1	1900	265,3
500	46,1	2000	283,7
550	52,2	2100	302,4
600	58,5	2200	321,5
650	65,0	2300	340,7
700	71,6	2400	360,3
750	78,4	2500	380,1
800	85,3	2600	400,1
850	92,4	2700	420,4
900	99,6	2800	441,0
950	106,9	2900	461,7
1000	114,4	3000	482,7

■ Tabelle 3.3.2.3: Function Point Methode, Tabelle zur Berechnung der MannMonate



■ Abbildung 3.3.2.4: Function Point Methode, Grafische Darstellung aus der Tabelle 3.3.2.3

Die Tabelle ermöglicht Ihnen, mit den berechneten FPs dann den Aufwand zu schätzen. Haben Sie für Ihr Arbeitspaket 1600 FPs berechnet, lässt sich daraus ein Aufwand von 211,7 MannMonaten ermitteln. Jedes Projekt, das von Ihnen realisiert wurde, sollte nach Projektabschluss ausgewertet und zur Verbesserung des Kurvenverlaufs genutzt werden (vgl. Abschnitt 5.2.2 Erfahrungssicherung).

3.3.3 CoCoMo-II

Im Folgenden wird ein formalisiertes Schätzverfahren vorgestellt, das auf die geschätzten SLOC (Source Lines of Code) zurückgreift. Dieses Verfahren ist unter dem Namen **CoCoMo (Constructive Cost Model)-Verfahren** bekannt.

CoCoMo II ist eine Entwicklung von Barry Boehm [Boe00]; es ist eine Kombination aus parametrischer Schätzgleichung und Gewichtungsmethode. Ausgehend von der geschätzten Anzahl der Anweisungen wird der Aufwand unter Berücksichtigung von **Qualitätszielen** und **Produktivitätsfaktoren** ermittelt.

Die Grundgleichung für die Bestimmung des Aufwands ist:

$$MM = A \cdot \text{Size}^E \cdot \prod_{i=1}^n EM_i$$

wobei

$$E = B + 0,01 \cdot \sum_{j=1}^5 SF_j$$

ist. (**Formel 2-3**)

Der entscheidende Faktor ist die **Größe (Size)** - vgl. Bestimmung der Größe weiter unten). Die Faktoren A und B wurden nach [Boe00] wie folgt bestimmt: A=2,94 und B=0,91. Diese Faktoren sind kalibrierbare Koeffizienten, die durch den Anwender verändert werden können. Das setzt eine Kalibrierungsphase voraus, in der Projekte ausgewertet werden müssen. Bei der oben getroffenen Annahme von A und B hängt die Berechnung der MM noch von den Größen EM und SF ab. EM steht für "Effort Multipliers". Die Anzahl variiert zwischen n=16 für weit entwickelte Architektur-Modelle bzw. n=6 für frühe Design-Modelle. In [Boe00] werden Tabellen für deren Bestimmung gegeben. Mit dieser Einflussgröße wird versucht, den Projektcharakter zu erfassen (Produktkomplexität, Wiederverwendbarkeit, Dokumentationsumfang usw.). Die einzelnen Faktoren variieren in einem Bereich zwischen 0,7 und 3. Wobei der Nominalwert bei 1 liegt und eine "normale" Ausprägung des Faktors kennzeichnet (z.B. Wiederverwendung des Quellcodes nur im eigenen Projekt).

SF steht für "Scale Factor". Er bestimmt sich aus der Summe von 5 Einflussgrößen. Diese berücksichtigen die Rentabilität der Softwareentwicklung. Der Faktor beeinflusst E, den Exponenten der Größe (Size), vgl. [Boe00].

"If $E=1,0$ the economies and diseconomies of scale are in balance. This linear model is often used for cost estimation of small projects. If $E>1,0$ the project exhibits diseconomies of scale. This is generally because of two main factors: growth of interpersonal communications overhead and growth of large-system integration overhead." [Boe00]

Barry Boehm gibt auch eine Gleichung für die Bestimmung der notwendigen Entwicklungszeit (TDEV – time to develop):

$$TDEV = C \cdot MM^F$$

wobei

$$F = D + 0,2 \cdot (E - B)$$

ist. (**Formel 2-4**)

In [Boe00] wird $C=3,67$ und $D=0,28$ gesetzt. B und E sind aus der Bestimmung des Aufwandes in MM bekannt.

Bestimmung der Größe (Size)

Ausgangspunkt des CoCoMo-Verfahrens sind die geschätzten "thousands of source lines of code (KSLOC)". Sie enthalten keine Kommentar- oder Testzeilen. Von dieser Regel gibt es Ausnahmen, die Boehm wie folgt beschreibt: "However, if these are developed with the same care as delivered software, with their own reviews, test plans, documentation, etc., then they should be counted."

Die Schätzung wird besonders dann sehr kompliziert, wenn in Projekten unterschiedliche Programmiersprachen zum Einsatz kommen. Das Software Engineering Institute (SEI) definiert eine Checkliste, um die richtige Anzahl der KSLOC zu bestimmen. Mehr dazu können Sie in [Boe00] nachlesen.

Eine andere Methode ist die Nutzung der Unadjusted Function Points (UFP) und deren Umrechnung in KSLOC. Diese Vorgehensweise ist besonders in frühen Projektstadien zu empfehlen. Für die Umrechnung der UFP in KSLOC gibt es Umrechnungstabellen.

In der Tabelle 3.3.3.1 ist ein Auszug aus einer Tabelle in [Boe00] gegeben.

Language	SLOC pro UFP
Visual Basic 5.0	29
C++	55

Java	53
HTML 3.0	15

■ Tabelle 3.3.3.1: Umrechnungsverhältnis von UFP in SLOC nach [Boe00]

Ein weiteres Problem bei der Bestimmung der Größe ist die Berücksichtigung von wiederverwendetem Code aus Bibliotheken. Boehm unterscheidet hier zwischen "reused" Code (auch Black Box Code) und "adapted" Code (auch White Box Code). White Box Code wird modifiziert, während Black Box Code unverändert übernommen wird. In [Boe00] werden einige Ansätze für die Berücksichtigung dieser Werte vorgestellt.

Weiteren Einfluss auf die Anzahl der eingehenden KSLOC in die Formel (2-3) haben sich ändernde Anforderungen (Requirements Evolution and Volatility – REVL), Software Reengineering und Software-Wartung (Maintenance). Für deren Berücksichtigung werden in [Boe00] Hinweise gegeben.

Anwendung des CoCoMo-II-Verfahrens

Genaue Schätzungen mit dem CoCoMo-II-Verfahren erfordern einen Aufwand und eine Menge Erfahrung. Der größte Anspruch entsteht bei der Schätzung der neuen SLOC. Hier müssen die Erfahrungen aus früheren Projekten gründlich ausgewertet werden.

Die Einordnung des Projektes anhand der Projektmerkmale zur Bestimmung von SF und EM wird durch gut handhabbare Tabellen in [Boe00] sehr gut unterstützt.

Die erzielten Schätzungen haben eine hohe Qualität, da Barry Boehm seine Ansätze aus einer umfangreichen Erfahrungsdatenbank gewonnen hat.

Boehm gibt ein Rechenbeispiel für ein durchschnittliches Projekt mit 100 KSLOC. Für dieses durchschnittliche Projekt wurden alle Effort Multipliers (EM) mit 1 angenommen. E wird 1,15 gesetzt und berücksichtigt damit ein durchschnittliches, aber größeres Projekt. Damit ergibt sich der Aufwand zu

$$MM = A \cdot \text{Size}^E \cdot \prod_{i=1}^n EM_i = 2,94 \cdot (100)^{1,15} = 586,61$$

Daraus kann die Entwicklungszeit

$$TDEV = C \cdot MM^F = 3,67 \cdot (586,61)^{0,28+0,2(1,15-0,91)}$$

$$TDEV = 3,67 \cdot (586,61)^{0,328} = \underline{\underline{29,7}} \text{ Monate}$$

bestimmt werden.

Auswertung der gewonnenen Ergebnisse

Das Ergebnis der Aufwandsschätzung ist der Aufwand für die betrachteten Arbeitspakete. Der Zusammenhang zwischen Aufwand, Dauer und Personal geht aus der uns schon bekannten Formel 2-2 (vgl. Abschnitt 3.2.3 Einsatzmittelplanung)

$$D = \frac{A}{T} \cdot \frac{1}{PA}$$

- D..... Bedarf an Mitarbeitern in Anzahl Mitarbeiter (MA)
- A..... Aufwand in MannMonaten (MM)
- T..... Dauer in Monaten (M)
- PA..... Produktivanteil

hervor. Diese Gleichung lässt sich mit Hilfe der Ergebnisse aus dem CoCoMo-II-Verfahren lösen. Das CoCoMo-II-Verfahren liefert den Aufwand und TDEV.

Um das zu verdeutlichen, greifen wir das Beispiel zum CoCoMo-II-Verfahren von Boehm auf.

Der Aufwand wurde mit 586,61 MannMonaten berechnet. Die Entwicklungszeit wurde mit 29,7 Monaten bestimmt. Daraus lässt sich mit (2-2) der Personalbedarf berechnen:

$$D = A/T * 1/PA = 586,61/29,7 * 1/0,85 = 23,23$$

Das Function-Point-Verfahren liefert nur den Aufwand. Entweder es wird die Dauer festgelegt (bei vorgegebenem Endtermin) oder die Anzahl der Mitarbeiter ist begrenzt und die Dauer ergibt sich dann aus:

$$T = \frac{A}{D} \cdot \frac{1}{PA}$$

Die obige Formel lässt den Schluss zu, dass die Dauer für ein Arbeitspaket beliebig verkürzt werden kann, indem die Anzahl der Mitarbeiter erhöht wird. Hier wird aber nicht berücksichtigt, dass Aufgaben nicht beliebig teilbar sind und der Aufwand durch zusätzliche Kommunikation vergrößert wird. Den richtigen Mitarbeitereinsatz festzulegen, ist Aufgabe des Projektleiters und letztendlich eine Erfahrungssache.

In vielen Fällen sind die betrachteten Arbeitspakete (APs) vernetzt, d.h. zur Erledigung eines APs müssen beispielsweise zwei andere APs erledigt sein. Diese können aber erst beginnen, wenn ein weiteres AP abgeschlossen wurde. Will man die Gesamtdauer aller Arbeitspakete ermitteln, entsteht eine aufwendige Berechnung. Solche vernetzten Probleme können mit der **Netzplantechnik** gelöst werden.

3.3.4 Netzplantechnik

Die Netzplantechnik – auch Netzwerkanalyse oder Netzwerktechnik genannt – entwickelte sich aus Überlegungen von Planern im Rahmen von Großprojekten. Erste Verfahren etablierten sich in der zweiten Hälfte der 50er Jahre des 20. Jahrhunderts in den USA und in Frankreich.

Der Planungsprozess wird durch die Netzplantechnik erheblich unterstützt und erleichtert, wobei sie die wohl raffinierteste Form der Terminplanung darstellt. Die Netzplantechnik kann überall dort angewendet werden, wo große Projekte zu realisieren sind, wie zum Beispiel bei Softwareprojekten, Industrieprojekten, im Anlagenbau, in Forschung und Entwicklung.

Nach DIN 69 900 wird der Begriff **Netzplan** wie folgt definiert:

Der Netzplan ist die grafische Darstellung von Ablaufstrukturen, die die logische und zeitliche Aufeinanderfolge von Vorgängen veranschaulichen.

Funktionale Elemente der Netzplantechnik sind:

- **Ereignisse** – definierte Zustände im Projektablauf
- **Vorgänge** - Ablaufelemente mit Anfang und Ende; sie repräsentieren Arbeitspakete
- **Anordnungsbeziehungen** – personelle, fachliche, terminliche Abhängigkeiten zwischen den Vorgängen bzw. Ereignissen

Netzplanarten

Als die wichtigsten und bekanntesten Basismethoden der Netzplantechnik werden in der Literatur folgende genannt:

- Ereignis-Knoten-Netzplan (EKN) bzw. Program Evaluation and Review Technique (PERT)
- Vorgangs-Pfeil-Netzplan (VPN) bzw. Critical Path Method (CPM)
- Vorgangs-Knoten-Netzplan (VKN) bzw. Metra Potential Methode (MPM)

Am häufigsten wird die **Vorgangs-Knoten-Netzplantechnik** verwendet. Die folgenden Aussagen beziehen sich auf diese Technik.

Ein **Vorgangsknoten** enthält folgende Informationen:

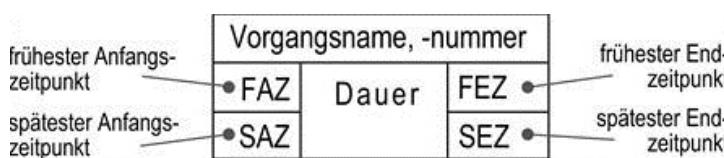


Abbildung 3.3.4.1: Netzplantechnik, grafische Darstellung eines Knotens

Die **Dauer** gibt die Anzahl der Arbeitstage eines Vorganges an.

Berechnung: **Dauer = FEZ - FAZ**

Regeln zum Aufstellen eines Netzplans:

1. Die Abhängigkeit zwischen zwei Vorgängen wird durch einen Pfeil dargestellt. Die Pfeilrichtung gibt dabei die Reihenfolge der Vorgänge an.
2. Ein Vorgang kann einen oder mehrere Vorgänger haben. Ein Vorgang kann einen oder mehrere Nachfolger haben.
3. Ein Netzplan darf keine Schleifen enthalten.
4. Vom Projektanfang bis zum Projektende muss mindestens ein ununterbrochener Ablauf vorhanden sein. Der Netzplan wird zweimal durchgerechnet. Zunächst werden die FAZs und FEZs vom Projektanfang bis zum Projektende bestimmt (progressiv). Anschließend werden die SEZs und SAZs vom Projektende bis zum Projektanfang berechnet (retrograd).
5. Der Vorgang am Projektanfang beginnt mit einem FAZ von Null. $FEZ = FAZ + Dauer$.
6. Ein Vorgang kann erst beginnen, nachdem sein Vorgänger abgeschlossen ist. Das bedeutet, daß der FEZ eines Vorgangs gleichzeitig der FAZ aller Nachfolger ist (auch **Normalfolge** genannt).
7. Besitzt ein Vorgang mehrere Vorgänger, so entspricht der FAZ des Vorgangs dem spätesten FEZ aller Vorgänger.
8. Der FEZ des Vorgangs am Projektende ist gleichzeitig der SEZ des Projekts. Damit können alle SEZs und SAZs ausgehend vom Projektende berechnet werden ($SAZ = SEZ - Dauer$).
9. Der SAZ eines Vorgangs ist gleichzeitig der SEZ aller Vorgänger des Vorgangs.
10. Besitzen mehrere Vorgänge einen gemeinsamen Vorgänger, so entspricht der SEZ des gemeinsamen Vorgängers dem frühesten SAZ aller Nachfolger.

Es werden auch verfeinerte Netzpläne angefertigt. Bei diesen Netzplänen ist die Regel 6 nicht gegeben, d.h. es liegt keine Normalfolge vor. Die folgenden Varianten können auftreten:

Anfangsfolge (AF): Anfang eines Vorganges hängt vom Anfang des Vorgängers ab

Endfolge (EF): Ende eines Vorganges hängt vom Ende des Vorgängers ab

Sprungfolge (SF): Ende eines Vorganges hängt vom Anfang eines Vorgängers ab

In vielen Fällen gibt es Vorgaben für einzuhaltende Abstände zwischen einzelnen Vorgängen. Besonders verständlich lässt sich das am Beispiel von Bauprojekten erklären. Es darf beispielsweise erst nach einer Wartezeit von drei Wochen ein gegossenes Fun-

dament weiter bebaut werden. Positive Zeitabstände für einen Vorgang bedeuten Warten, negative Zeitabstände bedeuten Überlappung.

Kritischer Vorgang, Kritischer Pfad

Ein Vorgang ist **kritisch**, wenn er keine Zeitreserven (=Puffer) besitzt, d.h. wenn durch eine Verzögerung oder Verschiebung Vorgänger oder Nachfolger oder sogar das Projektende beeinflusst werden.

Die Kette aller kritischen Vorgänge bildet den **kritischen Pfad**. Er ist der zeitlich längste Weg durch das Projekt und bestimmt die Gesamtdauer des Projekts. Kritische Vorgänge dürfen weder verschoben noch ausgedehnt werden. Sie müssen deshalb während der Projektdurchführung besonders beachtet werden.

Die Zeitreserven oder Pufferzeiten eines Vorganges lassen sich wie folgt berechnen:

Der **Gesamtpuffer** (GP) ist die Zeitdifferenz zwischen dem Zeitpunkt, zu dem der Vorgang frühestens anfangen kann und dem Zeitpunkt, zu dem der Vorgang spätestens anfangen muss, ohne dass das Projektende durch Verschiebungen in Gefahr gerät.

Berechnungsformel:

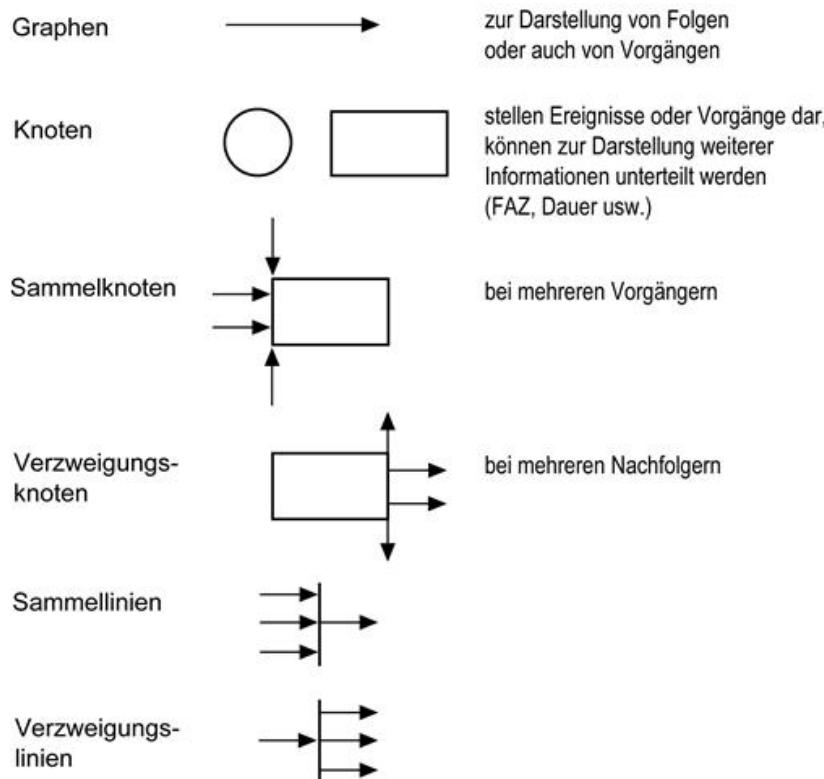
$$GP = SAZ - FAZ$$

Der **freie Puffer** ist die Zeitspanne, um die ein Vorgang ausgedehnt werden kann, ohne den nachfolgenden Vorgang verschieben zu müssen. Als Berechnungsformel gilt:

$$FP(x) = FAZ(x+1) - FEZ(x)$$

x = betrachteter Vorgang, $x+1$ = dessen Nachfolger

Für die grafische Darstellung des Netzplanes können die folgenden Symbole verwendet werden:

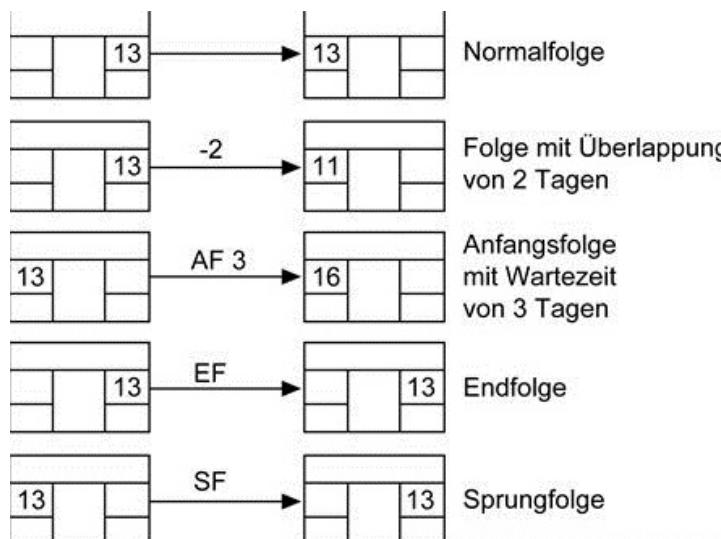


■ Abbildung 3.3.4.2: Netzplantechnik, grafische Elemente

Neben der grafischen Darstellung gibt es auch die tabellarische Darstellung:

Vorgangsnummer	Vorgangsbezeichnung	Dauer	Vorgänger	Nachfolger	Anfangszeitpunkt	Endzeitpunkt

■ Tabelle 3.3.4.3: Netzplantechnik, tabellarische Darstellung



■ Abbildung 3.3.4.4: Netzplantechnik: Besonderheiten bei der Darstellung von Folgen

Im Anhang 7.9 finden Sie ein vollständiges Beispiel zur Netzplantechnik.

Bewertung der Netzplantechnik

Vorteile:

- ermöglicht eine vollständige und konsistente Beschreibung des Projekts,
- stellt zeitliche und sachliche Abhängigkeiten dar,
- der Projektablauf kann transparent dargestellt werden,
- unterschiedliche Auswertungsmöglichkeiten,
- andere Berichte und Dokumente lassen sich ableiten,
- Engpässe hinsichtlich Terminen, Kosten und Ressourcen sind rechtzeitig erkennbar,
- Zuordnung von Mitarbeitern zu einzelnen Aufgaben im Sinne der Verknüpfung sind möglich,
- die Zusammenarbeit der beteiligten Stellen wird gefördert,
- erzwingt systematische Aufgabengliederung des Projekts,
- kann durch Planungstools (z.B. MS Project) unterstützt werden.

Nachteile:

- erfordert tiefgehende Kenntnisse und sehr viel Zeit,
- Änderungsaufwand bei manueller Anwendung sehr groß.

3.3.5 Balkendiagramme

Die Verwendung von Balkendiagrammen lässt sich am besten aus den folgenden Abbildungen ersehen. Ein solches Diagramm zeigt auf der waagerechten Achse die Dauer der Vorgänge und auf der senkrechten Achse die Nummern der Vorgänge.

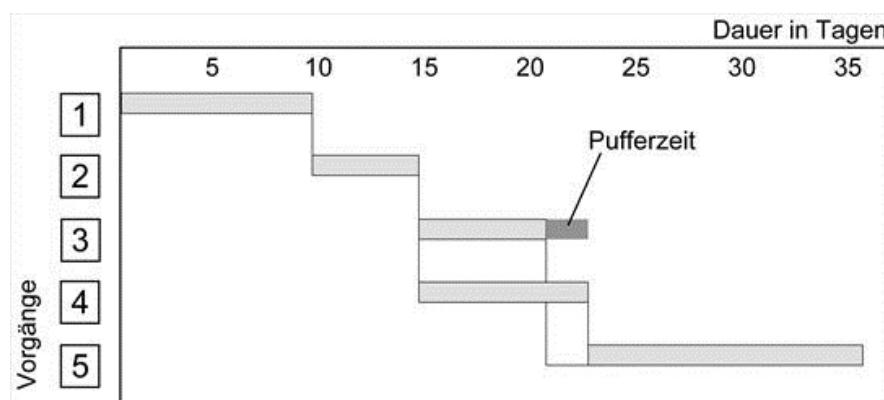
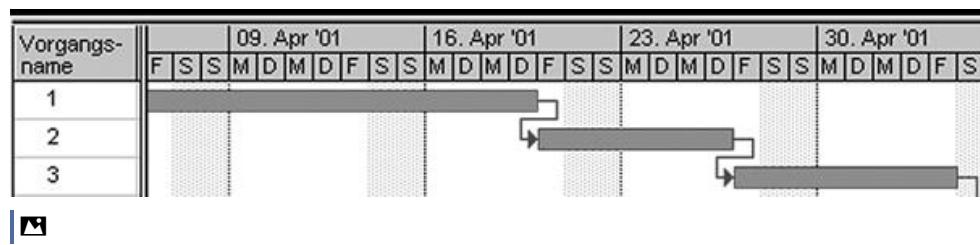


Abbildung 3.3.5.1: Balkendiagramm, Vorgänge über der Dauer

Balkendiagramme können auf der waagerechten Achse auch Kalenderangaben enthalten.



3.3.6 Aufgaben zu den Planungstechniken

Thema: Function Points



Aufgabe 2.2.2.6-1

Adjusted Function Points werden aus Unadjusted Function Points berechnet. Welche Einflüsse werden dabei berücksichtigt?

Thema: CoCoMo-II-Verfahren



Aufgabe 2.2.2.6-2

Ein Unternehmen möchte für ein Java-Projekt eine Schätzung der Personalkosten haben. Der Projektleiter hat dazu die UFP bestimmt. Er kommt auf die Anzahl von 1900. Welche Projektkosten entstehen unter den folgenden Annahmen?

CoCoMo-II-Parameter: alle Effort Multipliers sind =1, E=1,15.

Der Kostensatz für einen Mitarbeiter beträgt 4000 € pro Monat.

Thema: Netzplantechnik

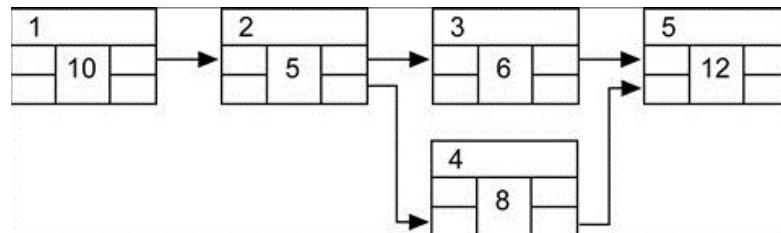


Aufgabe 2.2.2.6-3

a) Vervollständigen Sie den abgebildeten Netzplan!

Vorgangsnummer	Dauer	Vorgänger	Nachfolger
1	10	-	2
2	5	1	3, 4
3	6	2	5

4	8	2	5
5	12	3, 4	-

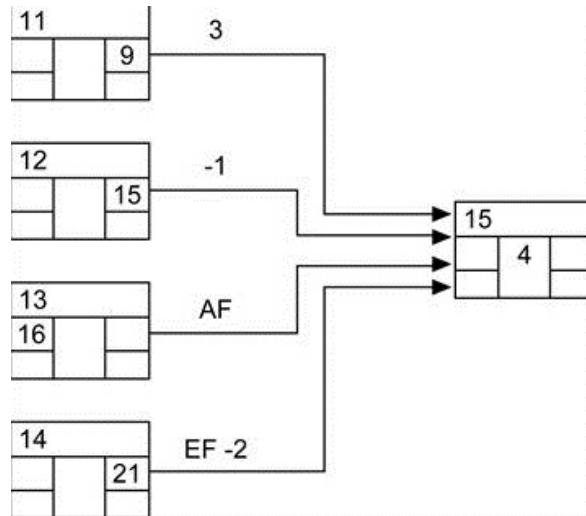


b) Wie lautet der kritische Pfad für den oben abgebildeten Netzplan?



Aufgabe 2.2.6-4

Bestimmen Sie den frühest möglichen Anfangszeitpunkt für Knoten 15 der folgenden Abbildung!



Aufgabe 2.2.6-5

Die Geschäftsleitung der Spielzeugfabrik Storz will aus den Angaben der beteiligten Abteilungen den kritischen Pfad für das Projekt Brettspiel feststellen.

Sie geht von folgender Vorgangsliste aus:

Vorgangs-Nr.	Vorgangsbezeichnung	Dauer in Ta- gen	Vorgänger	Nachfolger
1	Entscheidung	4	-	2, 3
2	Marktanalyse	25	1	13

3	Gesamtentwurf	12	1	4
4	Detailzeichnung	7	3	5
5	Stücklistenerstellung	3	4	6, 7, 8
6	Kalkulation	2	5	9
7	Materialbeschaffung	25	5	9
8	Arbeitsvorbereitung	8	5	9
9	Einzelteifertigung	10	6, 7, 8	10
10	Zusammenbau	1	9	11
11	Kontrolle	1	10	12
12	Lagerbildung	4	11	15
13	Werbevorbereitung	8	2	14
14	Werbekampagne	30	13	15
15	Auswertung	1	12, 14	-

Erstellen Sie den Netzplan für das Projekt Brettspiel und ermitteln Sie den kritischen Pfad!

Thema: Balkendiagramme



Aufgabe

Aufgabe 2.2.2.6-6

Zeichnen Sie für das obige Projekt Brettspiel aus Aufgabe 2.2.2.6-5 ein Balkendiagramm!



Lernziele



4 Projektkontrolle

4 Projektkontrolle

4.1 Voraussetzungen der Projektkontrolle

4.2 Kontrollgrößen und Metriken

Was sollten Sie nach diesem Kapitel wissen?

- Wie werden die Daten für die Kontrolle erhoben?
- Wie ist ein Tätigkeitsbericht aufgebaut?
- Wie werden Reviews durchgeführt?
- Wie können zweckmäßige Checklisten aufgebaut werden?
- Wie werden Termine, Kosten, Sachfortschritt und Qualität kontrolliert?
- Wie kann ein Plan-Ist-Vergleich durchgeführt werden?



4.1 Voraussetzungen der Projektkontrolle

4.1 Voraussetzungen der Projektkontrolle

4.1.1 Erhebung der Ist-Daten

4.1.2 Aufgaben zu den Voraussetzungen der Projektkontrolle

Motivation

Ohne eine Projektkontrolle kann der Projektleiter seine Aufgaben nicht wahrnehmen. Durch die Kontrolle werden Informationen bereitgestellt, die entweder zu Steuerungsmaßnahmen oder im schlimmsten Fall zu Plankorrekturen führen und damit einen geregelten Projektablauf ermöglichen.

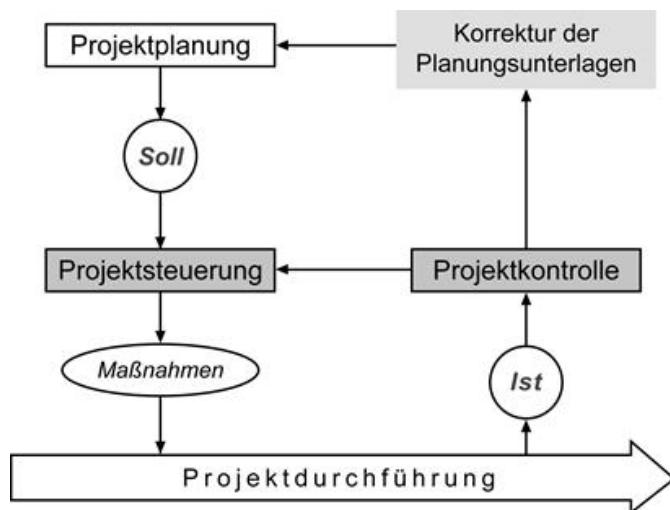


Abbildung 4.1.1: Kontrolle als Element im Regelkreis der Projektplanung

Mit der Kontrolle sind **alle** durch die Planung vorgegebenen Größen zu erfassen. Das trifft sowohl auf die quantifizierbaren Größen wie Termine, Kosten und Aufwand zu als auch für den Leistungsfortschritt sowie qualitative Ziele des Projekts.

Das rechtzeitige Erkennen von Abweichungen im Projektverlauf ist genauso wichtig wie die frühzeitige Reaktion auf Fehler bei der Produktentwicklung (z.B. fehlerhafte Anforderungen). Je eher Abweichungen einer Kontrollgröße, z.B. des Termins, vom geplanten Verlauf entdeckt werden, um so wirksamer sind die folgenden Korrekturmaßnahmen, z.B. die Einbeziehung von zusätzlichen Mitarbeitern. Die folgende Abbildung zeigt die Bedeutung des rechtzeitigen Erkennens von Abweichungen. Es ist zu sehen, dass bei rechtzeitiger Reaktion eine Planänderung vermieden werden kann.



■

Die Ergebnisse der Kontrolle fließen im Allgemeinen in **Berichte** ein. Berichte werden an das übergeordnete Management oder an die eigenen Mitarbeiter verteilt. Sie sind also ein Instrument der Projektsteuerung. Ein Bericht untermauert die Forderung nach mehr Ressourcen oder dient der Motivation der Mitarbeiter, um die Arbeitsleistungen zu erhöhen. Mitarbeitermotivation ist in Softwareprojekten das wichtigste Mittel zur Projektsteuerung und wird im Abschnitt 6 Teammanagement ausführlich besprochen. Diese Steuerungsmaßnahme hat zudem den großen Vorteil, dass das Ziel ohne die Veränderung der Planeckwerte erreicht wird. Mehr Ressourcen verursachen höhere Kosten ("magisches Dreieck") und können die Arbeit in anderen Abteilungen durcheinander bringen.

4.1.1 Erhebung der Ist-Daten



4.1.1 Erhebung der Ist-Daten

4.1.1.1 Tätigkeitsbericht

4.1.1.2 Reviews

Die Erhebung der Ist-Daten hat zwei Aspekte – einen zeitlichen und einen inhaltlichen. Der zeitliche Aspekt kennzeichnet die Erhebungszyklen. Da, wie oben festgestellt, Steuerungsmaßnahmen umso wirksamer sind, je früher Abweichungen festgestellt werden, müssen kleine Zyklen gewählt werden – für bestimmte Erhebungen bieten sich Wochenzyklen an. Inhaltlich wird betrachtet, welche Daten zu erheben sind und wie. Da die Projektkontrolle alle Größen berücksichtigen soll, müssen die Daten so erhoben werden, dass eine Termin-, Kosten-, Aufwands-, Leistungs- und Qualitätskontrolle ermöglicht werden. Folgende Erhebungstechniken haben sich bewährt:

- Tätigkeitsberichte und
- Reviews.

Mit Tätigkeitsberichten werden die Daten für die Termin-, Kosten-, Aufwands- und Leistungskontrolle erhoben. Mit den Reviews werden die Qualitäts- sowie Motivations- und Kommunikationsparameter erhoben.

4.1.1.1 Tätigkeitsbericht

Der **Tätigkeitsbericht** bildet das zentrale Element des Kontrollsystems im Hinblick auf die Erfassung des erfolgten Einsatzes von Ressourcen (Personal, technische Geräte, Infrastruktur). Jeder Mitarbeiter des Projekts füllt dazu wöchentlich ein – in den allermeisten Fällen – computergestütztes Formular aus. Dieses Formular wird automatisch

von einem Zeiterfassungs- oder Projektmanagementsystem bereitgestellt und mit den persönlichen Angaben des Mitarbeiters vorbereitet. Dadurch wird die Arbeit für das Ausfüllen des Formulars minimiert. Die bereitgestellten Einträge sollten

- Angaben zur Person,
- die zugeteilten Aufgabengebiete – in Form von Arbeitspaketen
- und eingeplante sowie zusätzliche Ressourcen

umfassen. Diese Angaben werden aus den Planungsunterlagen entnommen. Das vorbereitete Formular muss nun durch den Mitarbeiter ergänzt werden.

Für die genaue Erfassung des Aufwandes für ein Arbeitspaket trägt der Mitarbeiter seine Arbeitszeiten, gegliedert nach Wochentag, Arbeitspaket und Tätigkeit, seine Ausfallzeiten (je Wochentag) sowie einen Schätzwert für die Nutzungszeit der zugeteilten Ressourcen (je Woche) ein und macht Angaben über die voraussichtlich noch benötigten Stunden für die notwendigen Tätigkeiten in den einzelnen Aufgabenpaketen. Typische Tätigkeiten in einem Arbeitspaket in Softwareprojekten sind beispielsweise das Programmieren und das Testen.

Besonders der letzte Punkt – die "noch benötigten Stunden" – ist für die Einschätzung der erbrachten Leistungen sehr wichtig.

Das Formular sollte übersichtshalber auch den geplanten Anfangs- und Endtermin sowie den geplanten und geleisteten Gesamtaufwand der Arbeitspakete und des Projektes bzw. Teilprojektes ausweisen. Empfehlenswert ist dann das Abfragen der geschätzten Endtermine. Das erleichtert z.B. das spätere Anfertigen einer Meilenstein-trendanalyse.

Formulare auf Papier, die per Hand vollständig auszufüllen sind, haben sich aufgrund des zu hohen Aufwandes kaum bewährt. In Abbildung 4.1.1.1 ist ein Ausschnitt aus einem computergestützten Tätigkeitsbericht [Buh02] zu sehen, der die oben genannten Anforderungen berücksichtigt.

Arbeitszeiten											
Einzel-auftrag	Tätigkeit	Benötigte Zeit in Stunden							noch benötigt (geschätzt)	Bemerkungen	
		Sa	So	Mo	Di	Mi	Do	Fr			
EA005	Szenen erstell.	0	0	8	5	5	2	2	22	90	
EA005	Navigation	0	0	0	3	3	6	6	18	80	
EA005		0	0	0	0	0	0	0	0	0	



Ein Arbeitspaket kann mehrere Tätigkeiten erforderlich machen. Für deren Erfassung werden mehrere Formularzeilen eingerichtet. Die tatsächlich aufgewendete Zeit wird pro Wochentag eingegeben, dabei wird in der Spalte "Summe" automatisch die gesamte für diese Tätigkeit abgerechnete Arbeitszeit aktualisiert. Auf einer anderen Stelle des Formulars (hier nicht sichtbar) ist der Planwert für diese Größe ablesbar. Weiterhin schätzt der Mitarbeiter, wie viel Zeit er noch für die Tätigkeit benötigen wird. Im letzten Feld hat er die Möglichkeit, Bemerkungen anzubringen.

In jedem Projekt treten für die Mitarbeiter auch **ungeplante Aufgaben** auf, z.B. wenn andere Kollegen unterstützt werden. Auch diese Zeiten müssen erfasst werden. Wie man in folgendem Formularausschnitt sieht, geschieht das in der gleichen Weise wie für planmäßige Aufgaben.

Ungeplante Tätigkeiten										
Einzel-auftrag	Tätigkeit	Benötigte Zeit in Stunden							noch benötigt (geschätzt)	Bemerkungen
		Sa	So	Mo	Di	Mi	Do	Fr		
EA002	Azubi einweisen	0	0	2	0	0	0	0	2	0



Zur Erledigung von Aufgaben werden gegebenenfalls **zusätzliche Ressourcen** verwendet. Diese Ressourcen (z.B. externe Rechnerleistungen) können Kosten verursachen und müssen erfasst werden.

Ressourcen				
Einzelaufrag	Ressource		Geschätzte Zeit in Stunden	Bemerkungen
	ID	Name		
EA005	R005	3D Workstation	40	Rendering inklusive.



Das System sollte das Formular schon soweit vorbereiten, dass jede Kombination aus Einzelauftrag und dem Einzelauftrag zugeteilten Ressourcen, die in der Planung berücksichtigt wurden, auftauchen. Der Mitarbeiter schätzt, wie lange er die jeweilige Ressource während der Berichtswoche genutzt hat, und trägt den Wert in der Spalte "Geschätzte Zeit in Stunden" ein.

Im Bereich "**Ausfallzeiten**" gibt der Mitarbeiter ein, für welche Zeit und aus welchen Gründen er in der Berichtswoche nicht für die zugeteilten Arbeitspakete zur Verfügung stand. Dazu wählt er in der Spalte "Ursache" eine der vordefinierten Alternativen und gibt in der gleichen Zeile beim jeweiligen Wochentag seine Ausfallzeit an. Das System

summiert mit jeder Änderung eines Eingabefeldes die Zahlenwerte der Zeile und zeigt sie in der Spalte "Summe" an, um dem Mitarbeiter einen Wochenüberblick zu bieten. Verbale Äußerungen gibt der Mitarbeiter bei Bedarf im Eingabefeld der Spalte "Bemerkungen" ein.

Ausfallzeiten									
Ursache	Sa	So	Mo	Di	Mi	Do	Fr	Summe	Bemerkungen
Krankheit ▼	0	0	0	0	0	8	0	8	



Im Bereich "**Probleme**" gibt der Mitarbeiter Probleme ein, die seiner Meinung nach die Erfüllung seines Einzelauftrages oder aber auch das gesamte (Teil-)Projekt gefährden könnten.

Problem-Überblick	
Bereich	Einzelauftrag ▼
Bezeichnung	EA005 ▼
Beschreibung	Die Ladezeiten für die Shockwave-Datei sind zu hoch. ▲ ▼
Lösungsvorschlag	
Lösung	Reduzierung der Bilder pro Sekunde innerhalb der 3D-Szenen ▲ ▼



Hierbei ist wichtig, dass nicht nur das Problem, sondern auch der **Lösungsansatz** erfasst wird. Damit bekommen Sie als Projektleiter sofort auch Hinweise für die Projektsteuerung. Nachfolgende Maßnahmen haben eine höhere Akzeptanz.

Ein Tätigkeitsbericht ist nur so gut wie sein Wahrheitsgehalt: Mehr als 6 Stunden produktive Tätigkeit sind unrealistisch. Machen Sie Ihren Mitarbeitern klar, dass Sie Tätigkeitsberichte brauchen, um das Projekt zu kontrollieren und nicht die Mitarbeiter!

Mit der Erhebung der beschriebenen Daten können alle wichtigen quantifizierbaren Größen kontrolliert werden – Termin-, Aufwands-, Kosten- und Leistungsgrößen.

Die Qualitäts-, Motivations- und Kommunikationsgrößen können mit Hilfe von Reviews erhoben werden.

4.1.1.2 Reviews

Reviews dienen zur Kontrolle und gehören zu den statischen Prüfverfahren. Sie sind eine echte Kontrollmaßnahme des Projektmanagements zur Aufdeckung von Abweichungen und Einleitung von Steuerungsmaßnahmen. Sie ergänzen die Tätigkeitsberichte, da hier auch nicht quantifizierbare Größen (Qualität, Motivation der Mitarbeiter, usw.) erfasst werden können.

"Bei den Reviews gibt es Veranstaltungen, die die Form einer Präsentation annehmen und dazu dienen, den Kunden über den Projektfortschritt zu informieren. Sind keine Vertreter des Kunden dabei, kann es sich bei dem Review um ein internes Review des Unternehmens handeln, in dem sich das Management von einer Projektgruppe berichten lässt." [Tha00]

Reviews, die ohne Beteiligung des Kunden und des Managements stattfinden, werden häufig als **Peer Reviews** bezeichnet. "Der Ausdruck Peer bezeichnet im Englischen Personen, die hierarchisch auf der gleichen Stufe stehen." [Tha00]

Die Reviews lassen sich also einteilen in:

- Reviews unter Beteiligung des Kunden:
 - Software Requirements Reviews
 - Design Reviews
- Reviews für das eigene Management:
 - internes Review, quasi eine Berichterstattung
- Reviews innerhalb der Entwicklergruppe (**Peer Reviews**):
 - Modellinspektion,
 - Prüfung von Dokumentationen,
 - Codeinspektion,
 - Code Walkthrough,
 - Erhebung von Motivations- und Kommunikationsgrößen

Für die Projektkontrolle haben die Reviews innerhalb der Entwicklergruppe die höchste Bedeutung. **Dokumenten- und Modellinspektionen** haben den Vorteil, dass sie weit vor der Implementierung und dem Test von Software durchgeführt werden können. Wenn Fehler schon nicht vollständig zu vermeiden sind, dann sollten sie doch zumindest möglichst rasch gefunden werden können. Die Kosten für die Fehlerbeseitigung sind dann geringer (vgl. auch Abbildung 2.2.1.1 im Abschnitt 2.2.1 Risikoanalyse).

Die **Codeinspektion** hat die Vorteile der Ursachenanalyse von auftretenden Problemen, die sich beispielsweise in einer verspäteten Fertigstellung äußern, und der Einflussnahme z.B. auf den Programmierstil. Sie sollte im Team durchgeführt werden:

mindestens 2 Leute und der Autor des Programms. Die Länge des Reviews sollte 90 min nicht überschreiten und die Anzahl der LOCs kann zwischen 150 und 250 liegen. Die notwendigen Unterlagen müssen rechtzeitig verteilt werden. Es ist der folgende Ablauf denkbar:

- Vortrag des Programmierers,
- Analyse mit Checkliste,
- Fehlerbesprechung.

Die **Checklisten** müssen vorher erarbeitet werden. Weiter unten werden einige Beispiele vorgestellt. Für die Ausarbeitung der Checklisten der Codeinspektion wurde auf das Buch "Refactoring" von Martin Fowler [*Fow00*] zurückgegriffen. Darin werden mögliche Fehlerquellen aufgezeigt, die man sehr gut in einem Review aufdecken kann.

Die Vorgehensweise bei anderen Artefakten oder bei der Erhebung von Motivations- und Kommunikationsparametern ist ebenso zu organisieren wie für die Codeinspektion.

Wie schätzt der Mitarbeiter die Projektqualität ein?	
Kommunikation zwischen den Projektmitgliedern	
Checkliste	Checkliste fuer Projektmitgliederkommunikation ▾
Bewertung	sehr gut <input checked="" type="radio"/> <input type="radio"/> sehr schlecht
Bemerkung	Gute Kommunikation, geringer Overhead. ▾
Kommunikation mit dem Auftraggeber	
Checkliste	Checkliste fuer Auftraggeberkommunikation ▾
Bewertung	sehr gut <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> sehr schlecht
Bemerkung	Zuständigkeiten oft nicht klar ersichtlich. ▾
Mitarbeitermotivation	
Checkliste	Checkliste fuer Mitarbeitermotivation ▾
Bewertung	sehr gut <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> sehr schlecht
Bemerkung	Projektziele nicht immer klar ersichtlich. ▾



Reviews müssen protokolliert werden. Abbildung 4.1.1.2.1 enthält einen Ausschnitt aus einem computergestützten **Reviewprotokoll** [*Buh02*]. Der Ausschnitt zeigt die Erhebung von Kommunikations- und Motivationsparametern. Die Erhebung beinhaltet auch eine Bewertung für eine spätere mathematische Auswertung, z.B. für die Anfertigung von Nutzwert- und Trendanalysen.

Damit wären wir auch schon bei dem letzten Problem: Wann und in welchen Abständen sollten solche Reviews durchgeführt werden? Das Review von Dokumenten und Quellcode (Artefakte) sollte immer dann stattfinden, wenn eine Iteration oder ein Meilenstein abgeschlossen wurde. Die Erhebung von Motivations- und Kommunikationsparametern sollte periodisch (monatlich) durchgeführt werden. Sie ist ein Bestandteil der Führungstätigkeit.

Im Anhang 7.10 finden Sie zahlreiche Beispiele für Checklisten.

4.1.2 Aufgaben zu den Voraussetzungen der Projektkontrolle

**Aufgabe**

Aufgabe 4.1.2-1

Für die Kontrolle und Erhebung von Softwareprojekten können Tätigkeitsberichte, Tests und Reviews eingesetzt werden. Welche Daten können mit welcher Methode erhoben werden – ordnen Sie die folgenden Begriffe zu!

Review	Tätigkeitsbericht	Test	
Mitarbeitermotivation	?	?	?
Antwortzeitverhalten des Programms	?	?	?
Ausfallzeiten der Mitarbeiter	?	?	?
Einhaltung des Style Guide im Quellcode	?	?	?
Arbeitszeiten der Mitarbeiter	?	?	?
Lastverhalten des Programms	?	?	?
Kommunikation im Team	?	?	?

**Aufgabe**

Aufgabe 4.1.2-2

Was besagt der Begriff Peer Review?

- A) ein Review mit Kundenvertretern
- B) ein Review unter gleichgestellten Mitarbeitern
- C) eine Präsentation der Ergebnisse vor dem Management



Aufgabe 4.1.2-3

Aufgabe

In welchem Zeitintervall sollten Tätigkeitsberichte angefertigt werden?

- A) monatlich
- B) wöchentlich
- C) alle zwei Wochen

**Gliederung**

4.2 Kontrollgrößen und Metriken

4.2 Kontrollgrößen und Metriken

4.2.1 Termine

4.2.2 Kosten und Aufwand

4.2.3 Plan-Ist-Vergleich

4.2.4 Sachfortschritt

4.2.5 Qualität

4.2.6 Aufgaben zu Kontrollgrößen und Metriken

In diesem Abschnitt werden wichtige Größen und Metriken, die bei der Projektkontrolle zum Einsatz kommen, behandelt.

4.2.1 Termine

Bei vielen Projekten ist heute der Termin einer der wichtigsten Eckwerte, den man nicht gerne verschiebt. Das hängt in erster Linie mit der starken Konkurrenz und der damit verbundenen Wettbewerbsintensität zusammen.

Für die **Terminkontrolle** werden die Daten aus den Tätigkeitsberichten der Mitarbeiter ausgewertet. Dazu muss der Mitarbeiter im Tätigkeitsbericht Terminverschiebungen durch eine Schätzung des neuen Endtermins angeben. Auch der Grund oder das Problem müssen mit ausgewertet werden. Das Anzeigen einer Terminverschiebung muss zu einer **Reaktion** führen:

- Reduzierung der Aufgabenstellung oder
- Einsatz von mehr Personal oder
- Veränderung der Pläne.

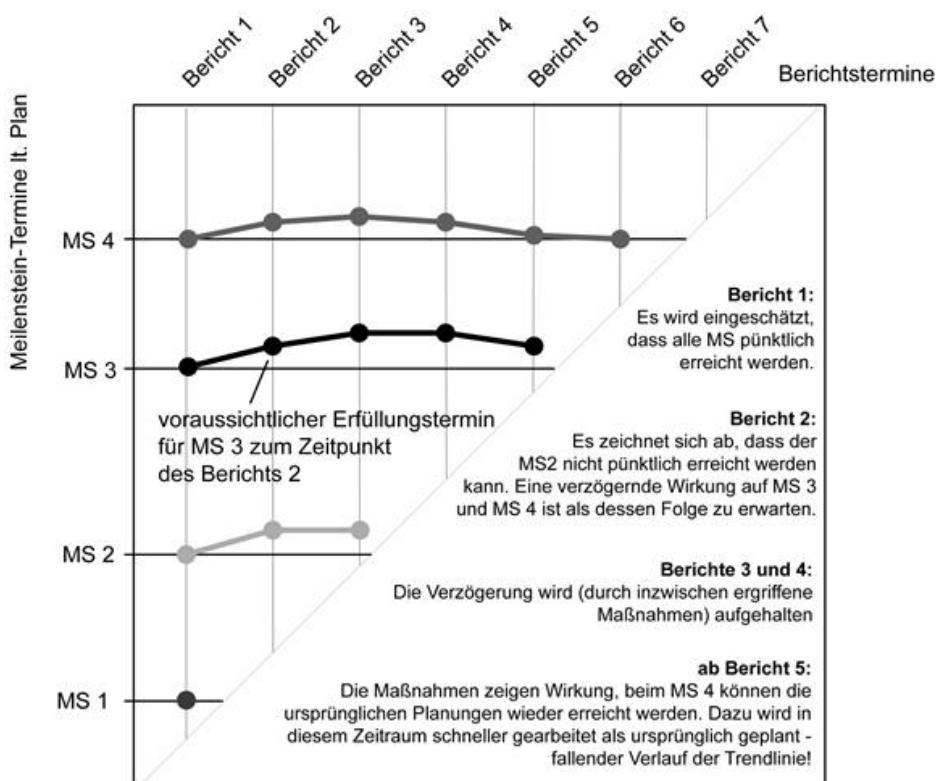
Wegen der oben genannten Konkurrenz wird man in vielen Fällen darüber diskutieren, welche Funktionalitäten weniger wichtig sind und unter welchen Umständen das Release doch pünktlich ausgeliefert werden kann. Es gibt in fast allen Projekten Funktionen, die man zwar gerne hätte, die aber im Hinblick auf die Abwicklung der relevanten

Geschäftsprozesse keine Bedeutung haben. Wählt man diesen Weg, sollten also die anstehenden Aufgaben nach der Wichtigkeit analysiert werden.

Der Einsatz von mehr Personal setzt voraus, dass die Aufgabe auch teilbar ist. Weiterhin muss abgeschätzt werden, ob der Einarbeitungsaufwand und die erforderlichen Hilfestellungen für die neu einzuarbeitenden Kollegen eine Zeiteinsparung zulassen. Von der planerischen Seite ist diese Art der Steuerung meistens mit dann höheren Kosten verbunden, das heißt, dass die Kostenpläne zu überarbeiten sind.

Bei der Veränderung der Terminpläne sind **Abhängigkeiten zwischen den Arbeitsspaketen** zu untersuchen. Im schlimmsten Fall muss der gesamte Netzplan neu erstellt werden. Iterative Vorgehensmodelle haben den Vorteil, dass nur ein relativ kurzer Zeitraum in der Detailplanung zu betrachten ist.

Besondere Aufmerksamkeit sollten Sie den **Meilensteinen** widmen, da diese Punkte im Projekt von besonderer Bedeutung sind. Sie bilden die Voraussetzung für eine Reihe nachfolgender Aktivitäten. In diesem Zusammenhang ist auf **Termintrendanalysen** zu verweisen. Diese bieten sich für das Gesamtprojekt, aber auch für Meilensteine an (**Meilensteintrendanalyse**).



■ Abbildung 4.2.1.1: Meilensteintrendanalyse, normale Verläufe von MS 3 und 4, MS 2 weist einen ansteigenden Verlauf auf

Der normale Verlauf ist dadurch gekennzeichnet, dass es kleine Verschiebungen nach oben und unten gibt. Ein extrem ansteigender Verlauf signalisiert, dass sich der Endtermin verzögert. Hier wurde von viel zu optimistischen Planungen ausgegangen. Der entgegengesetzte Fall zeigt an, dass die Planungen zu pessimistisch sind und der Fertigstellungstermin früher als geplant eintritt. Ein Zick-Zack-Verlauf deutet auf Planungssunsicherheiten hin.

4.2.2 Kosten und Aufwand

Neben den Terminen sind die Kosten eine sehr wichtige Kontrollgröße, die vom übergeordneten Management sehr aufmerksam betrachtet werden. Dazu soll eine Meinung wiedergeben werden, die bei einem Gespräch mit einer Beratungsfirma geäußert wurde:

"Eines der häufigsten Probleme, die wir zur Zeit von EDV-Leitern hören, ist die mangelnde Kontrolle über die Kosten für die Planung und Realisierung bei Datenmigrations- und Integrations-Projekten. Regelmäßig laufen hier die Projekte aus dem Budget, sind nicht innerhalb der geplanten Zeit fertig gestellt oder die Komplexität der Migrationsprojekte wird so hoch, dass sie kaum mehr führbar ist."

Der Aufbau einer zweckmäßigen Kostenkontrolle ist sicherlich nicht einfach, das liegt vor allem an den vielen verschiedenen Kostenarten, die zu betrachten sind und den verschiedenen Orten, wo diese dann auftreten. Es gibt zwei wesentliche Informationsquellen: die Kostenrechnung des Unternehmens – auch externe Kostenrechnung genannt, weil sie außerhalb des Projekts realisiert wird – und die interne Kostenrechnung, die im Projekt selbst realisiert wird.

Bei der **externen Kostenrechnung** können Kostenübersichten aus den Verwaltungsbereichen abgefordert werden. Das Projekt hat im Allgemeinen eine eigene Kostenstelle. Sie sollten mit Blick auf eine spätere Auswertung eine möglichst feine Aufteilung der Kostenarten fordern, z.B.:

- Personalkosten
- Arbeitsplatzkosten (Abschreibungen für Investitionen (Räume, Möbel, Büromaschinen))
- Reisekosten
- Kommunikationskosten (Netzkosten, Telefon, Post)
- Dienstleistungskosten (Netzbetreuung, Instandhaltung, Reinigung)
- Fremdkosten aus externen Aufträgen

Bei diesen Kosten muss man zwischen den direkt dem Projekt zuordenbaren Kosten und den nur indirekt zuordenbaren Kosten unterscheiden. Die Kostenarten, die direkt zugeordnet werden können, hängen von den oft technischen Möglichkeiten einer personen- oder arbeitsplatzbezogenen Erfassung ab. **Reisekosten** können beispielsweise eindeutig Personen und damit einem Projekt zugeordnet werden.

Die **Netzbetreuung** im Unternehmen wird meist zentral als Dienstleistung angeboten. Für die projektbezogene Weiterverrechnung müssen dann Verteilerschlüssel erarbeitet (ausgehandelt) werden. Dazu können nach Burghardt [Bur95] die folgenden Kriterien angesetzt werden:

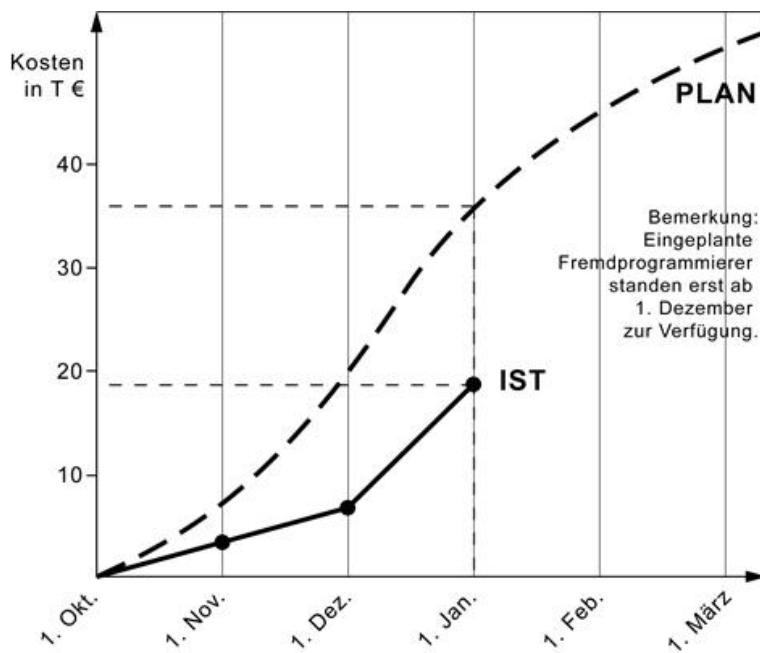
- Anzahl der Installationen
- Anzahl der Benutzer
- Anzahl der Transaktionen
- Anzahl der Terminals
- Umfang des Datenvolumens
- Umlage nach Wertschöpfungsanteilen, usw.

Bei der Vergabe von **Fremdleistungen** sind die einlaufenden Rechnungen zu prüfen. Der Prüfaufwand und die Möglichkeiten, bei Abweichungen rechtzeitig reagieren zu können, hängen im großen Maße auch von der Gestaltung der Verträge ab. Im Abschnitt Projektziele finden Sie ein Beispiel für einen Projektvertrag. Der größte Prüfaufwand entsteht bei der Bewertung der qualitativen Ergebnisse. Die Vergabe von Fremdleistungen setzt voraus, dass Sie den Aufwand für das zu erstellende Produkt einschätzen können. Wie wir aus den Ausführungen im Abschnitt Planungstechniken wissen, ist dies bei Software kein einfaches Unterfangen.

Eine **interne Kostenrechnung** (ohne Fremdleistungen) – **target costing** – erhöht die Aussagefähigkeit der buchhalterischen Kostenübersichten. Die interne Kostenrechnung stützt sich auf die Erfassung der geleisteten Stunden (Aufwand) aus den Tätigkeitsberichten und deren Verrechnung mit den entsprechenden Kostensätzen. Für die Analyse von Kostenüberschreitungen ist diese interne Kostenrechnung und Auswertung sehr wichtig, besonders dann, wenn im Tätigkeitsbericht nach Tätigkeiten und Ausfallzeiten unterschieden wird. Entscheidend dabei ist, dass die Berichte vollständig und ehrlich angefertigt werden. Sie haben damit die Möglichkeit, den Kostenverlauf getrennt nach Produktivzeit und Ausfallzeit der Mitarbeiter darzustellen. Ein überproportionales Ansteigen der Kosten durch Ausfallzeiten kann so erkannt werden, Steuerungsmaßnahmen können eingeleitet werden.

4.2.3 Plan-Ist-Vergleich

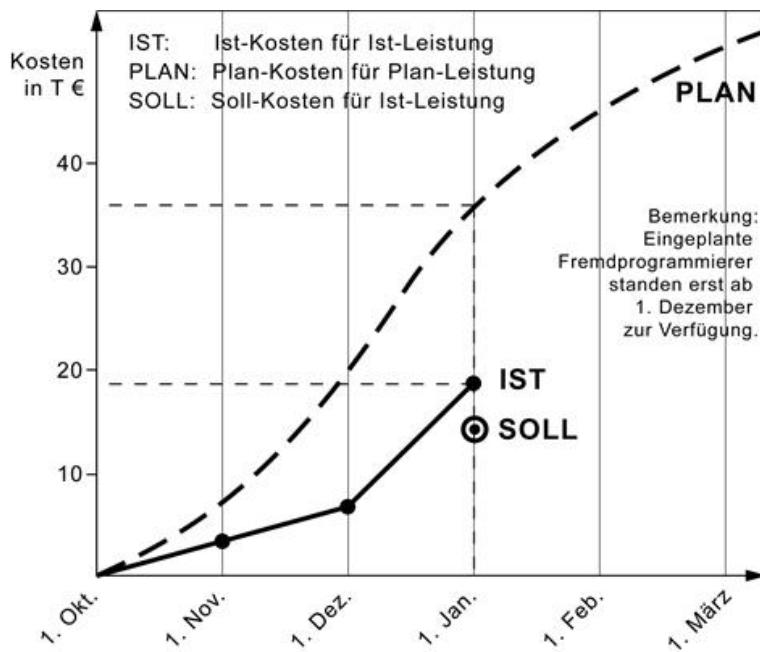
Beim Erstellen eines Plan-Ist-Vergleichs müssen einige Fragestellungen berücksichtigt werden. Vernachlässigt man diese, kann die reine Kostenübersicht zu falschen Schlussfolgerungen führen. Notwendige Steuerungsmaßnahmen bleiben aus. Diese Fragestellungen betreffen in erster Linie den Sachfortschritt und die Termine.



■ Abbildung 4.2.3.1: Kostenvergleichsdiagramm

Quelle: [Kup96]

Die Abbildung 4.2.3.1 zeigt einen typischen Verlauf für den Projektbeginn – die Kosten liegen unterhalb des Plans. Das verleitet zu der Annahme, dass auch am Ende die Kosten zumindest stimmen. Das tritt selten ein – denn wie steht es mit dem zu erbringenden Sachfortschritt? Wie sieht es mit Strafen beim Auftraggeber für ein zu spät geliefertes Programm aus? Dieses Vergleichsdiagramm ist eigentlich nur dann aussagefähig, wenn alle zu erbringenden Leistungen als Fremdleistungen mit festem Budget und eindeutigen Zielvorgaben vereinbart wurden. Es signalisiert aber, dass irgendetwas nicht stimmt in dem betrachteten Projekt. Der eingezeichnete Stichtag in Abbildung 4.2.3.1 ist der 1. Januar. Die eingezeichneten Kosten sind die Ist-Kosten für die Ist-Leistung (fertig gestellte Arbeitspakete zum Stichtag) und die Plan-Kosten für die Plan-Leistung (geplante Arbeitspakete zum Stichtag). Die Aussagekraft des Diagramms kann erhöht werden, wenn man zusätzlich auch die Soll-Kosten für die Ist-Leistung zum Stichtag einzeichnet – siehe Abbildung 4.2.3.2.



Die Differenz zwischen den Ist- und den Soll-Kosten ist dann die leistungsbezogene Kostenabweichung zum Stichtag. In dem bezeichneten Fall liegen die Soll-Kosten unterhalb der Ist-Kosten. Obwohl also die Ist-Kosten unter den Plan-Kosten liegen, haben wir höhere Kosten verursacht, als die erbrachte Leistung es rechtfertigen würde. Das Projekt ist auch noch aus dem Termin, denn auch die erbrachten Leistungen liegen unter den geplanten. Dazu im folgenden Abschnitt weitere Vergleichsmöglichkeiten unter Betrachtung des Sachfortschritts.

4.2.4 Sachfortschritt

Die Kontrolle des Sachfortschritts ist am kompliziertesten, da sie sich nicht auf messbare Größen stützen kann. Es wird in den allermeisten Fällen durch die Mitarbeiter geschätzt, wie viel Aufwand noch für die Erledigung des anstehenden Arbeitspaketes zu leisten ist. Bei kleinen und ergebnisorientierten Arbeitspaketen lässt sich der Projektfortschritt genauer bestimmen. Dafür werden in einer guten Planung die Voraussetzungen gelegt.

Eine mögliche, von vielen aber kritisch betrachtete (vgl. Ausführungen weiter unten) Kenngröße für die Ermittlung des Sachfortschritts ist der **Fertigstellungsgrad**. Er sagt aus, wie viel Prozent der geplanten MannMonate realisiert wurden.

$$FG = A_{fertig} / A_{ges} \text{ (gültig für einen Stichtag)}$$

FG Fertigstellungsgrad

A_{fertig} realisierter Aufwand in MannMonaten

A_{ges} geplanter Gesamtaufwand in MannMonaten

A_{fertig} muss aus den Tätigkeitsberichten ermittelt werden. Je nachdem, ob man nur die fertiggestellten Arbeitspakete betrachtet oder alle, ergibt sich aus den Istwerten der **relative** oder der **absolute Fertigstellungsgrad**.

- relativ: Einbeziehung aller Arbeitspakete,
- absolut: es werden nur die ganz fertigen Arbeitspakete berücksichtigt.

Der relative Fertigstellungsgrad enthält Schätzwerte (wie viel Prozent eines Arbeitspaketes sind schon fertig), das muss bei der Diskussion dieser Kennzahl berücksichtigt werden. Eine gute Datenerhebung bei den Tätigkeitsberichten erlaubt die Berechnung unterschiedlicher Fertigstellungsgrade:

- $FG_{plan} = A_{plan} / A_{ges,plan}$
- $FG = A_{fertig} / A_{ges,plan}$
- $FG_{korr} = A_{fertig} / A_{ges,korr}$
- A_{plan} geplanter Aufwand in MannMonaten
- $A_{ges,korr}$... geplanter Gesamtaufwand, korrigiert durch die Schätzungen der Mitarbeiter aus den Tätigkeitsberichten

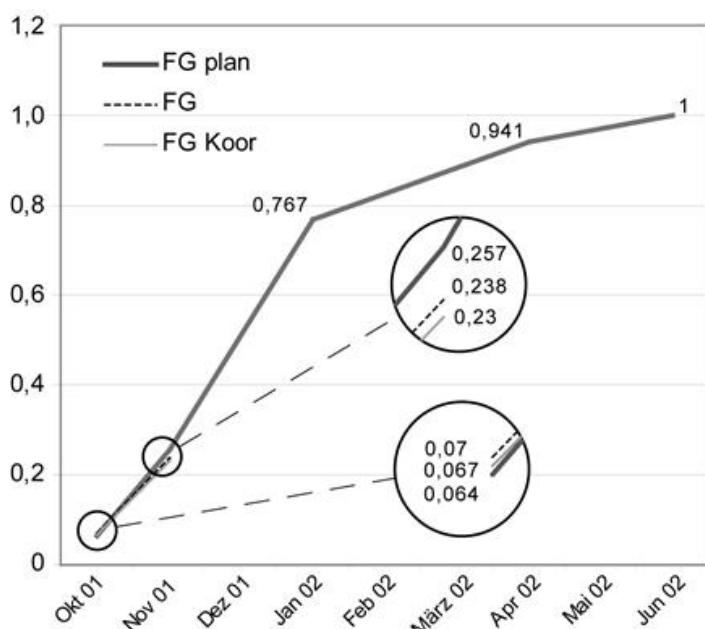
Besonders der FG_{korr} ist sehr aussagefähig. Er bildet einen guten Anhaltspunkt, um die Notwendigkeit von Steuerungsmaßnahmen zu erkennen. Das wird durch das folgende Beispiel verdeutlicht. Berechnet werden die oben genannten detaillierten Kennwerte jeweils als relative Kennwerte, d.h. es werden nicht nur die fertigen, sondern auch die angearbeiteten Arbeitspakete betrachtet.

Bezeichnung	A_{plan}	A_{ist}	$A_{gesch.}$	$A_{plan,kum.}$	$A_{ist,kum.}$	$A_{gesch.,kum.}$	FG_{plan}	FG	FG_{korr}
Start									
Machbarkeitsstudie	92	100	100	92	100	100	0,064	0,070	0,067
Release 1	276	240	320	368	340	420	0,257	0,238	0,230
Release 2	728	728	728	1096		1148	0,767		
Release 3	248	248	248	1344		1396	0,941		

Abschluss	84	84	84	1428		1480	1		
-----------	----	-----------	-----------	------	--	------	---	--	--

■ Tabelle 4.2.4.1: Beispielrechnung zur Bestimmung der unterschiedlichen Fertigstellungsgrade (kum.: kumulativ); die hervorgehobenen Werte wurden aus den Planwerten übernommen, da zum Berichtszeitpunkt noch keine Ist-Werte und Schätzwerte vorlagen

In der obigen Tabelle ist das Arbeitspaket Machbarkeitsstudie zum Berichtszeitpunkt im November (vgl. Abbildung 4.2.4.2) komplett abgearbeitet. Statt der geplanten 92 MM wurden 100 benötigt – 8 mehr als geplant. Das Arbeitspaket Release 1 sollte ebenfalls fertig sein, von den geforderten 276 MM sind aber nur 240 fertig und es wird sogar geschätzt, dass 320 MM, also 44 MM mehr als geplant, benötigt werden. In $A_{ges,korr}$, was sich im FG_{korr} ausdrückt, werden die insgesamt 52 MM, die mehr benötigt werden, berücksichtigt. Der Fertigstellungsgrad wird kleiner.



■ Abbildung 4.2.4.2: Grafische Auswertung der Fertigstellungsgrade (Werte aus Tabelle 4.2.4.1)

Wie das kleine Beispiel zeigt, müssen viele Randbedingungen betrachtet werden, damit dieser Kennwert tatsächlich eine Aussage hinsichtlich der Fertigstellung liefert. Vor allen Dingen muss eine Schätzung des Gesamtaufwandes vorliegen. Bei den iterativen Prozessmodellen liegen diese Schätzungen aber immer nur für einen relativ kleinen Zeitraum vor. In vielen Unternehmen wird deshalb auf die Berechnung dieses Kennwerts verzichtet. Stattdessen wird ein so genannter **prozessbezogener Fertigstellungsgrad** bestimmt. Es handelt sich hier um eine Aussage, nicht um einen numerischen Wert. In einem Bericht – auch Projektfortschrittsbericht genannt – werden alle erledigten und abgenommenen Arbeitspakete aufgelistet.

Projektfortschritt	
Berichtszeitraum	Datum
01.04.2002 - 05.04.2002	09.04.2002

Projekt			
geschätzter Endtermin ist der Durchschnittswert aus den Reviews im Berichtszeitraum			
Projekt-Nr.	Bezeichnung	Endtermin	Geschätzter Endtermin
01802	Animation 1	31.05.2002	31.05.2002

Nächster Meilenstein			
geschätzter Endtermin ist der Durchschnittswert aus den Reviews im Berichtszeitraum			
Meilenstein-Nr.	Bezeichnung	Endtermin	Geschätzter Endtermin
002	3D-Szenen R1	30.04.2002	03.05.2002

Einzelaufträge							
Ist-Werte aus den Tätigkeitsberichten im Berichtszeitraum							
EA-Nr.	geplante Stunden	erfasste Stunden	verbleibende Stunden	zusätzlich benötigte Stunden	Anfangstermin	Endtermin	Status
005	120	40	80	10	01.04.2002	19.04.2002	in Arbeit



Der in Abbildung 4.2.4.3 vorgeschlagene Projektfortschrittsbericht enthält wegen der Übersichtlichkeit auch Werte zum Projekt selbst sowie zu dem als nächstes anliegenden Meilenstein. Die erforderlichen Werte werden aus den Reviews und den Tätigkeitsberichten gewonnen.

Der Sachfortschritt oder die Reife eines Produkts kann auch durch einen **Stabilitätsindex** ausgedrückt werden. Dabei betrachtet man die vollzogenen Änderungen innerhalb eines Dokuments oder an dem Quellcode für ein Release 3:[*Tha00*].

$$S = (F_{\text{all}} - (F_{\text{neu}} + F_G + F_0)) / F_{\text{all}} \quad (2-6)$$

S	Stabilitätsindex		
F _{all}	Gesamtzahl der Funktionen		im betrachteten Release bzw. in der betrachteten Version eines Dokuments
F _{neu}	neue Funktionen		
F _G	geänderte Funktionen		
F ₀	gelöschte Funktionen		

Die Aussage lässt sich an dem folgenden Rechenbeispiel verdeutlichen:

Version	F _{all}	F _{neu}	F _G	F ₀	S
1.1	10	-	1	-	0,9
2.2	10	1	1	-	0,8
3.1	11	5	1	-	0,45

|  Tabelle 4.2.4.4: Bestimmung des Stabilitätsindex

Die Version 3.1 hat einen sehr niedrigen Stabilitätsindex, d.h. die Einführung der Version 3.1 kann Probleme bereiten, da sehr viele neue (und damit leider oft fehlerhafte) Funktionen eingefügt wurden.

Man kann diesen Stabilitätsindex auch periodisch, z.B. wöchentlich, ermitteln, um damit die Entwicklung der Stabilität über der Zeit darzustellen. Wünschenswert ist eine hohe Stabilität vor der Einführung eines Release – Änderungen in letzter Sekunde sind sehr fehleranfällig.

4.2.5 Qualität

Der Qualitätsbegriff deckt zwei Aspekte ab: Erstens die Erfüllung aller geforderten Funktionalitäten und zweitens deren Korrektheit und Fehlerfreiheit. Das spiegelt sich auch in den Definitionen wider:



Definition

Qualität nach EN ISO 8402:

Gesamtheit von Merkmalen (und Merkmalswerten) einer Einheit (d.h. ein Produkt, ein Prozess oder eine Dienstleistung) bezüglich Ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen.



Definition

Qualität nach B. Pothery:

Qualität beschreibt die Güte, Beschaffenheit und die Wertstufe eines Produkts. Qualität ist, wenn die Kunden zurückkommen, nicht die Produkte. [Rot00]



Definition

Qualität nach Thaller:

Ist die Fehlerfreiheit eines Produkts. [Tha00]

Bei der Entwicklung von Software treten statistisch gesehen immer Fehler auf, das hängt mit den besonderen Eigenschaften – besonders der Komplexität – zusammen. Dazu ein Beispiel: Die Explosion einer Ariane 5 Rakete im Jahre 1996 bei ihrem Jung-

fernflug. Die Entwicklungskosten betrugen 8 Milliarden US Dollar. Die versprochene Zuverlässigkeit betrug 98,5 %. Durch einen Softwarefehler fiel das Hauptsystem aus, das Kursdaten lieferte. Der Fehler trat in einem Unterprogramm auf, das eigentlich nur am Boden gebraucht wurde und einen Überlauf produzierte, der zum Anhalten des Prozessors führte.

Nach Thaller [[Tha00](#)] ergeben sich die folgenden Annahmen: In einem Programm treten im Durchschnitt 50 Fehler pro KSLOC auf, von diesen verbleiben 2 Restfehler pro KSLOC nach Auslieferung und Testung der Software. Davon wiederum sind 10% schwerwiegende Restfehler. Das verdeutlicht die folgende Tabelle:

Applikation	Umfang in SLOC	Fehler	Restfehler	Schwerwiegende Restfehler
Autopilot zur Steuerung einer Rakete	30.000	1.500	60	6
Navigationssysteme Space Shuttle	500.000	25.000	1.000	100
Software der Flugkontrolle über Europa	1.000.000	50.000	2.000	200
Software für Kernkraftwerke	1.500.000	75.000	3.000	300

■■■ Tabelle 4.2.5.1: Auftreten von Softwarefehlern

Quelle: [[Tha00](#)]

Ziel des Projektleiters muss also die Minimierung von Fehlern sein sowie die Erhöhung solcher Qualitätsparameter wie Benutzerfreundlichkeit und Robustheit, die aus Kundensicht besonders wichtig sind. Unter Betrachtung der Interessen der Entwickler spielt auch die Wiederverwendbarkeit und Wartbarkeit von Software eine große Rolle (vgl. Abschnitt 2.1 Projektziele).

Das Erreichen dieser Anforderungen setzt den Einsatz verschiedener Qualitätssicherungsmaßnahmen voraus, um die auftretenden Fehler zu minimieren. Die folgende Abbildung gibt einen Überblick über mögliche Maßnahmen.



Abbildung 4.2.5.2: Übersicht über Qualitätssicherungsmaßnahmen

Organisatorische Maßnahmen beziehen sich in erster Linie auf die Prozessgestaltung. Dazu gab es im Abschnitt 1.3 Projektphasen und Prozessmodelle sowie im Abschnitt 2.2 Risiken in Softwareprojekten viele Hinweise, die sich mit der klaren Empfehlung zusammenfassen lassen, iterative Prozessmodelle zu verwenden. Die Fortführung des Prozessgedankens führt zur Anwendung von standardisierten, zertifizierbaren Maßnahmen nach der ISO 9000 und dem Capability Maturity Model (CMM). Beim CMM handelt es sich um ein fünfstufiges Modell – jede Stufe charakterisiert den Reifegrad einer Organisation hinsichtlich seiner Vorgehensweise bei der Erstellung von Software. Die organisatorischen Maßnahmen betreffen nicht nur das Projekt, sondern es sind Qualitätssicherungsmaßnahmen, die das gesamte Unternehmen prägen.

Konstruktive Maßnahmen sind ebenfalls im Abschnitt 2.2 Risiken in Softwareprojekten besprochen worden. Hier empfiehlt es sich, für den Entwurf und die Programmierung objektorientierte Technologien zu verwenden. Die Wiederverwendbarkeit von Software kann besonders durch die Verwendung von Design Patterns nach Gamma [*Gam95*] erhöht werden. Die Wartbarkeit lässt sich durch den Einsatz von Refactory Methoden nach Fowler [*Fow00*] verbessern.

Beide Arten von Maßnahmen dienen dazu, Qualität zu bauen und nicht zu erprüfen. Softwaretests sind Arbeitsaufgaben der Mitarbeiter, die im Entwicklungsprozess berücksichtigt werden müssen. Diese werden hier nicht weiter besprochen.

In Abschnitt 4.1.1.2 wurde das Review als Erhebungstechnik für die Qualitätskontrolle des Projektmanagements empfohlen und beschrieben. Es wurde versucht, die Ergebnisse der Reviews über eine Nutzwertanalyse einer mathematischen Auswertung zu zuführen. Deshalb wurde in Auswertung der Beantwortung der Fragen zu einer Erhebungsgröße, z.B. Mitarbeitermotivation, eine Punktwertung abgegeben. Wertet man die Ergebnisse grafisch aus, so können Tendenzen schnell sichtbar gemacht werden –

z.B. eine sich verschlechternde Mitarbeitermotivation im letzten halben Jahr. In den Reviews werden folgende Kontrollgrößen erfasst:

- Vollständigkeit der Programmdokumentation
- Verständlichkeit und Übertragbarkeit des Quellcodes
- Vollständigkeit des Quellcodes
- Vollständigkeit des Programmtests
- Motivation der Mitarbeiter
- Kommunikation zwischen Projektmitgliedern
- Kommunikation mit dem Auftraggeber
- Qualität der Terminplanung
- Qualität der Personalplanung

All diese Kontrollgrößen charakterisieren die Qualität des Produkts und des Projektverlaufs und bilden damit eine wesentliche Voraussetzung für die Steuerung, z.B. Maßnahmen zur Erhöhung der Mitarbeitermotivation. In Abschnitt 4.1.1.2 wurden Vorschläge für die erforderlichen Checklisten unterbreitet.

Die Ergebnisse zur Qualitätskontrolle können in einem Bericht zusammengefasst werden. Ein Vorschlag dafür ist in Abbildung 4.2.5.3 dargestellt.

Projektqualität				
Berichtszeitraum	Datum			
01.04.2002 - 05.04.2002	09.04.2002			
Projekt				
geschätzter Endtermin ist der Durchschnittswert aus den Reviews im Berichtszeitraum				
Projekt-Nr.	Bezeichnung	Endtermin	Geschätzter Endtermin	
01802	Animation 1	31.05.2002	31.05.2002	
Planungsqualität				
Durchschnittswert aus den Reviews im Berichtszeitraum				
Kommunikation zwischen Projektmitgliedern	Kommunikation mit Auftraggeber	Mitarbeiter-motivation	Qualität Personal-planung	Qualität Personal-planung
2	3	4	2	2
Ergebnisqualität				
Werte aus den Reviews im Berichtszeitraum				
Prüfobjekt	Vollständigkeit	Korrektheit		
EA 005	1	2		

■ Abbildung 4.2.5.3: Aufbau eines Projektqualitätsberichts
Quelle: [Buh02]

4.2.6 Aufgaben zu Kontrollgrößen und Metriken



Aufgabe

Aufgabe 4.2.6-1

Gegeben ist der folgende Projektplan:

Vorgangsnummer	Vorgänger	Nachfolger	Dauer	Aufwand in MM
1	-	2	10	10
2	1	3,4	5	15
3	2	5	6	18
4	2	5	8	25
5	3,4	-	12	25

Zum Zeitpunkt x (Stichtag) sind 3 Arbeitspakete fertig und das 4. Arbeitspaket ist zu 70 % fertig.

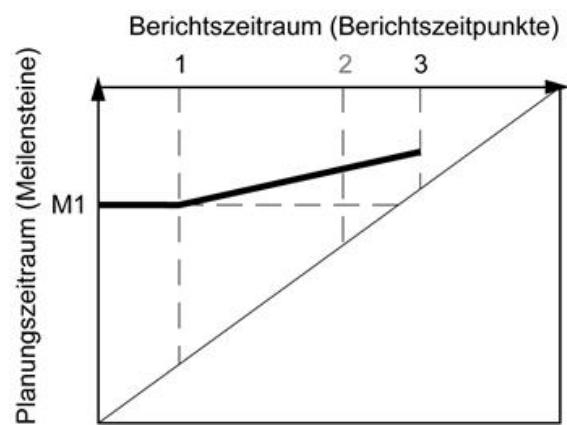
- a) Wie groß ist der absolute Fertigstellungsgrad?
- b) Wie groß ist der relative Fertigstellungsgrad?



Aufgabe

Aufgabe 4.2.6-2

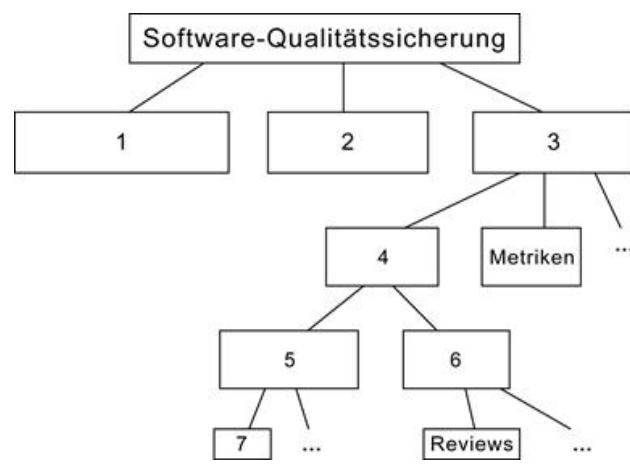
Die folgende Abbildung zeigt eine Meilensteintrendanalyse. Welche Aussage lässt sich für den dargestellten Meilenstein M1 (fette Linie) zu den Berichtszeitpunkten 1 und 3 treffen?



Aufgabe 4.2.6-3

Aufgabe

Ergänzen Sie die folgende Abbildung!





5 Projektabschluss

5 Projektabschluss

5.1 Produktübergabe

5.2 Projektanalyse



Lernziele

Was sollten Sie nach diesem Kapitel wissen?

- Wie wird die Produktabnahme gestaltet?
- Wie werden die Abweichungen und die Wirtschaftlichkeit analysiert?
- Wie werden Projekterfahrungen gesichert?



5.1 Produktübergabe

5.1 Produktübergabe

5.1.1 Produktabnahme

5.1.2 Produktbetreuung

5.1.3 Aufgaben zur Produktübergabe

Ein geregelter Abschluss des Projekts ist genauso wichtig wie ein geregelter Projektverlauf. Das Projekt muss eindeutig enden – das Produkt ist fertig und wird an den Auftraggeber übergeben. Ein positives und eindeutiges Ende hat einen positiven Einfluss auf die Motivation der Mitarbeiter für zukünftige Projekte. Zur Übergabe finden Abnahmetests statt. Ein Produkt muss weiter betreut werden. Dazu sind Regelungen zu schaffen. Für den Projektleiter sind eine Projektanalyse und eine daraus folgende Erfahrungssicherung besonders wichtig. Dadurch kann er für spätere Projekte bessere Planungen abliefern und Ausgangswerte für den Einsatz formalisierter Schätzverfahren gewinnen. Das Team, falls es erfolgreich gearbeitet hat, sollte als kostbares Gut behandelt werden und eventuell für ein nächstes Projekt erhalten bleiben. Es ist ein großer Vorteil, wenn die Fähigkeiten der Mitarbeiter und ihre Einsatzbereitschaft bekannt sind; denn Softwareprojekte hängen zum größten Teil von den Mitarbeitern ab.

5.1.1 Produktabnahme

Kernpunkt der Produktabnahme – wobei hier unter **Produkt** das im Projektauftrag spezifizierte Ergebnis verstanden wird – sind die Abnahmetests, die entsprechend zu protokollieren sind und in ein Übernahmeprotokoll münden.

Abnahmetests

Nach Burghardt [Bur95] lassen sich die Abnahmetests folgendermaßen unterteilen:

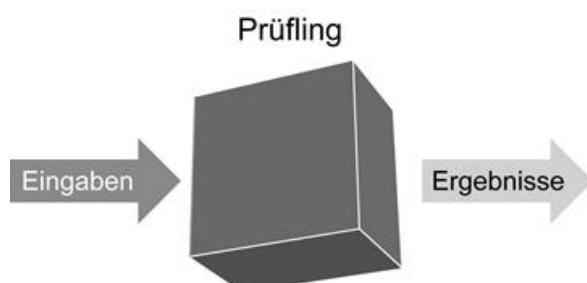
- **Produkttests** – Abnahmetests für Softwareprodukte
- **Abschlusstests** – Abnahmetests für Hardwareproduktentwicklungen, die anschließend gefertigt werden.
- **Akzeptanztests** – Abnahmetests für fertig entwickelte und gefertigte Systeme, die sowohl Hardwarekomponenten als auch Softwarekomponenten enthalten (Fertigungsstraßen in der Automobilindustrie).
- **Pilottests** – Abnahmetests für Verfahrensentwicklungen, die die volle Einsatzreife des Produkts nachweisen sollen. Es werden keine speziellen Testdaten generiert, sondern das installierte Produkt wird im Probebetrieb übernommen.

Die folgenden Ausführungen zum Abnahmetest beziehen sich nach Burghardt [Bur95] auf den Produkttest. Nach Frühauf u.a. [Frü00] unterscheiden sich die Abnahmetests von den "normalen" Tests durch folgende Eigenschaften:

*"Als Software-Entwickler haben wir in einem Projekt (hoffentlich) nur eine Abnahme zu bestehen; wir testen aber häufig Module, Teilsysteme und das ganze Software-System. Im Folgenden sprechen wir vom **Testen** als einer Prüfung, die Software-Lieferanten im Sinne der Definition von Myers durchführen (Def.: "Testen ist die Ausführung eines Programms mit dem Ziel, Fehler zu entdecken" [Mye95]), und reden von **Abnahmen** (oder Abnahmetest), wenn der Auftraggeber den Test beobachtet oder selbst durchführt."*

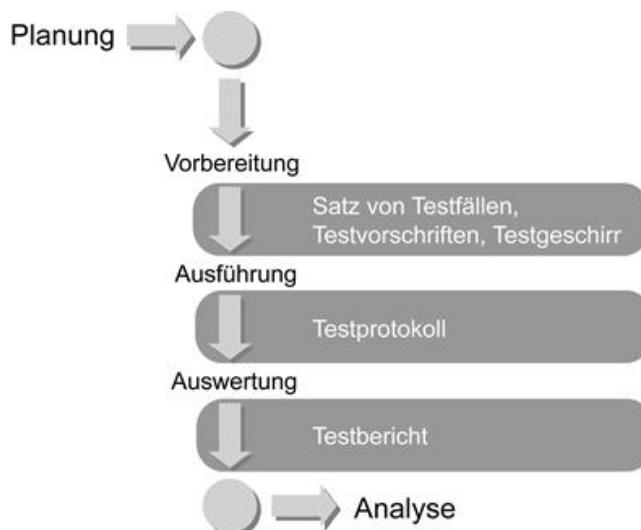
Gegenstand eines **Abnahmetests** sind ganze Programme (**Systemtest**). Dabei wird nicht nur der selbst erstellte Programmtext getestet, sondern das Objektprogramm. Dieses Programm enthält Bibliotheks Routinen, ist eingebettet in ein Betriebssystem und wirkt mit anderen Komponenten (z.B. Datenbankservern) zusammen.

Abnahmetests sind sogenannte **Black-Box-Tests**: Es wird von den funktionalen Anforderungen an das Programm ausgegangen – der innere Aufbau des Programms wird dazu nicht betrachtet. Letztendlich werden Eingaben realisiert und die Ergebnisse geprüft.



| Abbildung 5.1.1.1: Schematische Darstellung eines Black-Box-Tests

Der Ablauf des Abnahmetests kann in Anlehnung an Frühauf u.a. [Frü00] nach Abbildung 5.1.1.2 erfolgen.



■ Abbildung 5.1.1.2: Ablauf des Abnahmetests nach Frühauf
Quelle: [Frü00]

Die **Testvorbereitung** erfordert die meiste Zeit. Besonders gründlich müssen die Testfälle entwickelt werden. In der Literatur [Tha00], [Frü00], [2:Tha00] sind viele Verfahren beschrieben. Dazu zählen:

- Funktionsüberdeckung,
- Eingabeüberdeckung,
- Ausgabeüberdeckung,
- Äquivalenzklassen,
- Grenzwertanalyse.

Alle diese Verfahren erzeugen eine große Anzahl von **Testfällen**, um das Produkt ausreichend zu testen. Dabei ist davon auszugehen, dass ein Testfall dann gut ist, wenn er mit hoher Wahrscheinlichkeit einen Fehler aufzeigt. Mit der Entwicklung der Testfälle muss ein erfahrener Mitarbeiter beauftragt werden, der im Falle eines Abnahmetests eng mit dem Kunden zusammenarbeitet, um alle relevanten Funktionen zu berücksichtigen.

Das Testvorgehen soll so organisiert werden, dass möglichst wenig Zeit beansprucht wird. Berücksichtigen Sie dabei insbesondere die **Testumgebung**, deren Vorbereitung erhebliche Zeit beanspruchen kann. Unter dem Aspekt der Zeit sind auch die **Testgeschirre** zu betrachten. Unter Testgeschirren sind Programme (Testtools) zu verstehen, die den Test unterstützen, z.B. das wiederholte Ausfüllen (Treiben) von Oberflächen, usw. Die Testvorbereitung ist in einem **Testplan** zu dokumentieren. Dieser kann den im Beispiel dargestellten Aufbau haben.

Beispiel zum Testplan

1. Teststrategie
 - 1.1. Beschreibung der Komponente und Einbindung ins Gesamtsystem
 - 1.2. Teststart
 - 1.3. Testziele
2. Testfälle
 - 2.1. Testfallentwurf
 - 2.2. Formate der Testdaten
 - 2.3. Testfälle tabellarisch
3. Testumgebung
 - 3.1. Softwareumgebung (Server, Testklassen u. andere Komponenten)
 - 3.2. Anzulegende Verzeichnisse oder Log-Files
 - 3.3. Besondere Hardwarevoraussetzungen
 - 3.4. Hinweise für die automatische Abarbeitung
4. Testabarbeitung
5. Sonstige Hinweise

Während der Testausführung sollten beim Auftreten eines Fehlers keine Korrekturen am Programm vorgenommen werden, bevor nicht alle Testfälle tatsächlich abgearbeitet wurden.

Wichtig ist eine **Protokollierung von Fehlern**. Werden Testtools verwendet, übernehmen diese die Arbeit. Müssen Sie per Hand testen, sollten Sie sich eine entsprechende Vorlage schaffen.

Beispiel zum Fehlerprotokoll

Einzeltest Nr.	Fehlerbeschreibung	Kommentar zur Ursache	Priorität

Bei der Testauswertung sollten die Fehler und die daraus resultierenden Auswirkungen auf den Projektabschluss besprochen werden. Schwerwiegende Abweichungen können zur Wiederholung der Tests und damit zu einer Verschiebung der Übernahme führen.

Die schriftliche Fixierung erfolgt in einem Bericht bzw. Übernahmeprotokoll, das von den Vertragspartnern unterzeichnet wird. Darin können auch kleinere Abweichungen dokumentiert werden, die eine Übernahme aber prinzipiell nicht verhindern. Damit hat der Kunde die Sicherheit, dass auch nach Übernahme des Programms die Fehler behoben werden.

Übernahmeprotokoll

Neben den Ergebnissen der Tests enthält das Übernahmeprotokoll noch weitere Punkte, die Sie im Beispiel nachlesen können.

Beispiel zum Übernahmeprotokoll

1. Übernahmeobjekte
2. Durchgeführte Prüfungen
 - 2.1. an Programmen
 - 2.2. an Dokumenten
3. Festgestellte Mängel
 - 3.1. an Programmen
 - 3.2. an Dokumenten
4. Nachforderungen an den Auftragnehmer
 - 4.1. offene Mängel und Fehlerbehebung
 - 4.2. technische Änderungen
 - 4.3. Dokumentationserweiterungen
 - 4.4. Preiskorrekturen
5. Abnahmeentscheidung
 - 5.1. Bemerkungen
 - 5.2. Fristen

5.1.2 Produktbetreuung

In der Softwarebranche wird die Produktbetreuung in **Wartungsverträgen** geregelt. Nach Burghardt [Bur95] sind die folgenden Punkte in einem Wartungsvertrag zu berücksichtigen:

- Fehlerbehebung nach Ablieferung
- Anpassung an neue Betriebssystemversionen
- Einbindung neuer Versionen von implementierter Standard- und Basissoftware
- Optimierung der Benutzeroberflächen
- Verbesserung der Ablaufeigenschaften
- Anpassung an veränderte Ablauforganisation

Der Wartungsvertrag kann auch die Personen und Ansprechpartner benennen, die für die genannten Aufgaben bereitstehen. Das ist wichtig, damit Mitarbeiter mit Systemwissen berücksichtigt werden können. Insbesondere kann hier auch auf die Wünsche des Auftraggebers eingegangen werden (evtl. Vertrauenspersonen).

5.1.3 Aufgaben zur Produktübergabe



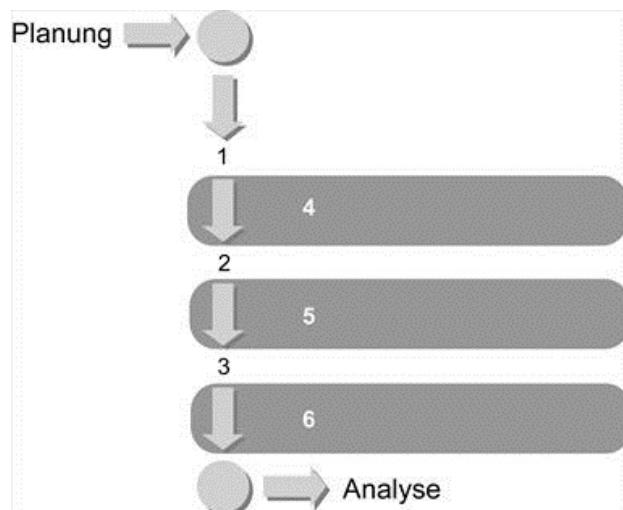
Aufgabe 5.1.3-1

An dieser Stelle befindet sich online ein Quiz.

https://vfhpmp.eduloop.de/loop/Aufgaben_zur_Prod%C3%BCbergabe

Aufgabe 5.1.3-2

Ordnen Sie die richtigen Begriffe aus der Tabelle den Nummern in der Abbildung zum Ablauf eines Abnahmetests zu:



A) Risikoanalyse	G) Satz von Testfällen / Testvorschriften / Testgeschrif
B) Ausführung	H) Testbericht
C) Vorbereitung	I) Interview
D) Aufwandsschätzung	J) Testprotokoll
E) Auswertung	K) Review
F) Realisierung	



Aufgabe

Aufgabe 5.1.3-3

Gehören die folgenden Punkte in ein Übernahmeprotokoll?

An dieser Stelle befindet sich online ein Quiz.

https://vfhpmp.eduloop.de/loop/Aufgaben_zur_Prod%C3%BCbergabe



Gliederung

5.2 Projektanalyse

5.2 Projektanalyse

5.2.1 Abweichungs- und Wirtschaftlichkeitsanalyse

5.2.2 Erfahrungssicherung

5.2.3 Aufgaben zur Projektanalyse

Die Analyse des abgeschlossenen Projekts erfordert einen umfassenden Soll-Ist-Vergleich. Abweichungen müssen analysiert werden und hinsichtlich der Ursachen und der Wirksamkeit der eingeleiteten Maßnahmen untersucht werden. Die Erfahrungen müssen im Unternehmen möglichst einheitlich gesichert und systematisch aufbereitet werden, damit z.B. positive Steuerungsmaßnahmen schnell verbreitet und gemachte Fehler in anderen Projekten vermieden werden.

5.2.1 Abweichungs- und Wirtschaftlichkeitsanalyse

Bei einer Abweichungsanalyse werden die ursprünglich geplanten Vorgaben für das Projekt (Termine, Kosten und Aufwände) sowie für das zu erbringende Produkt (Leistungs- und Qualitätsmerkmale) den aktualisierten Planvorgaben und den endgültig erreichten Ergebnissen gegenübergestellt (Soll-Ist-Vergleich). Die Abweichungen müssen hinsichtlich ihrer Ursachen analysiert werden. Sehr wichtig ist dann auch die Einschätzung der getroffenen Maßnahmen, die den Abweichungen entgegenwirken sollen, hinsichtlich ihrer Wirksamkeit.

Für die Abweichungsanalyse können die implementierten Anwendungsfälle des Programms oder aber auch einzelne Arbeitspakete untersucht werden (aufgabenbezogene Betrachtung). Bedenken Sie, dass eine detaillierte Betrachtung bei der Verwendung von iterativen Prozessmodellen für die Planung von zukünftigen Projekten den größten Nutzen bringt. Auch wenn das nächste Projekt ein ganz anderes Einsatzgebiet als Gegenstand hat, werden doch vielfach vergleichbare Anwendungsfälle zu implementieren sein (z.B. eine Datenbankanbindung).

Wird der Rational Unified Process (RUP) verwendet, können einzelne Use Cases (Anwendungsfälle) als ein betrachteter Teilbereich analysiert werden. Im eXtreme Programming (XP) können die User Storys diese Rolle übernehmen. Aufschlussreich kann auch eine phasenbezogene Analyse sein. Man betrachtet dazu alle Aufgaben, die z.B. in die Konzeptionsphase (oder auch Inception Phase) mit dem Schwerpunkt der Anforderungsspezifikation fallen (vgl. auch Kennzahlen nach Burghardt weiter unten in diesem Abschnitt).

Für jeden betrachteten Teilbereich sind die projektbezogenen Parameter (Termine, Kosten und Aufwände) sowie die aufgaben- oder produktbezogenen Parameter (Leistungs- und Qualitätsmerkmale) zu ermitteln. Hier offenbart sich schon eine Grundvoraussetzung für eine erfolgreiche Analyse – die richtige und vollständige Erfassung der notwendigen Daten im Projektverlauf. Dazu sollte Projektmanagementsoftware verwendet werden. Mindestanforderung ist die Führung eines Projekttagebuchs.

Als sehr nutzbringend erweist sich die Erfassung von Problemen (Ursachen, die oft zu Abweichungen führen) und die Speicherung der Lösungsvorschläge und realisierten Lösungen. So können Erfahrungen schon während des Projektverlaufs weitergegeben werden. Die Abbildung 5.2.1.1 zeigt einen entsprechenden Formularvorschlag aus [Buh02] für den Tätigkeitsbericht. Die Erfassung sollte auch in den Reviews erfolgen.

Problem-Überblick	
Bereich	<input type="text"/>
Bezeichnung	<input type="text"/>
Beschreibung	<input type="text"/>
Lösungsvorschlag	
Lösung	

■ Abbildung 5.2.1.1: Erfassung von Problemen und Lösungen

Quelle: [Buh02]

Im Folgenden wird auf einige Problembereiche eingegangen, die in vielen Softwareprojekten auftreten und dann zu den ungewünschten Abweichungen führen (vgl. auch Abschnitt 2.2 Risiken in Softwareprojekten).

Sehr häufig haben Probleme technische Ursachen. Dazu zählen:

- technologische Grenzen,
- nicht gelieferte Hardware,
- unsichere Systembasis.

Besonders die technologischen Grenzen zwingen zur Aufgabe von Einzelzielen oder ganzen Projekten. Aktuelle Beispiele sind im Bereich Java und XML zahlreich zu finden. Die Konkurrenz in der Softwarebranche ist riesengroß. Hersteller versprechen mehr als das Produkt tatsächlich hergibt und verschweigen Kinderkrankheiten. Bauen Sie auf solche Produkte Ihre Entwicklung auf, kann es ein böses Erwachen geben. Es ist verlockend, die neuesten Produkte mit den gerade benötigten Features in die eigene Entwicklung einzubauen – prüfen Sie vorher genau. Fordern Sie Testlizenzen und implementieren Sie anhand von Beispielen die Kernfunktionalitäten, die von dem eingesetzten Produkt abgedeckt werden sollen.

Andere technische und organisatorische Fehler, wie

- Planungsfehler,
- unvollständige Testdaten und
- mangelnde Toolnutzung

sind vermeidbar. Hier müssen Fehleinschätzungen und die eingeleiteten Maßnahmen ausgewertet und verfügbar gemacht werden.

Bei Softwareprojekten liegen oft **personelle Ursachen** für Abweichungen vor. Wobei

- Krankheit,
- Schwangerschaft oder
- eine Kündigung durch den Mitarbeiter

kaum zu vermeiden sind. Falls mangelnde Fähigkeiten der Mitarbeiter zu Abweichungen geführt haben, sollten Sie für ihre nächsten Projekte bei unbekannten Mitarbeitern Einstellungstests vornehmen. Auf wirkliche **Managementfehler** weisen solche Ursachen hin wie:

- Demotivation,
- Missverständnisse und
- Überlastung.

Abweichungen können auch **organisatorische Ursachen** haben. An erster Stelle sind da sicherlich personelle Engpässe zu nennen. Ärgerlich sind auch Wartezeiten für Tests (Testumgebung ist nicht abgesichert oder die Testabläufe sind nicht automatisiert). Eine Frage der Organisation ist auch die rechtzeitige und vollständige Ausstattung der Arbeitsplätze sowie die Bereitstellung ausreichender Räumlichkeiten. Falls hier Probleme auftraten, versuchen Sie im nächsten Projekt, diese Ursachen abzustellen.

en. Es gibt auch organisatorische Ursachen, die nicht zu beeinflussen sind – z.B. ein Auftragnehmer für eine Teilleistung geht in den Konkurs.

Im Abschnitt Projektziele wurden Betrachtungen zur Wirtschaftlichkeit von Softwareprojekten unternommen. Dazu wurden Verfahren der statischen Investitionsrechnung genutzt. Für nicht quantifizierbare Parameter wurde eine Nutzwertanalyse vorgeschlagen. Natürlich ist es jetzt besonders interessant, ob die angestrebte Wirtschaftlichkeit bzw. der angestrebte Nutzen mit den Ist-Zahlen für Kosten und Aufwand auch noch gegeben ist. Abweichungen von der Wirtschaftlichkeitsrechnung während der Zielbestimmung können eine Reihe von Ursachen haben:

- viele Änderungswünsche des Kunden,
- Preissteigerungen,
- Lohnsteigerungen,
- geringerer Nutzeffekt des Produkts,
- verspäteter Einsatzbeginn,
- höherer Aufwand.

Sie können die Rechenverfahren einsetzen, die im Abschnitt 2.1 Projektziele vorgeschlagen wurden.

5.2.2 Erfahrungssicherung

Unter Erfahrungen versteht man die Gesamtheit der Erlebnisse und der daran gewonnenen Erkenntnisse des Menschen in der Auseinandersetzung mit der Welt oder sich selbst. Erfahrungen müssen dokumentiert werden. Die Dokumentation sollte so erfolgen, dass die Weitergabe von Erfahrungswissen unabhängig von bestimmten Personen (Erfahrungsträgern) möglich ist. Der Nutzen von Erfahrungen in Softwareprojekten ergibt sich aus der Anwendung der gesammelten Daten, z.B:

- bei der Aufwandsschätzung zukünftiger Projekte,
- bei der Behebung von Problemen (Erfahrungen) in laufenden Projekten,
- bei der Beurteilung und der Auswahl von Steuerungsmaßnahmen in laufenden Projekten,
- bei der Anpassung von Prozessparametern in laufenden Projekten,
- bei der Beurteilung abgeschlossener Projekte.

Im Allgemeinen unterscheidet man zwischen **Messdaten** und **Merkmalsdaten**, die als Erfahrungsdaten zu einem Projekt gespeichert werden müssen.

Merkmalsdaten dienen vornehmlich dazu, die Vergleichbarkeit von Projekten herzustellen. Dazu müssen relevante Merkmale herausgearbeitet werden, die ein Projekt von anderen unterscheidet oder vergleichbar macht. Burghardt schlägt z.B. Vergleichsmerkmale vor, die sich wie folgt unterteilen lassen:

- Projekteinordnung (Neuentwicklung, Weiterentwicklung, Redesign, usw.)
- Aufgabe (wissenschaftlich-technische Programme, Compiler, Editoren, Web-Anwendungen, usw.)

Merkmalsdaten ergeben sich auch aus den verwendeten Techniken und Methoden, z.B. für die Aufwandsschätzung.

Im dort betrachteten **Function-Point-Verfahren** gibt es eine Reihe von Einflussfaktoren, die für die Umrechnung von Unadjusted Function Points in Adjusted Function Points genutzt werden. Diese Faktoren können als Merkmalsdaten betrachtet werden. Im ebenfalls besprochenen **CoCoMo-II-Verfahren** sind die Effort Multipliers (EM) und der Scale Factor (SF) eine sehr gute Grundlage zur Festlegung der Merkmalsdaten. Kontrollieren Sie, ob Ihre Merkmalsdaten tatsächlich eine sinnvolle Anwendung des CoCoMo-II-Verfahrens erlauben.

Im nächsten Schritt werden **Messdaten** (Metriken) erhoben, die dann z.B. für die oben genannten formalisierten (Aufwands-)Schätzverfahren eingesetzt werden können. Bei der **Function Point Analysis** erfolgt die Ermittlung der Function Points – bei der Verwendung der IFPUG-Methode (International Function Point User Group) – durch die Einschätzung des Umfangs und des Schwierigkeitsgrads. Gemessen und in die Erfahrungsdatenbank gespeichert werden (vgl. Abschnitt 3.3 Planungstechniken) die Anzahl und der Schwierigkeitsgrad der externen Eingaben, Ausgaben, Abfragen sowie der internen und externen Geschäftsentitäten.

Wollen Sie das Function-Point-Verfahren erfolgreich einsetzen, müssen Sie auch die Ist-Function Points ermitteln und den dazugehörigen Ist-Aufwand – beide Größen gehören auch in Ihre Erfahrungsdatenbank. Mit Hilfe der Analogiemethode und unter Be trachtung der Vergleichsmerkmale können Sie dann zukünftige Projekte schätzen.

Das **CoCoMo-II-Verfahren** benötigt als Eingangsgröße die KSLOC (vgl. Abschnitt 3.3 Planungstechniken). Deshalb ist es sehr sinnvoll, für jedes abgeschlossene Projekt die Anzahl der Quellcodezeilen zu ermitteln und zusammen mit den Merkmalsdaten zu speichern.

Neben den schon erwähnten Messdaten sollten weitere Daten als Erfahrung gespeichert werden:

- Speicherbedarf eines Programms,
- Kosten des Projekts,
- Dauer des Projekts,
- Anzahl der eingesetzten Mitarbeiter.

Burghardt schlägt die **Berechnung von Kennzahlen** vor, um weitere Erkenntnisse zu gewinnen. Die Auswertung der prozessorientierten Kennzahlen ist sehr nützlich, da sie weitere Hinweise für die Schätzung zukünftiger Projekte geben.

$$\text{Aufwandsmäßiger Phasenriss} = \frac{\text{Aufwand einer Phase}}{\text{Gesamtaufwand}} \cdot 100\%$$

Aus der praktischen Erfahrung ist bekannt, dass sich einzelne Phasen eines Projekts genauer schätzen lassen als andere Phasen (vgl. Abschnitt 3.3 Planungstechniken, Abbildung 3.3.1.1). Besondere Unsicherheiten gibt es z.B. bei der Schätzung des Aufwands für die Spezifikation der Systemanforderungen. Durch diesen Kennwert können Sie spätere Schätzungen überprüfen.

$$\text{Aufwandsmäßiger Tätigkeitsaufriss} = \frac{\text{Aufwand einer Tätigkeit}}{\text{Gesamtaufwand}} \cdot 100\%$$

Um diesen Kennwert auszuweisen, müssen in den Tätigkeitsberichten die Aufwände der einzelnen Tätigkeiten erhoben werden. Man kann hier auch die Ausfallzeiten auswerten. Für spätere Projekte hat man damit einen guten Anhaltspunkt, welche Zeit für welche Tätigkeit einzuplanen ist.

Für die folgenden beiden Kennwerte lässt sich das oben Gesagte übernehmen, nur dass hier nicht der Aufwand betrachtet wird, sondern die Dauer.

$$\text{Zeitlicher Phasenriss} = \frac{\text{Dauer einer Phase}}{\text{Gesamtdauer}} \cdot 100\%$$

$$\text{Zeitlicher Tätigkeitsaufriss} = \frac{\text{Dauer einer Tätigkeit}}{\text{Gesamtdauer}} \cdot 100\%$$

Es ist ganz klar, dass solche detaillierten Kennzahlen nur durch eine Speicherung aller abgerechneten Tätigkeitsberichte und deren Auswertung gewonnen werden können.

Weitere Hinweise für Projektmetriken finden Sie in [Lor94].

5.2.3 Aufgaben zur Projektanalyse

**Aufgabe**

Aufgabe 5.2.3-1

Für die Abweichungsanalyse können Teilbereiche des Projekts betrachtet werden.

- Welche Teilbereiche bieten sich bei Verwendung des Rational Unified Process bzw. des eXtreme Programming an? Geben Sie jeweils eine kurze Begründung!

**Aufgabe**

Aufgabe 5.2.3-2

Nennen Sie Vergleichsmerkmale für Softwareprojekte!

**Aufgabe**

Aufgabe 5.2.3-3

Woraus lassen sich Merkmalsdaten bei Verwendung der folgenden formalisierten Schätz-verfahren ableiten:

- Function Point Verfahren
- CoCoMo-II-Verfahren

**Aufgabe**

Aufgabe 5.2.3-4

Welche Messgröße ist für das CoCoMo-II-Verfahren zu ermitteln?

Musterlösung

Das CoCoMo-II-Verfahren beruht darauf, dass die KSLOC bekannt sind.



Lernziele



6 Teamführung

6 Teamführung

- 6.1 Motivationstheorien
- 6.2 Führungshinweise

Was sollten Sie nach diesem Kapitel wissen?

- Welche Faktoren beeinflussen das Verhalten von Menschen?
- Welche Rückschlüsse lassen sich aus der Menschentypforschung ziehen?
- Welche Motivationstheorien gibt es?
- Wie sollten Sie Ihre Führungstätigkeit organisieren?



6.1 Motivationstheorien

6.1 Motivationstheorien

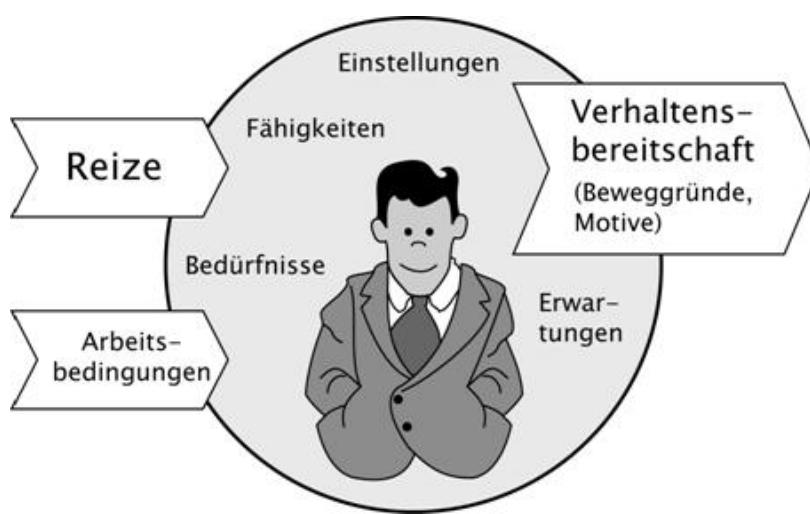
- 6.1.1 Wichtige Begriffe
- 6.1.2 Menschentypen
- 6.1.3 Einige Beispiele für Motivationstheorien
- 6.1.4 Aufgaben zu Motivationstheorien

Softwareprojekte hängen ganz entscheidend von den Mitarbeitern ab. Deren Einsatz und deren Fähigkeiten bestimmen, ob ein Softwareprojekt tatsächlich Erfolg hat oder nicht. Aus eigener Erfahrung weiß ich, dass es sehr schnell mit einem Projekt bergab gehen kann, wenn im Team die Motivation nicht stimmt und Krisenstimmung aufkommt. Als Projektleiter müssen Sie solche Situationen erkennen und durch geeignete Maßnahmen entschärfen. Dieser erste Abschnitt geht auf Grundlagen der Führungsslehre ein – besonders auf die Arbeit mit dem einzelnen Mitarbeiter. Das Verhalten von Teams und deren effektive Führung wird im Abschnitt 6.2 behandelt.

6.1.1 Wichtige Begriffe

Führung beinhaltet die Einflussnahme auf das Verhalten anderer Menschen. Wie ist auf das Verhalten einzuwirken, damit der Betreffende in einer gewünschten Weise tätig wird? Das Verhalten von Mitarbeitern ist nicht in dem Maße vorhersagbar wie das Verhalten technischer Systeme! Die einzelnen Mitarbeiter reagieren auf gleiche Maßnahmen ganz unterschiedlich und die Reaktionen hängen auch von der jeweiligen

Situation ab. Dabei spielen die in der Abbildung 6.1.1.1 dargestellten Faktoren eine entscheidende Rolle:



■ Abbildung 6.1.1.1: Faktoren, die das Verhalten von Menschen beeinflussen

Die Verhaltensbereitschaft wird in erster Linie von der Motivation eines Mitarbeiters bestimmt. Die Motivation lässt sich durch gute Führungstätigkeit positiv verändern. Weiteren Einfluss haben aber auch:

- Einstellungen
 - Hierbei handelt es sich um Grundauffassungen und Ansichten. So hat ein Mitarbeiter beispielsweise Motive, um im Projekt mehr zu leisten, aber seine Einstellung lässt es nicht zu: "Wenn ich Überstunden machen würde, käme das Projekt besser voran, aber so was tu ich grundsätzlich nicht – wozu hab ich einen Arbeitsvertrag?!"
- Erwartungen
 - Wie wahrscheinlich ist es, dass bestimmte Ergebnisse als Folge des Verhaltens eintreten?
 - Mit welcher Wahrscheinlichkeit haben die Ergebnisse auch die gewünschten Konsequenzen?
 - Erwartungen werden geprägt durch Erfahrungen (Er hat Motive, aber erwartet keine Besserung).
- Fähigkeiten
 - Potential einer Person, mit entsprechender Kompetenz in einer Lebenssituation zu handeln (Er hat Motive, aber er schafft es nicht).

Diese Faktoren lassen sich im Projektverlauf schwerer beeinflussen und bilden die Grundlage der Einordnung von Menschen in so genannte "Menschentypen" (vgl. nächsten Abschnitt).



Motivation ist die Summe aller Motive eines Menschen, etwas zu tun. Sie bildet sich durch Erfahrungen und Erlebnisse positiver und negativer Art. Durch äußere Anreize werden diese Antriebe mehr oder weniger stark empfunden.

Die eigentliche Führungstätigkeit eines Projektleiters besteht also in der Aufgabe, für jeden Mitarbeiter die richtigen Anreize zu schaffen.



Abbildung 6.1.1.2: Erhöhung der Motivation durch auf den Mitarbeiter abgestimmte Anreize

Aus Abbildung 6.1.1.2 ist ersichtlich, dass die beiden dargestellten Mitarbeiter unterschiedlich starke Anreize benötigen.



Anreize sind Gegebenheiten, die wahrgenommen werden und Motive aktivieren.

Man unterscheidet:

- Finanzielle Anreize
 - Bedürfnis nach Geld als Mittel zur Befriedigung von Grundbedürfnissen,

- Geltungsbedürfnis, wenn in der Gehaltshöhe ein Statussymbol gesehen wird,
- Bedürfnis nach Geld als Selbstzweck.
- Soziale Anreize: Sie ergeben sich aus dem Verhältnis Vorgesetzter und Mitarbeiter sowie den Mitarbeitern untereinander. Dazu zählen:
 - **Kontaktbedürfnisse** von Mitarbeitern - auch Machtbedürfnis: das Bedürfnis in einem gut funktionierendem Team zu arbeiten, das Bedürfnis auch mit dem Chef Kontakt zu haben, das Bedürfnis auch in der Freizeit gemeinsam etwas zu unternehmen.
 - **Aufgabeninhalt:** ergibt sich aus der Bedeutung, die einer Aufgabe zugemessen wird; Bedürfnis nach sozialer Geltung; Selbstverwirklichung.
 - **Entfaltungsmöglichkeiten:** Anreize, welche eine Aktivierung der Selbstentfaltungsmöglichkeiten bewirken; Aufstiegschancen, Weiterbildungsmöglichkeiten; Grad an Autonomie am Arbeitsplatz.

Da nicht alle Mitarbeiter in der gleichen Weise auf gegebene Anreize reagieren, kann es hilfreich sein, verschiedene grundlegende Menschentypen zu betrachten. Motivieren (Anreize schaffen) setzt die Kenntnis der Bedürfnisse und Lebensziele voraus. Mitarbeiter werden Ihnen aber selten ihre Beweggründe des Handelns offen mitteilen.
"Wahre" Motive werden selten zugegeben!



■ Abbildung 6.1.1.3: Faktoren, die das Verhalten von Menschen beeinflussen

Grundlegend lässt sich hier zusammenfassen, dass die Motivation gemeinsam mit den Fähigkeiten, den situativen Einflüssen und den Arbeitsbedingungen das Arbeitsergebnis bestimmt.

6.1.2 Menschentypen

Wie aus dem vorigen Abschnitt hervorgeht, hängt die Auswahl der Anreize zur Erhöhung der Motivation auch von den Menschentypen ab. Das hat die Forschung veranlasst, zu untersuchen, ob sich allgemeine Beschreibungen zu Menschentypen finden lassen. Grundlage der Einordnung von Mitarbeitern in bestimmte Typen sind die Bilder vom Menschen – in der Literatur werden 4 Typen beschrieben:

- Rationaler Mensch
- Sozialer Mensch
- Selbstverwirklichender Mensch
- Komplexer Mensch

Der **rationale Mensch** ist hauptsächlich durch monetäre Anreize motiviert. Er ist passiv und wird von der Organisation manipuliert und kontrolliert. Sein Handeln ist ökonomisch rational. Für das Management ergeben sich daraus die folgenden Aufgaben: Planen, Organisieren, Kontrollieren. Die Organisation und deren Effizienz stehen im Mittelpunkt. Organisation hat die Aufgabe, irrationales Verhalten zu neutralisieren und kontrollieren.



Der **soziale Mensch** ist in erster Linie durch soziale Bedürfnisse motiviert. Er wird stärker durch soziale Normen seiner Arbeitsgruppe als durch Anreize und Kontrolle des Vorgesetzten gelenkt. Für das Management ergeben sich daraus die folgenden Aufgaben: Aufbau und Förderung von Gruppen. Die Bedürfnisse nach Anerkennung, Zugehörigkeitsgefühl und Identität müssen befriedigt werden.



Der **selbstverwirklichende Mensch** strebt nach Autonomie und bevorzugt Selbst-Motivation und Selbst-Kontrolle. Es gibt keinen zwangsläufigen Widerspruch zwischen Selbstverwirklichung und organisatorischer Zielerreichung. Manager sind Unterstützer und Förderer. Die Delegation von Entscheidungen an diese Mitarbeiter ist möglich und wünschenswert. Es findet ein Übergang von Amts-Autorität zu Fach-Autorität und von extrinsischer Motivation zu intrinsischer Motivation statt.



Intrinsische und extrinsische Faktoren beschreiben die beiden grundsätzlichen Arten von Motivation. **Intrinsische Motivation** beruht auf selbst bestimmten Faktoren, die jeder Einzelne für sich als wichtig erachtet. Typische intrinsische Faktoren sind das Streben nach verantwortungsvollen und wichtigen Tätigkeiten, Entscheidungsfreiheiten, persönlichen Entwicklungsmöglichkeiten und interessanten Arbeitsinhalten.

Extrinsische Faktoren werden von Dritten (Vorgesetzte, Personalabteilung) mit dem Ziel vorgegeben, jemanden zu einem gewünschten Verhalten zu motivieren. Typische Beispiele sind Gehaltserhöhungen, Belobigungen, Beförderungen, aber auch Bestrafungen wie Gehaltsreduzierung oder disziplinarische Maßnahmen. I.d.R. haben extrinsische Motivationsfaktoren einen stärkeren, aber kurzfristigeren Effekt, während intrinsische Faktoren eher eine langfristige Wirkung zeigen." [Rec01]

Der **komplexe Mensch** ist hochgradig lernfähig. Er erwirbt im Arbeitsprozess neue Motive und in unterschiedlichen Systemen werden unterschiedliche Motive bedeutsam. Manager sind Diagnostiker von Situationen. Sie müssen Unterschiede erkennen und eigenes Verhalten situationsgemäß variieren können.



Die "Einstufung" Ihrer Mitarbeiter in einen dieser Menschentypen setzt voraus, dass Sie sich mit Ihren Leuten auseinandersetzen und sie wichtig nehmen. In Gruppen äußern sich die unterschiedlichen Typen durch die Besetzung von **Rollen**. Da gibt es

z.B. den Macher, den Tüftler, den Selbstdarsteller oder den Schlichter. Es wäre sicherlich zu einfach, die Mitarbeiter ausschließlich einem Typ zuzuordnen oder auf eine Rolle festzulegen. Ein Mitarbeiter kann durchaus in einer Verhandlung ein Schlichter, am Arbeitsplatz aber ein Tüftler sein. Menschen beinhalten Komponenten aller beschriebenen Typen in unterschiedlichen Ausprägungen – welche Rolle zum Tragen kommt, hängt neben der Veranlagung auch von der Situation und den Verhältnissen im Team ab. Das eröffnet aber auch die Chance der Einflussnahme durch den Leiter.

Sie sollten dafür sorgen, dass eine gesunde Teamzusammensetzung vorhanden ist. Das heißt, dass die Rollen in einem vernünftigen Verhältnis aufgeteilt sind. Auch in harten Diskussionen sollte es ein Mitglied geben, das beruhigt (sozialer Typ). In Kreativitätsveranstaltungen sollte jemand dabei sein, der machbare Vorschläge vorbringt (rationaler Typ). Versuchen Sie der Tendenz zum Übergewicht bestimmter Rollen entgegenzuwirken.

6.1.3 Einige Beispiele für Motivationstheorien

Am Anfang dieses Kapitels soll D e n k - würdiges von REINHARD K. SPRENGER stehen:

- Alle Motivierung zerstört die Motivation.
- Die Motivierung ist die massenhafte Verführung zur inneren Kündigung.
- Das System der Motivierung ist methodisiertes Misstrauen. Das Menschenbild der Motivierung ist ... Menschen sind tendenziell Leistungsverweigerer, Menschen sind hierarchisch gestaffelte Bedürfnisbündel, Menschen sind Reiz-Reaktions-Maschinen.
- "Und wie ist es dir gelungen, ihn fertigzumachen?" "Durch Lob..." (Kishon).
- Diese von ihrer Aufgabe Besessenen (und nicht die Profilierungsakrobaten) sind in Wahrheit die Stützen des Unternehmens.
- Anerkennung ist eine Pflanze, die vorwiegend auf Gräbern wächst.
- Die Bonus-Praxis in den Unternehmen ist die Krankheit, für deren Heilung sie sich hält.
- Der Rückgriff auf selbstregelnde Anreizsysteme ist der Offenbarungseid der Führungskraft.
- Die Motivation jedes einzelnen ist die natürliche Ordnung der Dinge.
- "Führen durch Zielvereinbarung" heißt es; "Führen durch Zielvorgabe" ist es zumeist.
- Leistungsbereitschaft kann man nur behindern.
- Führen ist vor allem das Vermeiden von Demotivation.

- Im Unternehmen gibt es zwei Menschentypen – signierende und resignierende.
- Wenn Sie Mitarbeiter für unselbständig halten, werden sie es sein.
- Abteilung kommt von abteilen.
- Alles was Menschen wollen, ist wählen können.
- Ron Sommer über erfolgreiches Innovationsmanagement: "Pick a team from among the company misfits, and give them 100% freedom."
- "Das häufigste Vergehen im Wirtschaftsleben ist die fundamentale Missachtung der Menschenwürde." [Spr14]

So verwirrend wie diese Aussagen – so verwirrend sind auch die immer wieder neuen und unzähligen Theorien zur Motivation der Mitarbeiter.



Unter Motivationstheorien versteht man theoretische Aussagensysteme, die das erkennbare Verhalten der Mitarbeiter zu erklären versuchen.

Wie umstritten, aber gleichzeitig notwendig solche Theorien sind, zeigt die folgende Aussage von D. RECKLIES: "Wir sind es längst gewohnt, dass uns alle paar Jahre ein neues Konzept des Human Resource Management vorgestellt wird. Diese sollten – auch wenn der Gedanke nahe liegt – nicht unbedingt sofort als Modewellen abgetan werden. In der Regel entstehen neue Managementkonzepte, wenn die vorhandenen Instrumente geänderten Bedingungen nicht mehr ausreichend gerecht werden. In Zeiten relativ kurzer Lebenszyklen von Produkten, Geschäftsmodellen, Verfahren der Leistungserbringung, aber auch sozialer und gesellschaftlicher Werte muss auch das Personalwesen zeitnah neue Konzepte anbieten, die den geänderten betrieblichen Anforderungen und den geänderten Erwartungen der Mitarbeiter gleichermaßen gerecht werden. Es ist eine andere Seite der Medaille, dass nicht jedes neue Instrument hält, was es nach Ankündigung seiner Entwickler verspricht." [Rec01]

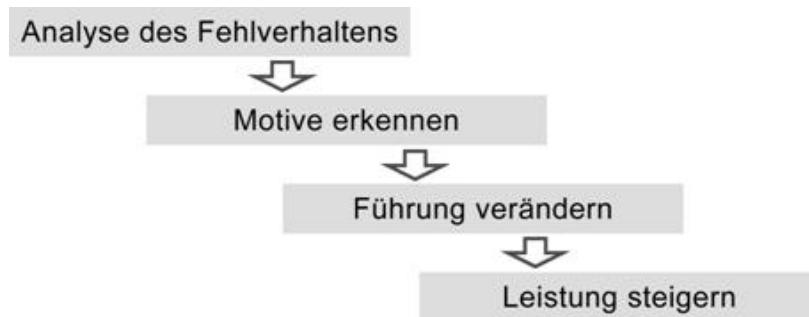
An dieser Stelle werden die klassischen Theorien, die sich in der Praxis bewährt haben, vorgestellt:

- Individualpsychologischer Ansatz (ADLER, 1920)
- Dynamische Motivationstheorie (ABRAHAM MASLOW, 1954)
- Zweifaktoren-Theorie (FREDERIK HERZBERG, 1966)

Im wesentlichen sind nachfolgend die Aussagen nach [Arm99] und [Ans95] zusammengefasst.

Individualpsychologischer Ansatz

These: Das Fehlverhalten ist umso stärker, je schwächer das Selbstwertgefühl ist bzw. je weiter die Motive vom Verlangten abweichen.



■ Abbildung 6.1.3.1: Übersicht zum individualpsychologischen Ansatz

Die Analyse des Fehlverhaltens kann durch Beobachtungen vorgenommen werden. Adler unterscheidet zwischen leichtem und schweren Fehlverhalten, das sich wie in Abbildung 6.1.3.2 dargestellt äußert.



■ Abbildung 6.1.3.2: Einstufung von Fehlverhalten

Die Lehre, die sich aus dieser Theorie ziehen lässt, ist z.B. der folgende Hinweis:

Nicht pauschal abstrafen, sondern Fehlverhalten hinterfragen und analysieren. Arbeitsumfeld oder Arbeitsaufgabe oder beide Faktoren stimmen nicht mit den Vorstellungen des Mitarbeiters überein. Er erwartet z.B. soziale Kontakte und gerät in eine Atmosphäre von Ich-bezogenen Kollegen, die nur Geld- und Karriere-Interessen haben.

Dynamische Motivationstheorie

These: Alle beim Menschen auftretenden Verlangen lassen sich auf fünf Grundbedürfnisse zurückführen, die sich auf Grund ihrer unterschiedlichen Dringlichkeiten in eine hierarchische Ordnung bringen lassen.



Abbildung 6.1.3.3: Bedürfnispyramide nach Maslow

Die Bedürfnisse lassen sich zwischen Mangel- und Wachstumsbedürfnissen (siehe Abb. 6.1.3.3) unterscheiden. Das Bedürfnis der Selbstverwirklichung wird stark betont: "...fortschreitende Verwirklichung der Möglichkeiten, Fähigkeiten und Talente als Erfüllung einer Mission oder einer Berufung, eines Geschicks, eines Schicksals, eines Auftrags..."; "Er kommt dem Kern seines Seins näher, wird menschlicher" (MASLOW).

Diese Gedanken sind vor allem auch für die Mitarbeiter in Softwareprojekten – im Allgemeinen mit hohem Bildungsniveau – zutreffend. Vielen Mitarbeitern geht es ganz eindeutig um ihre Selbstverwirklichung.

Gewähren Sie daher Freiräume und bieten Sie Autonomie an!

Zwei-Faktoren-Theorie

These: Es gibt zwei Gruppen von Faktoren:

- Motivationsfaktoren, die erhalten und gefördert werden müssen, und
- "schlechte Hygiene" faktoren, die vermieden werden müssen.



Abbildung 6.1.3.4: Zwei-Faktoren-Theorie

Sorgen Sie daher für gute Arbeitsbedingungen, versuchen Sie die Inhalte des Projektes und der Aufgaben sowie der Lösungsansätze zu verstehen und bilden Sie sich weiter. Stärken Sie den Teamgeist und bieten Sie Entwicklungsmöglichkeiten für die Mitarbeiter!

6.1.4 Aufgaben zu Motivationstheorien



Aufgabe

Aufgabe 6.1.4-1

Die Verhaltensbereitschaft eines Mitarbeiters wird von verschiedenen Faktoren bestimmt. Gruppieren Sie dazu die folgenden Begriffe und begründen Sie Ihre Auswahl kurz!

Anreize – Motive – Fähigkeiten – Arbeitsbedingungen – Einstellungen



Aufgabe

Aufgabe 6.1.4-2

Welche Rückschlüsse lassen sich aus der "Menschentypforschung" für die Rollenverteilung in Gruppen ziehen?



Aufgabe

Aufgabe 6.1.4-3

Welche Gefahren sehen Sie, wenn die Rolle "Führer" überbesetzt ist und welche Maßnahmen können eingeleitet werden?



Aufgabe

Aufgabe 6.1.4-4

Aus welcher Motivationstheorie kann die Empfehlung zur Hinterfragung und Analyse von Fehlverhalten der Mitarbeiter abgeleitet werden?

- A) Individualpsychologischer Ansatz
- B) Dynamische Motivationstheorie
- C) Zweifaktoren-Theorie

6.2 Führungshinweise



6.2 Führungshinweise

6.2.1 Führungstheorien

Gliederung

- [6.2.2 Teamentwicklung](#)
- [6.2.3 Praktische Hinweise](#)
- [6.2.4 Aufgaben zu Führungshinweise](#)

Dieser Abschnitt beschäftigt sich mit der Anleitung Ihres **Teams**. Dazu sollen einige Aspekte beleuchtet werden. Von einem Team spricht man ja ganz allgemein, wenn mehrere Personen versuchen, ein gemeinsames Ziel zu erreichen und diese Personen voneinander abhängig sind. Es gibt unterschiedliche Arten von Teams:

- Face-to-Face-Gruppen, arbeiten physisch zusammen am gleichen Ort
- Verteilte Gruppen, es wird örtlich getrennt gearbeitet
- Kleingruppen, 3 bis 7 Mitglieder
- Großgruppen, viele Mitglieder mit der Bildung von Hierarchien

Diese Arten erfordern unterschiedliche Methoden der Führung. Wir werden uns in den folgenden Ausführungen besonders auf Face-to-Face-Gruppen mit Kleingruppencharakter konzentrieren, da solche Arbeitsgruppen oft Auslöser von Krisen oder Problemen sind und am häufigsten auftreten. Teams zeichnen sich dadurch aus, dass sie für eine bestimmte Aufgabe temporär – oft auch abteilungsübergreifend – zusammengestellt werden. Dabei will man ganz bewusst vorhandenes Knowhow nutzen. Die Teamzusammensetzung und damit die Mitarbeiterauswahl (vgl. Abschnitt 3.1.3 Personal) ist ein ganz wichtiger Punkt, damit nicht durch innere Reibungen Vorteile verloren gehen.

6.2.1 Führungstheorien

In der Führungslehre gibt es **Führungstheorien**, die Hinweise auf richtiges Verhalten geben können. Die Führungstheorien lassen sich z.B. ordnen nach den Elementen, die in den Mittelpunkt gestellt werden:

- Führer→Eigenschaftenansatz,
- Geführte→Verhaltensansatz (Führungsstilforschung),
- Situation→Situationstheorie (abhängig vom Verhalten des Geführten und der Situation).

Der **Eigenschaftenansatz** geht davon aus, dass die Führung durch besonders ausgeprägte Eigenschaften des Führenden zustande kommt, die situationsunabhängig – aufgabenunabhängig – gruppenunabhängig wirksam sind.

Erster Ansatz: Führungspersönlichkeiten besitzen ein einzigartiges, zeitlich stabiles Persönlichkeitsmerkmal, das sie von anderen Personen (Geführten) abhebt. Dem widerspricht aber, dass sich kein einheitliches Merkmal bei Führungspersönlichkeiten finden lässt, das ihren Erfolg begründet.

Zweiter Ansatz: Die Führungsqualifikation hängt von einem bestimmten Eigenschafts- oder Merkmalskomplex ab (z.B.: Intelligenz, Kontaktfähigkeit, Durchsetzung). Dem widerspricht, dass kein Merkmalsbündel ein erfolgreiches Verhalten in Führungsrollen (zu große Streuung) garantiert.

Die Theorie entspricht Wunschvorstellungen, wird aber dennoch besonders in Krisensituationen verfolgt.

Stattdessen verfolgte man im **Verhaltensansatz** die Untersuchung unterschiedlicher Führungsstile. Man unterscheidet dabei nach:

- **autoritärem Stil:**

- der Führer ist Herr
- der Geführte ist Untergebener
- der Konkretisierungsgrad der Aufgabe ist hoch
- der Führer trifft Einzelentscheidungen



- **kooperativem Stil:**

- der Führer ist Lenker und Koordinator
- die Geführten sind Mitarbeiter und Partner
- nur Rahmenregeln für die Aufgabenerfüllung
- Entscheidungskollegien – die Mitarbeiter werden einbezogen



Es gibt nicht den richtigen Führungsstil. Untersuchungen von KUPPER [Kup96] haben ergeben, dass kooperativer und autoritärer Führungsstil ungefähr gleich gut ankommen. Der kooperative Führungsstil bringt den Unternehmen Vorteile: Mitarbeiter setzen ihre Phantasie und Arbeitskraft für das Unternehmen ein. Für beide Stilarten ist ein entscheidender Punkt das Engagement des Vorgesetzten – laissez faire-Einstellungen (kein Interesse, Laufenlassen, Gleichgültigkeit) kommen bei Mitarbeitern sehr schlecht an [Kup96].

Die **Situationstheorie** geht davon aus, dass Führung bzw. die Position des Führers nicht ausschließlich von Persönlichkeitseigenschaften abhängt, sondern auch und gerade von spezifischen Situationen. Führung passiert also in Interaktion zwischen Eigenschaften und günstigen Situationskonstellationen. Folgende Variablen haben Einfluss im Modell der Situationstheorie:

- Gruppenmitglieder (Fähigkeit, Anspruch, Erwartung)
- Aufgabenstellung der Gruppe – Ziele müssen vorhanden sein – Führer muss sich zielrelevant verhalten
- Externe Einflüsse auf die Gruppe (z.B. Gesellschaft)
- Art der Gruppenstruktur
- Gruppennormen

An dieser Theorie wird kritisiert, dass bereits das Wahrnehmen einer Situation gesellschaftlichen Normen entspricht und daher verschieden ausfällt. Die Situationstheorie wird deshalb zu einer Lehrformel, die für alle möglichen Ergebnisse Lösungen liefert. Bislang ist auch ungeklärt geblieben, wie die einzelnen Situationsaspekte zueinander in Beziehung stehen.

Die Führungstheorien geben noch keine konkreten Hinweise für die tägliche Arbeit mit den Projektteams. Sie bilden einen Rahmen, aus dem solchen Hinweise abgeleitet werden können. Viele praktische Anregungen für die Leitung von Projekten habe ich einem Vortrag von HEDWIG KELLNER auf der Advanced Developers Conference [Kel01] entnommen, auf den sich die folgenden Ausführungen im Wesentlichen stützen. Ihre Erfahrungen decken sich auch mit meinen. Ihr Ausgangspunkt ist die Einordnung des Projektteams in Phasen und deren Entwicklung.

6.2.2 Teamentwicklung

Neue Teams müssen sich entwickeln und benötigen dazu Förderungen. Eine gezielte Förderung setzt das Wissen um den Entwicklungsstand voraus. In [Kel01] werden fünf typische Phasen der Teamentwicklung unterschieden:

1. Orientierung
2. Frust
3. Akzeptanz
4. Routine
5. Auflösung

Das folgende umfangreiche Zitat ist mit Einverständnis der Autorin [Kel01] entnommen.

"Viele Führungskräfte wissen nicht genau, ob ihr Team bereits gut funktioniert und sich die Teammitglieder lediglich in gesunder Weise immer wieder einmal aneinander reiben oder ob ihr Team in Wirklichkeit unterschwellige Probleme wälzt und deshalb in seiner Effektivität behindert ist. Der erste Schritt besteht für Sie darin zu erkennen, in welcher Phase sich Ihr Team befindet."

In der **Orientierungsphase** kann man noch nicht wirklich von einem Team sprechen. Die Gruppe besteht vielmehr aus einer Ansammlung von Individuen, die noch nicht zusammengefunden haben. Der Einzelne weiß noch nicht genau, wie er die anderen zu nehmen hat und wie er selbst von den anderen gesehen wird. Rollenverteilungen sind noch völlig offen.



Es gibt **6 Merkmale**, an denen Sie erkennen können, ob sich die Gruppe noch in der Orientierungsphase befindet:

1. Extrovertierte treten übertrieben lässig und scheinbar extrem selbstbewusst auf. Sie fürchten, ihnen könnten ihre Ängste angemerkt werden.

2. Introvertierte äußern sich nur sehr zögerlich mit Vorschlägen oder Meinungen. Sie halten sich zurück, weil sie nicht wissen, wie man ihre Beiträge aufnehmen wird.
3. Alle Mitarbeiter suchen immer wieder den Blickkontakt zu Ihnen. Sie wollen von Ihnen, dem Leiter der Gruppe, wissen, was zu tun ist und wie Sie sich verhalten sollen.
4. In Ihrer Abwesenheit kann die Gruppe nicht selbstständig arbeiten. Extrovertierte diskutieren endlos herum, während die Introvertierten sich in ihre Einzelarbeiten zurückziehen.
5. Insgesamt herrscht eine Stimmung der Unsicherheit vor. Keiner weiß, wer eigentlich wofür genau zuständig ist. Man kann noch nicht miteinander umgehen, und es kommt zu Missverständnissen bis hin zu verletzten Gefühlen.
6. Die Gruppe hat die gemeinsame Aufgabe noch nicht so richtig im Griff. Keiner weiß, wie er die Sache anpacken soll. Alle verlangen von Ihnen, dass Sie "als Chef" klare Anweisungen geben, und sie reagieren dann doch sofort empfindlich auf Anzeichen von "autoritärem" Verhalten. Das wiederum macht Sie unsicher. Sie fragen sich: "Was wollen meine Mitarbeiter eigentlich?"

Auch in der **Frustphase** bestimmen heftige Gefühle das Klima. Diese Gefühle sind jedoch fast alle negativ und werden sehr deutlich gezeigt.



Sie erkennen die Frustphase an häufigen Reibereien bis hin zu Streitigkeiten um Vorgehensweisen bei der Arbeit, um Kompetenzen und andere Dinge, von denen jeder eigentlich meinen sollte, dass vernünftige Menschen sie friedlich klären könnten. Die Mitarbeiter sprechen viel von früheren Projekten oder Jobs, bei denen angeblich alles viel besser war. Rechnen Sie auch damit, dass robuste Mitarbeiter Sie persönlich in Ihrer Führungsrolle angreifen.

Verstehen Sie bitte die Frustphase als ganz normalen Prozess der gesunden Teambildung. Die Streitereien und Angriffe sollten Sie nicht überbewerten. In der Gruppe finden oft unbewusste Machtkämpfe um Rollenverteilungen statt. Der Versuch, unbedingt

Harmonie schaffen zu wollen, würde zu gefährlichen verdeckten Konflikten führen. Diese können dann dauerhaft das Klima vergiften. Die Frustphase ist unangenehm, kostet viel Kraft, muss aber sein. Auch durch die Konflikte der Frustphase lernen die Mitarbeiter sich kennen. Sie wissen nun, wer worauf empfindlich reagiert, wer sich durchsetzen kann, wen Sie wie nehmen sollten.

In der **Akzeptanzphase** bildet sich das "Wir-Gefühl". Das Team ist noch keine verschworene Gemeinschaft, erlebt sich jedoch bereits deutlich als eine Einheit im Unterschied zu anderen Gruppen oder Abteilungen. Die Rollen finden sich. Die Mitarbeiter werden mit der Aufgabe vertrauter.



Sie erkennen die Akzeptanzphase daran, dass Meinungsverschiedenheiten zwar immer wieder den Fortschritt der Arbeit aufhalten, dass die Mitarbeiter aber nicht mehr nur streiten, sondern durch Diskussionen mit erkennbarem Einigungswillen zu Ergebnissen kommen. Ein weiteres typisches Merkmal ist der lockere Umgangsstil miteinander. Während jeder in der Frustphase noch sofort verärgert reagierte, nehmen die Mitarbeiter es jetzt mit Humor, wenn sich im Eifer eines Wortwechsels mal jemand im Ton vergreift. Jeder kennt sich inzwischen und weiß, wie es gemeint ist.

In der **Routinephase** ist das Team eine verschworene Gemeinschaft und kann und will gemeinsam erfolgreich sein. Die Mitarbeiter gehen menschlich miteinander um und sind fachlich mit der Aufgabe vertraut.



Sie erkennen diese Phase daran, dass Ihr Team nun weitestgehend selbstständig arbeiten kann und will. Jeder weiß, was er zu tun hat. Alle helfen sich gegenseitig, und jeder ist daran interessiert, dass alle zusammen die Teamziele erreichen.

Die **Auflösungsphase** tritt zum Ende eines Projekts ein oder auch wenn ein neues Mitglied dazukommt oder wenn ein oder mehrere Mitarbeiter ausscheiden. In den letztgenannten Fällen befindet sich das Team in einer vorübergehenden Auflösung und beginnt anschließend wieder mit der Orientierungsphase. Erkennungsmerkmale sind, dass sich Einzelne bereits mit anderen Aufgaben befassen. Es wird ein anderer Tonfall angeschlagen, er ist distanzierter. Für Rituale ist plötzlich keine Zeit mehr. Die Späße werden als albern und unnötig abgetan.

Fördern Sie die gruppendifynamische Entwicklung zu einem Erfolgsteam

Als Führungskraft wollen Sie natürlich Ihr Team möglichst schnell durch die Phasen der Orientierung und des Frusts hin zu Akzeptanz und dann schließlich zur Routine führen, um es dann möglichst lange dort zu halten.

Erst in der Routinephase ist Ihr Team wirklich erfolgreich, steuert sich weitgehend selbst und kann Ihnen Freiräume für Ihre anderen Aufgaben lassen. Vor allem während der ersten beiden Phasen gehen bis zu 70% der Arbeitszeit für Einarbeitung, unkoordinierte Doppelarbeiten, Fehler und Konflikte verloren. Wenn Sie nicht genau hinschauen, kann es Ihnen wie vielen Führungskräften gehen: Sie bemerken es nicht einmal, wie viel Zeit die Mitarbeiter damit verbringen, sich untereinander über ihren Frust zu beklagen oder sich gegenseitig von angeblich viel besseren Erfahrungen in anderen Teams zu berichten.

Überlassen Sie die Teamentwicklung nicht dem Zufall, achten Sie dabei besonders auf Veränderungen. Die erste Regel lautet "Never change a winning Team".

Drei Führungs-Tipps für die Orientierungsphase:

1. Arbeiten Sie in kreativen Workshops gemeinsam mit der ganzen Gruppe. Wenden Sie Techniken wie Mind Mapping oder Kartenabfragen an. So lernt das Team den Umgang miteinander und erlebt gleichzeitig gemeinsam die Freude an kreativen Prozessen.
2. Sie sollten in dieser Phase stets die Workshops Ihres Teams selbst moderieren. Sie etablieren sich damit als Führungsfigur und entwickeln ein Gespür für die Gruppe und für die Individuen.
3. Achten Sie jetzt sehr auf Disziplin, vor allem auch in kleinen Dingen. Warten Sie bei Ihren Workshops nicht auf Zuspätkommer, sondern fangen Sie pünktlich auf die Minute an.

Drei Führungs-Tipps für die Frustphase:

1. Versuchen Sie nicht in konfliktären Diskussionen Harmonie anzumahnen. Lassen Sie die harten Auseinandersetzungen zu. Ihre Mitarbeiter wollen sich zusammenraufen und austesten, wie weit sie miteinander gehen können. Auch bei leiden-

schaftlichen Wortgefechten kommt es fast nie zu Verletzungen. Verpflichtung zur Harmonie in dieser Phase verlagert die Konflikte in den Untergrund! Das fördert Heimtücke und Intrigen als schleichende Konflikte.

2. Sorgen Sie für einen schnellen und sichtbaren Erfolg. Kluge Projektleiter lassen ihre Mitarbeitern jetzt den ersten Meilenstein erreichen. Dadurch erlebt die Gruppe, dass sie gemeinsam etwas auf die Beine stellen kann. Außerdem ermutigt es sie, dass sie dem gemeinsamen Ziel einen erkennbaren Schritt näher gekommen ist. Das steigert die Leistungsbereitschaft.
3. Sorgen Sie dafür, dass der erste Erfolg Ihres Team Außenstehenden bekannt wird. Organisieren Sie eine Präsentation oder lassen Sie einen Artikel für die Mitarbeiterzeitung schreiben. Ihr Team erlebt dadurch die Freude: "Wir machen uns im Unternehmen als Erfolgsteam einen Namen!" Das schweißt die Gruppe enger zusammen und steigert ebenfalls die Leistungsbereitschaft.

Drei Führungs-Tipps für die Akzeptanzphase:

1. Geben Sie langsam von Ihren Führungsaufgaben ab. Lassen Sie Mitarbeiter, die Lust dazu haben, die Meetings moderieren. Sie fördern damit die Fähigkeit zur Selbststeuerung des Teams. Außerdem schaffen Sie sich Freiräume für andere Aufgaben.
2. Wenn Ihr Team recht groß ist, sollten Sie jetzt auch Teilgruppen einrichten. Die Mitarbeiter wissen dabei selbst am besten, wer mit wem gut arbeiten kann. Sie fördern damit die Bildung sehr effektiver Kleingruppen, in denen sich keiner verstecken kann. Wo zwei oder drei Mitarbeiter gemeinsam an einer Sache arbeiten, lässt keiner die Kollegen hängen. Ab einer Gruppengröße von vier Personen kommt es immer wieder vor, dass sich auch mal ein Mitarbeiter mitschleifen lässt oder sich hinter den anderen versteckt.
3. Rufen Sie regelmäßig alle Mitarbeiter zu einer gemeinsamen Teambesprechung zusammen. Damit fördern Sie den Zusammenhalt als ganze Gruppe und verhindern, dass sich Teilgruppen oder Einzelpersonen in ihrer Arbeit zu weit von der gemeinsamen Marschrichtung entfernen.

Zwei Führungs-Tipps für die Routinephase:

1. Lassen Sie sich gern in Ihren Führungsaufgaben von dem Teammitglied entlasten, das im Kollegenkreis als "Wortführer" oder "inoffizieller Leader" Anerkennung gefunden hat. Da sich die anderen freiwillig unterordnen, ist das Teamklima von Harmonie und Konzentration auf die gemeinsame Aufgabe bestimmt. Beachten Sie: Unter einem "sich selbst steuernden Team" versteht man ein Team, in dem alle Mitglieder ihre jeweilige Rolle gefunden haben und alle gemeinsam sich frei-

willig von der Person aus der eigenen Mitte lenken lassen, die über die größte natürliche Autorität verfügt.

2. Nutzen Sie Ihre Freiräume zur Herstellung und Pflege von Außenkontakte. Lassen Sie das Team mit Mitarbeitern anderer Bereiche kommunizieren. Unterstützen Sie den Info-Austausch und gerne auch einmal den Austausch von "Gastarbeitern", zum Beispiel bei Workshops. Lassen Sie Ihre Mitarbeiter auch an Workshops anderer Gruppen teilnehmen. Holen Sie Kollegen anderer Bereiche zu Ihren Meetings. Damit verhindern Sie "Bunkermentalität" im Team mit der Neigung, sich von anderen Bereichen abzuschotten. Stattdessen fördern Sie das Ansehen Ihres Teams als sympathische und professionelle Erfolgsmannschaft im Unternehmen. Je stolzer die Mitarbeiter darauf sind, zu Ihrem Team zu gehören, desto motivierter gehen sie daran, Ihren Ruf durch Spitzenleistungen immer wieder zu bestätigen!

Motivieren Sie durch Befriedigung der drei wichtigsten Grundausrichtungen des menschlichen Verhaltens

Wenn Sie Ihre Mitarbeiter erfolgreich in die Routinephase geführt haben, sollten Sie kontinuierlich immer wieder darauf achten, dass jeder Einzelne die Chance bekommt, die drei wichtigsten Grundausrichtungen des menschlichen Verhaltens im Beruf zu befriedigen. Dies sind:

1. Der Wille, gute Arbeit zu leisten. Ihre Mitarbeiter müssen Sie gar nicht mit psychologischen "Tricks" motivieren. Sie haben selbst Lust, sich mit interessanten Aufgaben zu beschäftigen, anspruchsvolle Ziele zu erreichen, Wichtiges zu tun.
2. Die Lust, sich in eine Gemeinschaft einzugliedern. Wir Menschen sind soziale Wesen. Wir begeben uns gerne in Gesellschaft, tauschen uns mit anderen aus und möchten uns in einem Kreis fester Beziehungen anerkannt und geborgen fühlen.
3. Der Wunsch nach Individualität. Wir möchten uns auch als Individuen abgrenzen können und manches ganz allein entscheiden oder gestalten. Wir sehnen uns danach, ganz allein auf etwas stolz sein zu dürfen, was wir ohne die Hilfe anderer erreicht haben. Und wir wollen Privateigentum oder zumindest einen eigenen Bereich, den wir uns so ausgestalten dürfen, wie es uns persönlich am besten gefällt. Außerdem brauchen wir bei aller Freude an der Gemeinsamkeit immer wieder auch Rückzugsmöglichkeiten.

Um die drei wichtigsten Grundausrichtungen für alle möglichst gut zu befriedigen und um unnötige Störungen zu vermeiden, sollten Sie mit Ihren Mitarbeitern **Spielregeln** für die Zusammenarbeit entwickeln. Organisieren Sie den ersten Workshop für die "Spielregeln" in der Akzeptanzphase. Die Mitarbeiter kennen sich inzwischen und sind bereit, sich nach der Frustphase selbst durch solche Regeln eine Ordnung zu geben.

Wichtig ist auch, dass die "Spielregeln" bei Bedarf geändert oder ergänzt werden können. Mehr als 10 Regeln sollten es jedoch nicht sein. Ob Ihre Mitarbeiter sich mit den gemeinsamen Spielregeln wohl fühlen, erkennen Sie daran, dass diese nicht regelmäßig in Frage gestellt werden und dass nicht immer wieder über Vorgehens- und Verhaltensweisen diskutiert wird.

Hier Beispiele für Spielregeln:

- Wir bringen geliehene Unterlagen dem Kollegen unaufgefordert sofort zurück.
- Wer die Tür hinter sich schließt, bleibt – außer in Notfällen – ungestört.
- Wir klären unsere Meinungsverschiedenheiten und Konflikte unter uns.
- Die ersten beiden Tage nach jeder Meilensteinabnahme bleiben von Terminen frei.
- Wir nutzen diese Tage, um Liegengeliebenes vom Tisch zu arbeiten.
- Wir akzeptieren uns in unseren unterschiedlichen Meinungen und Temperaturen. Jeder von uns bemüht sich, mit den Kollegen letztlich zur Einigung zu kommen.
- Keiner von uns legt auf die Goldwaage, was in heftigen Diskussionen rausrutschen kann.
- Jeder von uns ist freiwillig an diesem Projekt beteiligt und setzt sich mit ganzer Kraft für den Erfolg ein. Wer den unbedingten Willen zum Erfolg nicht mehr hat, steigt aus dem Projekt aus.

Sorgen Sie für ein gesundes Team

In **gesunden Teams** gibt es ein ausgewogenes Verhältnis der unterschiedlichen Menschentypen (vgl. Abschnitt 6.1). Auch in diesen Teams treten Meinungsverschiedenheiten, Eifersüchteleien oder Streit auf. Von **kranken Teams** wird erst dann gesprochen, wenn dauerhafte Störungen auftreten und der Leistungspegel sinkt. Ausdruck findet das oft in Formen von Mobbing und Intrigen. Es gibt Auflösungerscheinungen. Wie sollte man sich in so einem Fall verhalten?

Sie können die Häufigkeit der Meetings reduzieren, damit sich die Gemüter beruhigen. Die Aufgabenfelder sind zu überarbeiten und sollten streng abgetrennt sein, um das Konfliktpotential zu reduzieren. Ermöglichen Sie einen sichtbaren Erfolg durch die Verlegung von Meilensteinen. Falls die Konflikte immer um *einen* Mitarbeiter entstehen, muss dieser Mitarbeiter aus dem Team genommen werden. Das gilt auch dann, wenn diese Person eigentlich unschuldig ist. Definieren Sie neue Spielregeln.

Verhindern Sie schleichende Konflikte. Diese entstehen oft durch eine übertriebene Teamharmonie, dort, wo offene Konflikte tabuisiert sind. Es besteht die Gefahr, dass negative Gefühle hinter Masken verborgen werden. Hinter den Masken gibt es dann Misstrauen, Neid und Ärger.

6.2.3 Praktische Hinweise

Die folgenden Hinweise sind aus [Kel01] entnommen – Kurzanleitung für Ihre erfolgreiche Teamführung:

1. Unterscheiden Sie die fünf Phasen der gesunden Teamentwicklung: Orientierung, Frust, Akzeptanz, Routine, Auflösung.
2. Fördern Sie in der Orientierungsphase durch gemeinsame Workshops das Zusammenwachsen der Gruppe.
3. Legen Sie in der Orientierungsphase durch Disziplin die Basis für einen dauerhaft leistungsorientierten Arbeitsstil.
4. Fördern Sie in der Frustphase durch sichtbare Erfolge die zügige Entwicklung zu einer motivierten Gemeinschaft der Mitarbeiter.
5. Fördern Sie in der Akzeptanzphase durch Delegation von Zuständigkeiten und Verantwortungen die Selbststeuerfähigkeit des Teams.
6. Fördern Sie in der Akzeptanzphase durch regelmäßige gemeinsame Besprechungen den festen Zusammenhalt der Teammitglieder.
7. Fördern Sie in der Routinephase durch Pflege der Außenkontakte die Vernetzung und das hohe Ansehen Ihres Teams im Unternehmen.
8. Motivieren Sie Ihre Mitarbeiter durch Befriedigung der drei wichtigsten Strebungen: Freude an hochwertiger Arbeit, Wohlfühlen in der Gemeinschaft, Recht auf Individualität.
9. Entwickeln Sie gemeinsam mit Ihren Mitarbeitern Spielregeln für die Zusammenarbeit.
10. Achten Sie bei dauerhaft gereiztem Klima auf "Teamkrankheiten".
11. Fördern Sie Heilungsprozesse durch Möglichkeiten zur Distanz und durch sichtbare Erfolge.
12. Bannen Sie die Gefahr von schleichenden Konflikten durch Zulassen von gesunden, offenen Auseinandersetzungen.
13. Lösen Sie Konflikte zwischen Mitarbeitern durch gesteuerte Aussprachen und initiierte erste Zugeständnisse auf.

6.2.4 Aufgaben zu Führungshinweise



Aufgabe



Aufgabe 6.2.4-1

In der Literatur wird der Verhaltensansatz unter den Führungstheorien empfohlen. Nennen Sie zwei weitere Theorien und Vorteile des Verhaltensansatzes gegenüber den anderen Theorien!

**Aufgabe****Aufgabe 6.2.4-2**

Neue Teams müssen sich entwickeln. Nennen Sie die 5 Entwicklungsstufen nach KELLNER!

**Aufgabe****Aufgabe 6.2.4-3**

In welcher Phase (nach KELLNER) befindet sich die folgende Gruppe: Alle Mitarbeiter suchen immer wieder den Blickkontakt zu Ihnen. Sie wollen von Ihnen, dem Leiter der Gruppe, wissen, was zu tun ist und wie Sie sich verhalten sollen. In Ihrer Abwesenheit kann die Gruppe nicht selbstständig arbeiten. Extrovertierte diskutieren endlos herum, während die Introvertierten sich in ihre Einzelarbeiten zurückziehen. Insgesamt herrscht eine Stimmung der Unsicherheit vor. Keiner weiß, wer eigentlich wofür genau zuständig ist. Man kann noch nicht miteinander.

**Aufgabe****Aufgabe 6.2.4-4**

Nennen Sie Maßnahmen zur Heilung kranker Teams!



Gliederung

7 Anhang

7 Anhang

- 7.1 Beispiel zu Kostenvergleichs-, Gewinnvergleichs- und Amortisationsrechnung
- 7.2 Beispiel zur Nutzwertanalyse
- 7.3 Beispiel für einen Projektvertrag zur Erstellung einer Animation
- 7.4 Formular Projektauftrag (manuelle Bearbeitung)
- 7.5 Formular Projektauftrag (computergestützt)
- 7.6 Use-Case Schablone
- 7.7 Beispiel zur Projektstrukturplanung
- 7.8 Planungshinweise für das XP-Prozessmodell
- 7.9 Beispiel zur Netzplantechnik
- 7.10 Beispiele für Checklisten

7.1 Beispiel zu Kostenvergleichs-, Gewinnvergleichs- und Amortisationsrechnung

Kostenvergleichsrechnung

Kosten	Beibehalten der alten Technologie	Einführung der neuen Technologie
Projektkosten pro Jahr bei linearer Abschreibung auf 5 Jahre (Gesamt 140000,-)	0,-	28000,-
Betriebskosten pro Jahr	75000,-	45000,-
Sonstige Kosten pro Jahr	3000,-	5000,-
Gesamtkosten pro Jahr	78000,-	78000,-

Wenn man ausschließlich die Kosten betrachtet, ergibt sich keine Einsparung und somit auch noch kein überzeugendes Argument für das Projekt.

Gewinnvergleichsrechnung

Kosten	Verwendung der alten Technologie	Verwendung der neuen Technologie
Projektkosten pro Jahr bei linearer Abschreibung auf 5 Jahre (Gesamt 140000,-)	0,-	28000,-
Betriebskosten pro Jahr	75000,-	45000,-

Sonstige Kosten	3000,-	5000,-
A Summe Kosten	78000,-	78000,-
B Erträge	100000,-	150000,-
C Gewinn (B - A)	22000,-	72000,-

Wenn man jedoch die zu erwartende Ertragssteigerung einbezieht, fällt durch den höheren Gewinn die Entscheidung für das Projekt positiv aus.

Amortisationsrechnung

	Investitionssumme	Gewinnzunahme pro Jahr	Amortisation in ... Jahren
Neue Technologie	140000,-	50000,-	2,8

Das betrachtete Projekt amortisiert sich in 2,8 Jahren.

7.2 Beispiel zur Nutzwertanalyse

Entscheidungs-kriterium	(A) Wichtigkeit (%)	(B) Wertaus-prägung	Sehr gut (5)	Gut (4)	Ausprägung Neutral (3)	Schlecht (2)	Sehr schlecht (1)	(A)x(B)
Finanz-Risiko	Kein Risiko	Kaum zu erwarten	Unternehmerrisches Risiko	Hoch (Entwicklungsosten nicht abdeckbar)	Entwicklungsosten werden nicht gedeckt
Gewinn-Erwartung	Überdurchschnittlich	Normale Spanne	Knapp unter Durchschnitt	Kosten werden gedeckt	Kosten werden nicht gedeckt
Wettbewerbs-Situation	Neuer Markt kann erschlossen werden	Keine Nachteile gegenüber Wettbewerbern	Bisheriger Markt	Marktverluste	Bedeutende Marktverluste
Technisches Risiko	Keines, Know how vorhanden	Kaum, Know how großteils vorhanden	gering	Überschreiten des bisherigen Knowhows	Aufbau neuen Know hows erforderlich
Wieder-verwendung von Daten	Bestehende Technologien wieder-verwendbar	Bestehende Technologien weitgehend wieder-verwendbar	Ähnliche Technologien vorhanden	Einzelne Elemente wieder-verwendbar	Kaum Unterlagen vorhanden
Kapazitäts-situation	Termingerechte Bearbeitung sicher möglich	Termingerechte Bearbeitung wahrscheinlich möglich	Engpass zu erwarten	Einige zusätzliche Kapazitäten erforderlich	Viele zusätzliche Kapazitäten erforderlich

7.3 Beispiel für einen Projektvertrag zur Erstellung einer Animation

Projektvertrag: Erstellung einer Animation, basierend auf dem Storyboard vom 13.02.2002.

Aufgabenstellung

Erstellung einer Animation auf Basis Ihres Storyboards vom 13.02.02. Die Animation soll webtauglich und bei Bedarf auch als Stand-Alone-Lösung konzipiert sein. Der Umfang von Interaktivität mit dem User wird durch das Storyboard definiert. Eine flexible Lösung (Client-Server-Modell mit Trennung von Interaktivität und Animation) wird nicht entwickelt.

Ansprechpartner

Die Projektleitung auf Seiten des Auftraggebers übernimmt Herr XY und auf Seiten von Grafik GmbH Herr YX. Zur Gewährleistung kurzer Abstimmungswege findet die gesamte Kommunikation über diese beiden Personen statt.

Vorgehensweise

Die Erstellung der Animation erfolgt voraussichtlich in Flash 5 (kostengünstigste Lösung, größte Flexibilität, hoher Verbreitungsgrad des Plugins). Die Produktion erfolgt dabei ohne "Sound", um einerseits den Kostenrahmen nicht zu sprengen ("Billig"-Sprecher klingen tatsächlich abschreckend!) und andererseits die Größe der entstehenden Dateien zu begrenzen. Das Endprodukt können einzelne, miteinander verknüpfte Flashszenen sein (optimal für die Darstellung im Web) oder auch ein einzelner Film, der alle Szenen enthält. Die Entscheidung darüber sollte im Laufe des Projektes von den Projektverantwortlichen auf Basis der bis dahin abschätzbaren Dateigrößen der Flashfilme getroffen werden. Um Absprachen und Korrekturphasen im Hinblick auf das zur Verfügung stehende Budget möglichst unkompliziert zu halten, wird ein erster Teil des Storyboards umgesetzt und zur Ansicht und Freigabe zugesandt. Die weiteren Teile des Storyboards werden nach Freigabe dieser ersten "Demo" umgesetzt. Änderungen der grundlegenden Gestaltung oder der Art und Weise, wie Interaktivität realisiert wird, ist dann ohne Mehraufwand nicht mehr möglich. Sprich: Die Diskussion über die "Demo" sollte ausführlich erfolgen und alle Bestandteile umfassen.

Rechte

Die Verwertungsrechte (Verkauf, Veränderungen, Weiterverwendung) an der Animation liegen nach Beendigung des Projektes auf Seiten des Auftraggebers. Dies betrifft auch alle Rohdaten.

Zeitrahmen

Auslieferungstermin für die "Demo" ist der 20.03.02, der Endtermin für die Gesamtproduktion ist der 12.04.02.

Zeitlicher Verzug

Es wird eine "Pufferphase" von zwei Wochen vereinbart: Innerhalb dieses Zeitraumes kann die Produktion finalisiert werden ohne eine Reduktion der in diesem Angebot vereinbarten Kosten. Der weitere Verzug der Produktion zieht einen Preisnachlass auf den Netto-Gesamtbetrag von xyz € pro Woche nach sich. Verzögerungen bei Absprachen und Freigaben, die sich durch zu lange Antwortzeiten des Auftraggebers ergeben, gehen zu Lasten des Auftraggebers und ziehen keinen Preisnachlass nach sich. Ein eventuell entstehender Mehraufwand (allgemeine Änderungen oder Erweiterungen, Kundenkorrekturen) wird von uns nach Aufwand berechnet. Dies geschieht jeweils vorab in Absprache mit Ihnen.

Zahlungsmodalitäten

Nach Übergabe der "Demo" sind 25% der Gesamtkosten fällig, nach insgesamt 4 Wochen Produktionszeit weitere 25% und nach Abschluss des Projektes die restlichen 50%.

Verbindlichkeit

Aufgrund der zu erbringenden Vorleistungen durch Grafik GmbH wird die Auftragsbestätigung durch den Auftraggeber für beide Seiten als verbindlich angesehen.

.....
Datum: Unterschrift Auftraggeber:

.....
Datum: Unterschrift Auftragnehmer:

7.4 Formular Projektauftrag (manuelle Bearbeitung)

Projektauftrag Verteiler		Projektnummer:						
		<input type="checkbox"/> Neuentwicklung <input type="checkbox"/> Komponentenentwicklung <input type="checkbox"/> Individuelle Anpassung einer Eigenentwicklung <input type="checkbox"/> Individuelle Anpassung von Standard-software/ -technik						
Benennung des Auftrages:		Projektkurzbezeichnung:						
Erläuterung des Auftrages (Anlass, Zweck):								
	Beginn	Ende	Dauer (Tage)	Aufwand (MM)	Personal-einsatz (wer / wie viele?)	Person-al-kosten in €	Sonsti-ge Kosten in €	Gesamt-kosten in €
AP, Phase								
AP, Phase								
AP, Phase								
AP, Phase								
AP, Phase								
...								
Summe								
Gesamtkosten: € davon Personalkosten: €								
Erläuterungen zum geplanten Vorgehen und den Planungsdaten:								
Projektleiter: Name:		Erweiterte Projektleitung / Planungsteam (falls schon bekannt): Name: Aufgabe: Name: Aufgabe: Name: Aufgabe:						
Projektteam (falls schon bekannt): Name: Aufgabe: Name: Aufgabe: Name: Aufgabe:								
Unterschrift (Projektleiter):					Unterschrift (Geschäftsführung):			

7.5 Formular Projektauftrag (computergestützt)

Projektauftrag anzeigen	
Projektname	Multimedia Animation "Prozessmodelle"
Projektbeschreibung	Es ist eine Multimedia Animation zum Thema "Prozessmodelle" zu erstellen. Zielplattform ist HTML mit Flash.
Teilprojekt	-
Hauptprojekt	-

Kundendaten			
Kundenname	pm consult		
Straße, Nr.	Nebenstraße 13		
Ort	Untendorf		
Postleitzahl	08150	Telefon	+12 345 678 - 9
E-Mail	pro@pm.ag	Faxnummer	+12 345 678 - 10
Ansprechpartner			
Anrede	Herr		
Vorname	Armin		
Name	Eisen		

Projektdaten			
Eingabedatum	10.02.2002	Projektleiter	Edgar Kraus
Projektbeginn	04.03.2002	Beginndatum	04.03.2002
Projektende	31.05.2002	Enddatum	31.05.2002

Projektbudget			
Projektaccount	Datum	Budget extern	Budget intern
018020053	01.02.2002	20000	50000

■ Abbildung 7.5.1

Quelle: [Buh02]

7.6 Use-Case Schablone

1. Übersicht

- 1.1. Name/Kurzbeschreibung [Ziel des Anwendungsfalls]
- 1.2. Actors [Personen, Organisationen oder andere Systeme, die an dem Anwendungsfall beteiligt sind]
- 1.3. Abhängige Use Cases [Anwendungsfälle, die Informationen erhalten oder Hilfsprozesse bereitstellen]
- 1.4. Auslösendes Ereignis [gibt es ein Ereignis, das den Anwendungsfall auslöst?]

2. Aufgabenbeschreibung

2.1. Hauptfluss

- 2.1.1. Erste Aktion
- 2.1.2. Zweite Aktion
- 2.1.3. ...
- 2.2. Sonderfälle oder Erweiterungen [alternative Ausführungen zum Anwendungsfall]
- 3. Vorbedingungen [Aktion wird nur ausgeführt, wenn diese Bedingung erfüllt ist]
- 4. Nachbedingungen [wichtige Ergebnisse der Anwendungsfälle bei Erfolg oder auch Misserfolg]
- 5. Spezielle Erfordernisse [z.B. nichtfunktionale Anforderungen wie bspw. Antwortzeiten]

7.7 Beispiel zur Projektstrukturplanung

Die Abbildung 7.7.1 zeigt ein Beispiel, bei dem Arbeitspakete unter Berücksichtigung des Prozesses und des Produkts bestimmt wurden.

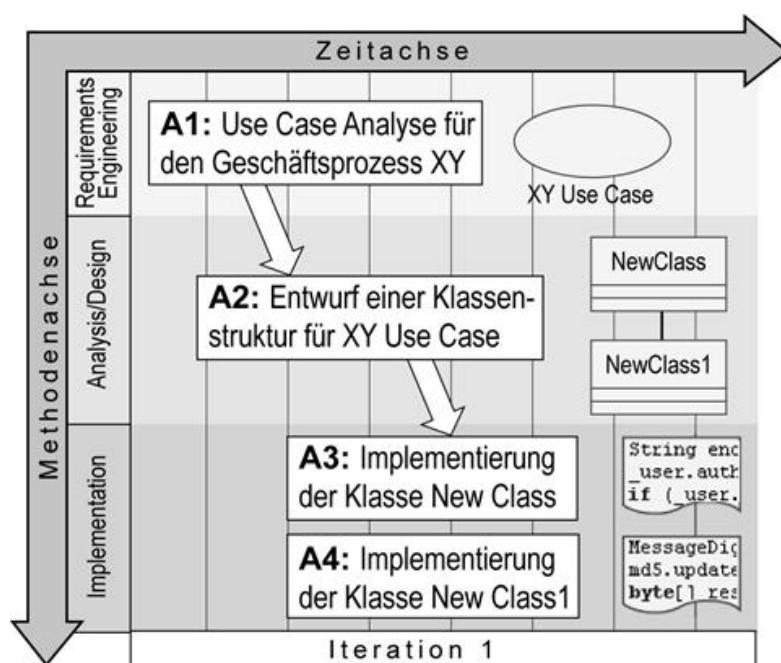


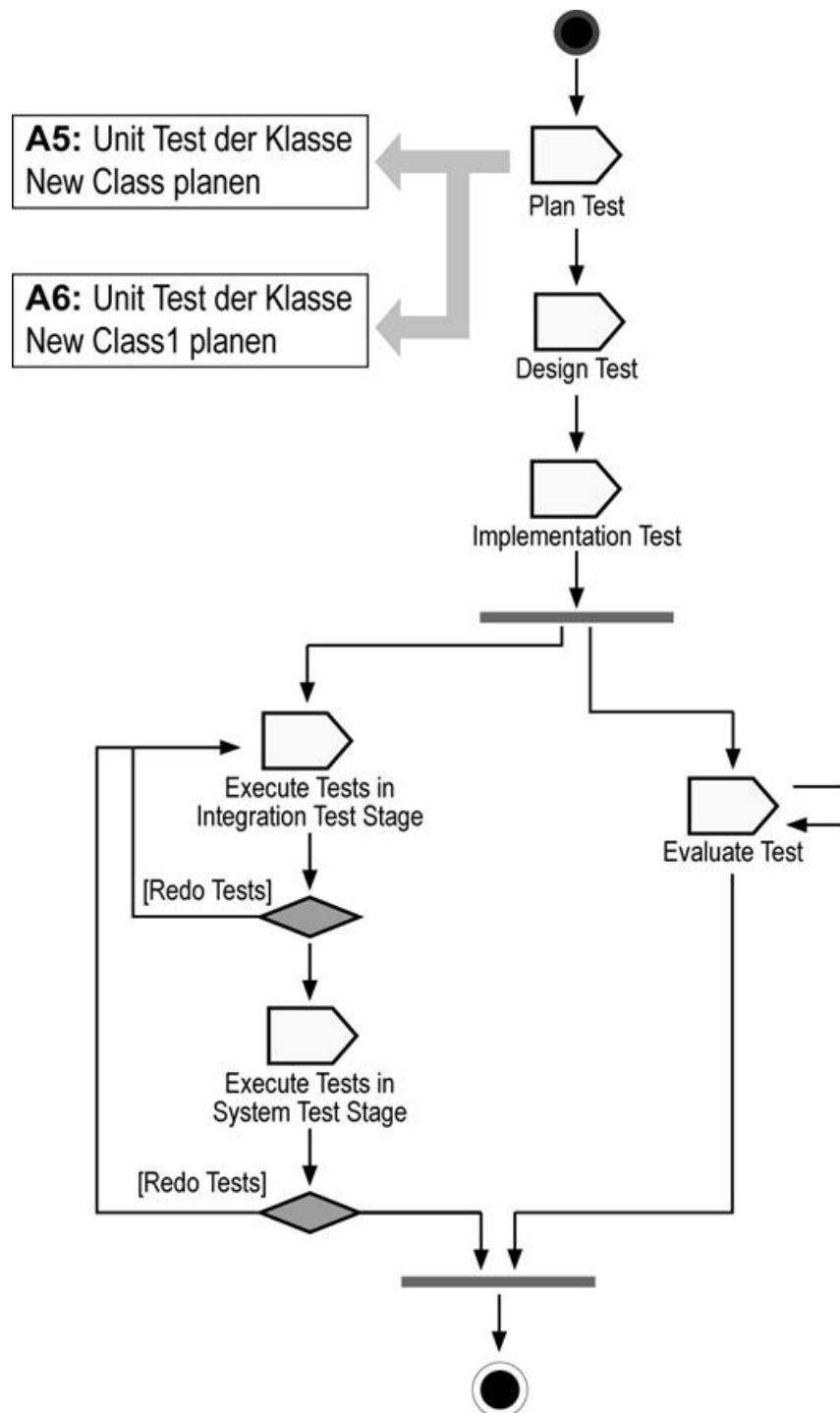
Abbildung 7.7.1: Arbeitspakete A1 bis A4 unter Berücksichtigung von Produkt und Prozess

Wie diese Abbildung zeigt, müssen zwei Arbeitspakete (A1 und A2) aus dem Prozess resultieren, der eine detaillierte Use Case Analysis und den Entwurf eines Klassendiagramms vorschreibt.

Das Klassendiagramm selbst kann zu einem späteren Zeitpunkt der Feinplanungen als Ausgangspunkt für zwei weitere Arbeitspakete (A3 und A4) verwendet werden, die eine Zerlegung der Programmieraufgabe in zwei Klassen dokumentiert (Abstraktionen). Auch wieder ein gutes Beispiel, dass man die Planungen dem Wissensstand an-

passen muss – man kann erst dann über die Programmierung von Klassen entscheiden, wenn man weiß, welche Klassen überhaupt erforderlich sind.

Aus der folgenden Abbildung wird der Zusammenhang aus Prozess und Arbeitspaketen noch deutlicher. Hier wurde ein Ausschnitt des Rational Unified Process ausgewählt. Es handelt sich um ein Aktivitätsdiagramm (Bestandteil der UML), das den Testablauf beschreibt. Die erste notwendige Aktivität ist das Planen der Tests (Plan Test). Da diese Aktivität vorgeschrieben ist, muss sie durch ein Arbeitspaket im Projektstrukturplan abgebildet werden. Daraus resultieren die Arbeitspakete A5 und A6.

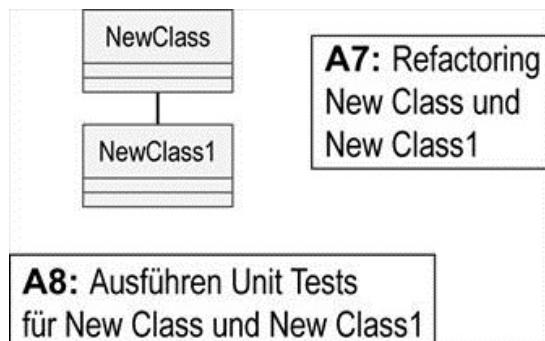


■ Abbildung 7.7.2: Arbeitspakete A5 und A6, die aus dem Prozess resultieren. Aus: Rational Unified Process 5.1.1, Firma Rational Co.

Berücksichtigen Sie auch Aufgaben, die nicht so auf der Hand liegen, die aber oft vernachlässigt werden und Projekte aus der Bahn werfen. Einer dieser Punkte ist die Überarbeitung von schon funktionierendem Quellcode, in der Literatur auch unter dem Begriff "**Refactoring**" oder **Refaktorisierung** bekannt [Fow00]. Folgende Ziele werden damit verfolgt:

- Vorbereiten des Codes für nachfolgende Erweiterungen,

- Einfügen von Variationspunkten,
- Aufräumen, wenn Code schlecht strukturiert,
- Kapselung (Get-, Set-Methoden für Eigenschaften),
- Beseitigung von Redundanz,
- Neue abstrakte Klassen einfügen,
- Beseitigung unnötiger Abhängigkeiten,
- Selbstdokumentierenden Code erzeugen,
- Aussagekräftige Namen,
- Verständliche Algorithmen (keine langen Methoden),
- Verbesserung der Rechengeschwindigkeit.



■ Abbildung 7.7.3: Arbeitspakete A7 und A8 in Folge von Refactoring

Weitere Aufgaben ergeben sich aus den folgenden Punkten:

- Hilfen für die Programmhandhabung,
- Guidelines (Installation, Abarbeitung von Prozessen),
- Referenzen (z.B. wenn auch ein Application Program Interface zum Produkt gehört),
- Schulungsmaterial für Präsenz oder virtuelle Tutorien,
- Agenten/Assistenten (z.B. "lebende" Büroklammern).

Bei der Bildung von Arbeitspaketen muss besonders darauf geachtet werden, dass jede Aufgabe so formuliert wird, dass ein prüfbares Ergebnis herauskommt. Ohne diese Bedingung wird das folgende Projektcontrolling sehr erschwert.

Arbeitspakete müssen beschrieben werden. Dazu wird in Abbildung 7.7.4 ein computergestütztes Formular vorschlagen.

Einzelauftrag anzeigen					
Bezeichnung	Flash-Szenen				
Beschreibung	Flash-Szenen erstellen und Navigation realisieren				
Status	in Bearbeitung				
Meilenstein	002 3D-Szenen Release 1				
Vorgänger	001 Storyboard				
Nachfolger	003 3D-Szenen Release 2				

Zeitplanung					
Planung vom	Anfangsdatum	Enddatum	Sollstunden		
02.03.2002	01.04.2002	19.04.2002	120		

Mitarbeiterplanung					
Qualifikation	Std-Soll	Std-Satz	Mitarbeitername*	Einsatzbeginn*	Einsatzende*
Flash	120	60	Claudia	01.04.2002	19.04.2002

Ressourcenplanung					
Typ / Art	Std-Soll	Std-Satz	Bezeichnung*	Einsatzbeginn*	Einsatzende*
Workstation	120	20	3D Workstation	01.04.2002	19.04.2002

■ Abbildung 7.7.4: Computergestütztes Formular für einen Einzelauftrag
Quelle: [Buh02]

7.8 Planungshinweise für das XP-Prozessmodell

Im Folgenden wird ein Projektplan vorgestellt, der nach dem eXtreme Programming entwickelt wurde. Bei dem Projekt selber handelt es sich um die Entwicklung einer Komponente für die Online-Recherche einer Bibliothek. Die hier vorgestellte Detailplanung bezieht sich auf das Release 1, das inhaltlich eine Recherche im eigenen Bestand ermöglichen soll. Das Release 2 soll dann auch eine Recherche in fremden Beständen ermöglichen. Die Gesamtdauer des Projekts beträgt 80 Tage. Für das Release 1 wurden 35 Tage veranschlagt. Das Team besteht aus 5 Mitarbeitern – zwei Arbeitsgruppen mit jeweils zwei Mitarbeitern und einem Projektleiter, der auch inhaltlich in den Gruppen aktiv wird und sich um das Projekt- und Konfigurationsmanagement kümmert. Für das Release 1 wurden drei Iterationen geplant – die Iterationsplanung selber wurde nach Abarbeitung der Vorgänge 5, 6, 7, 8 überarbeitet. Der abgebildete Plan liegt also in der Version 2 vom 15.2.2002 vor.

	Vorgangsnr.	Vorgangsnr.	Dauer	Anfang	Ende	Vorgänger
1	Online Recherche Bibliothek - eXtreme Programming		80 Tage	Fr 01.02.02	Mo 03.06.02	
2	Konfigurationsverwaltung		80 Tage	Fr 01.02.02	Mo 03.06.02	
3	Projektmanagement		80 Tage	Fr 01.02.02	Mo 03.06.02	
4	Release1 - Recherche im eigene Bestand		35 Tage	Fr 01.02.02	Fr 22.03.02	
5	Kunde schreibt Stories/Entwickler schätzen Zeit für das erste Release		5 Tage	Fr 01.02.02	Do 07.02.02	
6	Auswahl der Kern - "Stories" nach Risiko- und Wirtschaftlichkeitsbetrachtung		2 Tage	Fr 08.02.02	Mo 11.02.02	5
7	Festlegen des Systemmetaphors und des Termins für die erste Iteration		2 Tage	Di 12.02.02	Do 14.02.02	6
8	Festlegen der Stories für die erste Iteration (alle Stories müssen eine lauffähige Version ergeben)		1 Tag	Fr 15.02.02	Fr 15.02.02	7
9	Iteration I1		10 Tage	Mo 18.02.02	Fr 01.03.02	
10	Arbeitsgruppe 1 - Client (Formular), Servlet und Request Handler mit Dummy Ausgabe		10 Tage	Mo 18.02.02	Fr 01.03.02	
11	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
12	Partner wählen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
13	Funktionstests für die Story schreiben (Kunden einbeziehen)		10 Tage	Mo 18.02.02	Fr 01.03.02	8
14	Unit-Tests schreiben, Module implementieren, Tests ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
15	Refaktorieren, Tests ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
16	Funktionstest für Story ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
17	Story fertig		0 Tage	Fr 01.03.02	Fr 01.03.02	16
18	Arbeitsgruppe 2 - Datenbank analysieren, Datenbankservice implementieren (SQL mittels API)		10 Tage	Mo 18.02.02	Fr 01.03.02	
19	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
20	Partner wählen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
21	Funktionstests für die Story schreiben (Kunden einbeziehen)		10 Tage	Mo 18.02.02	Fr 01.03.02	8
22	Unit-Tests schreiben, Module implementieren, Tests ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
23	Refaktorieren, Tests ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
24	Funktionstest für Story ausführen		10 Tage	Mo 18.02.02	Fr 01.03.02	8
25	Story fertig		0 Tage	Fr 01.03.02	Fr 01.03.02	24
26	Iteration I2		10 Tage	Mo 04.03.02	Fr 15.03.02	
27	Arbeitsgruppe 1 - SQL Befehle entwickeln, Request Handler abschließen		10 Tage	Mo 04.03.02	Fr 15.03.02	
28	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
29	Partner wählen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
30	Funktionstests für die Story schreiben (Kunden einbeziehen)		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
31	Unit-Tests schreiben, Module implementieren, Tests ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
32	Refaktorieren, Tests ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
33	Funktionstest für Story ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
34	Story fertig		0 Tage	Fr 15.03.02	Fr 15.03.02	33

Abbildung 7.8.1: Projektplan für das eXtreme Programming (1)

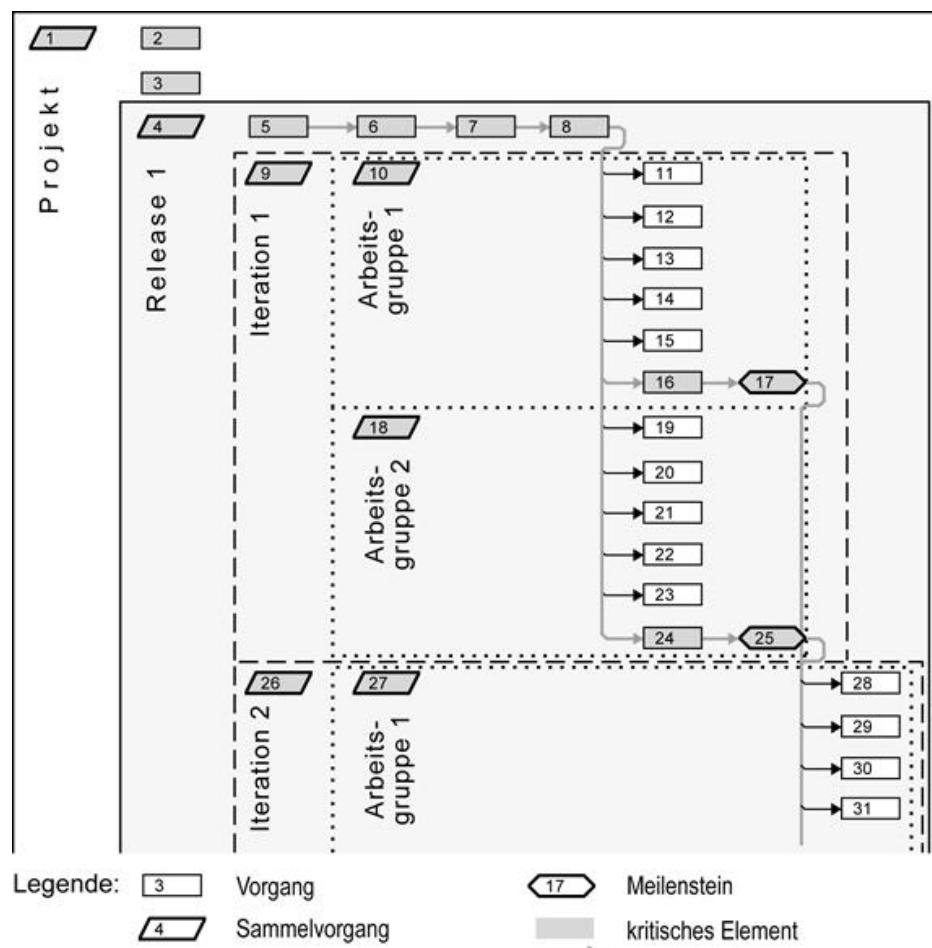
	Iteration I2 (Fortsetzung)		10 Tage	Mo 04.03.02	Fr 15.03.02	
35	Arbeitsgruppe 2 - Datensicherheit und Error Handler entwickeln		10 Tage	Mo 04.03.02	Fr 15.03.02	
36	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
37	Partner wählen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
38	Funktionstests für die Story schreiben (Kunden einbeziehen)		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
39	Unit-Tests schreiben, Module implementieren, Tests ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
40	Refaktorieren, Tests ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
41	Funktionstest für Story ausführen		10 Tage	Mo 04.03.02	Fr 15.03.02	17,25
42	Story fertig		0 Tage	Fr 15.03.02	Fr 15.03.02	41
43	Iteration I3		5 Tage	Mo 18.03.02	Fr 22.03.02	
44	Arbeitsgruppe 1 - Layout und Hilfefunktion		5 Tage	Mo 18.03.02	Fr 22.03.02	
45	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
46	Partner wählen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
47	Funktionstests für die Story schreiben (Kunden einbeziehen)		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
48	Unit-Tests schreiben, Module implementieren, Tests ausführen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
49	Refaktorieren, Tests ausführen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
50	Funktionstest für Story ausführen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
51	Story fertig		0 Tage	Fr 22.03.02	Fr 22.03.02	50
52	Arbeitsgruppe 2 - Integrieren, Installieren, Gesamttest Release 1		5 Tage	Mo 18.03.02	Fr 22.03.02	
53	Story in Task unterteilen, Aufwand schätzen Taskcards anlegen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
54	Partner wählen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
55	Funktionstests für das Release schreiben (Kunden einbeziehen)		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
56	Funktionstest für Release ausführen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
57	Refaktorieren, Funktionstests ausführen		5 Tage	Mo 18.03.02	Fr 22.03.02	34;42
58	Story fertig		0 Tage	Fr 22.03.02	Fr 22.03.02	57
59	Release fertig		0 Tage	Fr 22.03.02	Fr 22.03.02	58;51
60	Release2		45 Tage	Mo 25.03.02	Mo 03.06.02	59

Abbildung 7.8.2: Projektplan für das eXtreme Programming (2)

Was an dem Projektplan lt. Abbildung 7.8.1, Abbildung 7.8.2 und Abbildung 7.8.3 besonders auffällt, ist die Tatsache, dass die Vorgänge sich in den Iterationen stetig wiederholen. Die extreme Anwendung eines begrenzten Umfangs an Methoden ist ja ein besonderes Kennzeichen dieses Prozessmodells. Sie sehen aber auch, dass die Vor-

gänge selber in diesen Iterationen nicht miteinander verknüpft sind. Selbst das "Wählen der Partner" für das paarweise Programmieren ist über die gesamte Zeitspanne möglich. Es kann also der jeweils passende Partner für eine gerade anstehende Aufgabe gewählt werden.

In dem Netzplanauszug aus Abbildung 7.8.3 ist zu sehen, dass die Vorgänge innerhalb einer Iteration alle parallel angeordnet sind.



■ Abbildung 7.8.3: Netzplanauszug mit Vorgangsnummern aus dem Projektplan lt. Abbildung 7.8.1 und Abbildung 7.8.2

7.9 Beispiel zur Netzplantechnik

Problemstellung:

Der Ablauf eines Fertigungsprozesses soll optimiert werden. Dazu benötigt man folgende Informationen:

- die üblichen Vorgänge zur Bewältigung des Projekts,
- die gegenseitigen Abhängigkeiten dieser Vorgänge untereinander,
- die Festlegung wichtiger Termine bei der Projektrealisierung,

- den Bedarf an finanziellen und materiellen Ressourcen.

Lösung:

Mit Hilfe einer **Strukturanalyse** werden zunächst alle zur Durchführung des Projekts erforderlichen Arbeitsgänge, Tätigkeiten oder Aktivitäten (so genannte Vorgänge) erfasst und in ihrer zeitlichen und sachlogischen Abfolge näher bestimmt. Das Ergebnis dieser Überlegungen wird tabellarisch als so genannte **Vorgangsliste** festgehalten.

Im folgenden Beispiel will eine kleine Firma ein Netzwerk aufbauen, bei dem alle Mitarbeiter Zugriff auf einen zentralen Server bekommen.

Vorgangs-Nr.	Bezeichnung	Dauer	Vorgänger	Nachfolger
1	Start	0		2, 3, 7
2	Recherche Netzwerk-Anbieter (Software)	3	1	4
3	Recherche Anbieter Server (Hardware)	3	1	4
4	Entscheidung für eine HW-SW-Kombination	1	2, 3	5
5	Lieferzeit der HW/SW	5	4	6, 11
6	Server aufstellen incl. Installation der Software	3	5	12
7	Rechner der Mitarbeiter auf Netzwerk-Fähigkeit prüfen	1	1	8
8	fehlende Hardware-Komponenten nach-bestellen	1	7	9
9	Lieferzeit der Hardware-Komponenten	6	8	10
10	bestellte Komponenten installieren	2	9	11
11	Client-Software installieren	3	5, 10	12
12	Rechner miteinander verbinden	1	6, 11	13
13	Ziel	0	13	

|  Tabelle 7.9.1: Vorgangsliste

Im Anschluss an die Strukturanalyse wird ein **Netzplan** aufgestellt. Bausteine eines Netzplans sind die Vorgangsknoten, die wichtige Informationen zum jeweiligen Vorgang enthalten, und Richtungspfeile, die die Anordnungsbeziehungen zwischen den Vorgängen kennzeichnen.

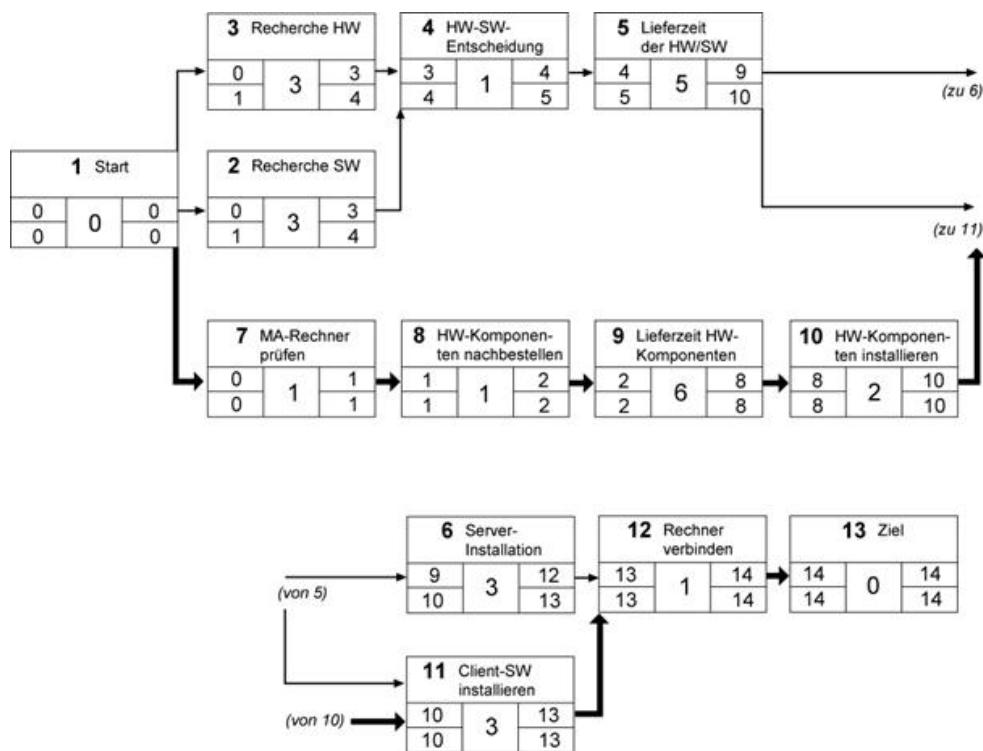


Abbildung 7.9.2: Netzplan in graphischer Darstellung; der kritische Pfad ist mit fetter Linie dargestellt

7.10 Beispiele für Checklisten

Jedes Projekt ist anders und Sie werden Ihre eigenen Checklisten erarbeiten müssen – aber vielleicht ist es ganz hilfreich, einige Anregungen zu haben.

Vollständigkeit der Programmdokumentation

- Sind alle geforderten Funktionen vollständig beschrieben?
- Sind alle Schnittstellen zur Hardware beschrieben?
- Sind alle Schnittstellen zu anderen Programmteilen beschrieben?
- Gibt es für die Dokumentation ein Inhaltsverzeichnis?
- Gibt es eine Zusammenfassung aller System- und Fehlermeldungen?
- Gibt es eine Zusammenfassung aller Hilfetexte?
- Enthält die Dokumentation alle für die Anwendung notwendigen Informationen?

Verständlichkeit und Übertragbarkeit des Quellcodes

- Sind die Autoren und die Änderungen mit Namen und Datum vollständig kommentiert?
- Gibt es eine Kurzbeschreibung zu der Klasse?
- Gibt es eine Beschreibung der verwendeten Klassen?
- Gibt es eine Beschreibung, wie diese Klasse zu verwenden ist (Beschreibung der Schnittstellen)?
- Ist die Bezeichnung der Klassen, Methoden, Variablen einheitlich?
- Werden Homonyme vermieden?
- Werden Synonyme vermieden?
- Ist die Klasse überschaubar und sauber gegliedert?
- Werden 50 Zeilen Code in einer Methode nicht überschritten?
- Ist die Verschachtelungstiefe von Schleifenanweisungen und Verzweigungen höchstens drei?
- Gibt es höchstens drei Unterklassen?
- Werden Sprunganweisungen vermieden?

Vollständigkeit des Quellcodes

- Sind alle in den Anwendungsfällen angegebenen Funktionen ausführbar?
- Sind alle notwendigen Daten für die Funktionen vorhanden?
- Sind die Fehlermeldungen vollständig?
- Sind die Hilfetexte ausreichend?
- Gibt es ausreichende Rückmeldungen?
- Ist der Quelltext vollständig kommentiert?

Vollständigkeit des Programmtests

- Liegt ein Testplan vor?
- Gibt es im Testplan eindeutige Namen und Bezeichnungen für den Testverantwortlichen, das Projekt, die Aufgabe und die Komponente?
- Gibt es im Testplan eine genaue Funktionsbeschreibung der zu testenden Komponente?
- Gibt es im Testplan eine genaue Beschreibung der zu erbringenden Nachweise?
- Gibt es eine Beschreibung für die Strategie des Testfallentwurfs?
- Liegen alle Testfälle mit den zu erwartenden Ergebniswerten vor?
- Gibt es Angaben zu den erforderlichen Formaten für die Eingabedaten?
- Gibt es eine Beschreibung der Testumgebung?
- Sind alle Komponenten, die in den Test einbezogen werden, beschrieben?

- Gibt es eine Beschreibung der Verzeichnisstruktur für die Ablage von Test- und Ergebniswerten?
- Sind die Hardwareanforderungen beschrieben?
- Gibt es Hinweise für die Abarbeitung der Tests?
- Liegt ein Testprotokoll vor?
- Enthält das Testprotokoll genaue Bezeichnungen zum Testverantwortlichen, zu Projekt, Arbeitsaufgabe, Komponente, Datum und Testplan?
- Liegen die Schlussfolgerungen aus dem Test vor?

Mitarbeitermotivation

- Kennen Sie das Ziel und den Nutzen des Gesamtprojekts?
- Sind Sie vom Nutzen des Projekts überzeugt?
- Werden Sie in die Diskussion zu den Projektzielen einbezogen?
- Halten Sie die technische Lösung für richtig?
- Ist die personelle und materielle Ausstattung ausreichend?
- Werden Sie entsprechend Ihrer Qualifizierung und Ihrer Erfahrungen eingesetzt?
- Fühlen Sie sich stark überfordert?
- Fühlen Sie sich stark unterfordert?
- Werden Sie ausreichend entlohnt?
- Werden Ihre Leistungen ausreichend öffentlich gewürdigt?
- Besteht die Möglichkeit, durch gute Arbeit eine verantwortungsvollere Position zu erringen?

Kommunikation zwischen Projektmitgliedern

- Gibt es die richtige Anzahl von Arbeitsberatungen?
- Wird vor den Arbeitsberatungen eine Tagesordnung verschickt?
- Gibt es in den Arbeitsberatungen eine offene und gleichberechtigte Diskussion?
- Gibt es ausreichende Kommunikationswege im Projekt?
- Können Sie problemlos und ohne Zeitverzug mit dem Projektleiter sprechen?
- Können Sie Arbeitsaufgaben auch mit anderen Kollegen besprechen?
- Werden Ihnen von anderen Kollegen wichtige Informationen vorenthalten, um sich im Projekt Machtpositionen zu sichern?

Kommunikation mit dem Auftraggeber

- Hatten Sie persönlichen Kontakt zum Auftraggeber?
- Gibt es eine feste Kontaktperson beim Auftraggeber oder wechselt die Kontakt-person ständig?

- Ist der Auftraggeber ausreichend über das Projekt informiert?
- Identifiziert sich der Auftraggeber mit dem Projekt und hat er positive Erwartungen?
- Arbeitet der Auftraggeber an der Anforderungsspezifikation mit?
- Nimmt der Auftraggeber regelmäßig an Arbeitsberatungen teil?

Qualität der Terminplanung

- Konnten Sie ihre bisherigen Arbeitsaufgaben termingerecht erledigen?
- Haben Sie die geplanten Termine überzogen?
- Waren Sie früher als geplant fertig?
- Wurden Sie bei der Festsetzung der Termine angehört?
- Kennen Sie das Verfahren zur Festsetzung der Termine?
- Wurden die Termschätzungen nach der Auswertung abgelaufener Projektphasen überarbeitet?
- Gab es genügend Erfahrung, um eine fachgerechte Terminplanung vornehmen zu können?
- Gibt es ein schnelles Entscheidungsverfahren bei drohenden Terminverschiebungen?

Qualität der Personalplanung

- Ist Ihr Team ausreichend qualifiziert für die Arbeitsaufgabe?
- Gibt es im Gesamtprojekt ausreichend qualifiziertes Personal?
- Gab es ein Gespräch zum Anforderungsprofil der anstehenden Arbeitsaufgaben?
- Wurden die Arbeitsaufgaben klar abgegrenzt und eindeutig definiert?
- Wurden notwendige Lernphasen berücksichtigt?
- Wurde anstehender Urlaub berücksichtigt?
- Wird unentschuldigtes Fehlen ausreichend verfolgt?
- Gibt es eine angemessene Arbeitszeitregelung, z.B. flexible Arbeitszeiten?

8 Zusatzmaterialien

Als Zusatzmaterialien werden ein Lernspiel und zahlreiche Videos zu unterschiedlichen Problemstellungen des Projektmanagements zur Verfügung gestellt.

8.1 Lernspiel

In diesem Teil des Lernmoduls werden Sie Ihre erworbenen Kenntnisse über die Führung von Mitarbeitern innerhalb von Softwareprojekten testen und dieses in mehreren Beispielen anwenden können. Wie schon beim Planungsspiel werden Sie auch hier die Rolle des Projektleiters übernehmen und sich mit dem Ablauf eines ersten Projektmeetings beschäftigen. Dabei werden Sie durch Videobeispiele unterstützt und durch ihre Aufgaben geleitet.

Ein Lernspiel sollte in der vorgegebenen Reihenfolge durchlaufen werden. Es empfiehlt sich keines der Kapitel zu überspringen und auszulassen.

Handlungsrahmen:

Ihre Firma hat einen Auftrag durch die Reederei Container Express erhalten. Die Voraussetzungen sind denkbar schlecht. Die Anforderungen wurden nicht eindeutig gestellt und die eigentliche Aufgabe muss erst erarbeitet werden.

Das vierköpfige Team des Projektes "Container Express" trifft sich nun in einem ersten Arbeitsmeeting mit dem Projektleiter. Um folgenden Aufgaben möglichst gut lösen zu können, sollten Sie die entscheidenden Kapitel im Lehrbuch noch einmal durcharbeiten. Dabei empfehlen sich vor allem die Kapitel über Teamführung, Projektplanung und Projektstart.

8.1.1 Das erste Meeting

Schauen Sie sich nun das komplette Video des ersten Meetings an. Versuchen Sie dabei vor allem die Ist-Situation zu erfassen.

Wie ist die Ausgangssituation der Besprechung?

Wie steht es um die Motivation des Teams?

Sie können sich Notizen machen, um Details festzuhalten, die Ihnen hierbei besonders aufgefallen sind.



An dieser Stelle befindet sich online ein Video.

[https://vfhpmp.eduloop.de/loop/Das erste Meeting](https://vfhpmp.eduloop.de/loop/Das%20erste%20Meeting)

Schauen Sie das Video danach ein zweites Mal und notieren Sie sich alle Fehler, die der Projektleiter Ihrer Meinung nach innerhalb des Meetings gemacht hat. Überlegen Sie sich hierbei auch, wie man es besser machen könnte und diskutieren Sie die gefundenen Fehler anschließend in kleinen Gruppen.

8.1.2 Analyse des Meetings

Sicherlich ist Ihnen im Video eine ganze Reihe von Fehlern aufgefallen, die die Projektleiterin begangen hat. Der folgende Abschnitt fasst diese Fehler noch einmal zusammen und diskutiert sie anhand von kleinen Videosequenzen. Sind Sie auf alle Fehler aufmerksam geworden?

- Die fehlende Vorstellung

 An dieser Stelle befindet sich online ein Video.

https://vfhpmpm.eduloop.de/loop/Analyse_des_Meetings

Im Projekt Container Express wird ein neues Team eingesetzt. Die einzelnen Mitarbeiter kennen sich zum Teil noch nicht. Obwohl es sich um kein eingespieltes Team handelt, ist die Stimmung innerhalb des Teams sehr gut. Dies ändert sich jedoch mit dem Erscheinen der Projektleiterin merkbar.

Die Projektleiterin macht bei Ihrer Ankunft einen aufgewühlten Eindruck, sie wirkt unvorbereitet und nervös. Offensichtlich verfügt sie über keine große Erfahrung. Dies äußert sich vor allem im Kramen in den Unterlagen noch bevor das Meeting begonnen hat. Die Vermutung liegt nahe, dass sie sich über das besagte Suchen in den Unterlagen zu konzentrieren und zu beruhigen versucht.

Die Projektleiterin versucht, ihre Nervosität über Formalität und Förmlichkeit zu überspielen. Diese Stimmung schlägt sofort auf das Team über. Das Team zieht sich merkbar zurück und wird automatisch auch förmlich.

An dieser Stelle hätte die Projektleiterin mit einer Vorstellungsrunde die Anspannung aus dem Team nehmen können und auch so ihrer eigenen Unsicherheit entgegen kämpfen können. Dies hat sie allerdings versäumt. Die Begrüßung erfolgt lediglich durch ein kurzes "Hallo!".

- Fehlende Unterlagen

 An dieser Stelle befindet sich online ein Video.

https://vfhpmpm.eduloop.de/loop/Analyse_des_Meetings

Da es sich im Projekt "Container Express" um ein neues Projekt handelt, sollte an dieser Stelle eigentlich noch einmal eine Einführung in die Thematik durch die Projektleiterin erfolgen. In unserem Beispiel ist dies jedoch nicht der Fall. Außerdem wird in der Videosequenz noch einmal deutlich, dass es die Projektleiterin nicht nur versäumt hat, wichtige Informationen über die Rederei weiterzuleiten, vielmehr weiß sie nicht, welche Unterlagen an das Team weitergereicht wurden und nun vorliegen.

Durch die mangelnde Informationszufuhr haben die Mitarbeiter keine ausreichenden Informationen über Container Express und das anstehende Projekt, um sich ein eigenes Bild zu machen.

- Fehlende Vorbereitung



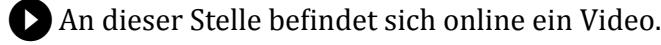
An dieser Stelle befindet sich online ein Video.

https://vfhpmpm.eduloop.de/loop/Analyse_des_Meetings

Nachdem die Projektleiterin durch das Team darauf aufmerksam gemacht wurde, dass wichtige Unterlagen fehlen, versucht sie nun eine Einführung in das Projekt zu geben. Allerdings gelingt ihr das nur teilweise. Sie muss die Informationen in ihren Unterlagen suchen und ist gezwungen diese vorzulesen.

Dies weist auf eine mangelnde Vorbereitung seitens der Projektleiterin hin. Solch ein Verhalten wirkt auf das Team unprofessionell und lässt an der Führungsqualität der Projektleiterin zweifeln.

- Der passive Projektleiter



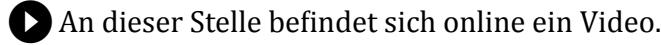
An dieser Stelle befindet sich online ein Video.

https://vfhpmpm.eduloop.de/loop/Analyse_des_Meetings

In der Videosequenz wird deutlich, wie sehr die Diskussion der Projektleiterin Stück für Stück entgleitet. Sie nimmt sich immer mehr zurück, was sowohl an ihrer schlechten Vorbereitung als auch an ihrer Nervosität liegt. Die Mitarbeiter führen die Diskussion praktisch ohne sie – die Projektleiterin wird zum stillen Beobachter. Sie hat kein eigenes Ziel.

Durch die fehlende Einführung hat die Diskussion keine Richtung, was sich nun nachteilig auf das gesamte Meeting auswirkt.

- Der Abbruch



An dieser Stelle befindet sich online ein Video.

https://vfhpmpm.eduloop.de/loop/Analyse_des_Meetings

Im Verlauf des Meetings wird schnell klar, dass dem Team wichtige Informationen fehlen. Die Projektleiterin muss sich zwischendurch erkundigen gehen. Ihr Verhalten wirkt abermals unprofessionell – die Informationen hätten bereits vor der Besprechung eingeholt werden müssen. So unterbricht sie nun die Projektsitzung ohne Grund und entzieht sich der Situation, als sie merkt, dass ihr das Projekt entgleitet.

- Die fehlende Zusammenfassung



https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Die Projektleiterin hätte nach Rückkehr aus der Pause den Zwischenstand der Besprechung noch einmal zusammenfassen sollen. Sie informiert die Mitarbeiter allerdings nur darüber, dass noch keine weiteren Informationen vorliegen und gibt die Aufgabe der Zusammenfassung der Zwischenergebnisse an einen Mitarbeiter weiter.

- Einwände abblocken



https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Die Projektleiterin lässt es nicht zu, dass sich eine Diskussion entwickelt. Sie geht nicht auf Einwände und Missverständnisse ein. Vorschläge, die in der Brainstormingrunde gemacht werden, werden von ihr nicht aufgegriffen und werden undiskutiert als Aufgaben umgesetzt.

Die Projektleiterin erweckt damit den Eindruck, dass sie das Meeting so schnell wie möglich beenden möchte. Ihr geht es nicht darum, sinnvolle Aufgaben zu finden, sondern die Mitarbeiter möglichst schnell zu beschäftigen.

- Die Namensfrage



https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Je länger die Sitzung dauert, desto mehr wirken sich bereits zu Beginn gemachte Fehler der Projektleiterin auf den Verlauf des Meetings aus. Ganz entscheidend hierbei ist die Tatsache, dass die Projektleiterin die Namen der Mitarbeiter nicht kennt.

Die fehlende Begrüßungsrounde führt nicht nur zu Problemen mit den Namen, sondern auch dazu, dass die einzelnen Stärken und Schwächen der Mitarbeiter nicht bekannt sind. So kann die Projektleiterin die Aufgaben nur willkürlich und nicht nach den Fähigkeiten und Erfahrungen ihrer Mitarbeiter verteilen.

- Nachfragen? Nein, danke!

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Während des Meetings werden die Aufgaben durch die Mitarbeiter kritisch hinterfragt. Dies resultiert vor allem aus den unklar gestellten Aufgabestellungen der Projektleiterin und zum anderen aus der willkürlichen Aufgabenverteilung. Die gemachten Vorschläge der Mitarbeiter werden allerdings durch die Projektleiterin nicht akzeptiert.

- Die lieben Nerven

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Gegen Ende des Meetings wird immer mehr deutlich, dass die Projektleiterin sich eigentlich schon aus der Diskussion verabschiedet hat, nachdem sie die Aufgaben vergeben hat. Sie wirkt zunehmend genervt und ungeduldig und distanziert sich von den Mitarbeitern immer mehr. Besonders verdeutlicht wird dies durch ihre Körperhaltung. Sie baut eine Mauer zwischen sich und der Thematik auf.

- Die Flucht

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Analyse_des_Meetings

Das Verhalten der Projektleiterin findet seinen Höhepunkt in der Flucht aus dem Meeting. Sie beendet die Diskussion abrupt, hinterlässt verwirzte Mitarbeiter und flieht rechts gelacht vor ihren Aufgaben.

8.1.3 Weitere Szenarien eines Meetings

Im letzten Abschnitt des Lernspiels haben Sie die Fehler des Projektleiters analysiert und sich dabei sicherlich schon Gedanken darüber gemacht, wie solche Fehler vermieden werden könnten und wie man es besser machen kann.

Nun ist es an Ihnen, in die Rolle des Projektleiters zu schlüpfen und ihr Team gut durch verschiedene Szenarien eines Meetings zu führen. Finden Sie sich in einem kleinen Team zusammen und versuchen Sie die folgenden Szenarien möglichst realistisch umzusetzen:

Eröffnung eines Meetings und Begrüßung der Mitarbeiter

Schauen Sie sich bevor Sie die Situation selbst umsetzen ruhig noch einmal die Begrüßungsszene im Meeting Container Express an.



An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Weitere_Szenarien_eines_Meetings

Versuchen Sie nun, ein fiktives Meeting einzuleiten. Überlegen Sie sich im Vorfeld den Namen des Projektes und wie Sie sich Ihrem Team am besten vorstellen können.

Lassen Sie das Meeting auf Video festhalten oder suchen Sie sich ein Publikum (andere Gruppen), denen Sie ihre Begrüßungsszene vorspielen. So können Sie diese danach in einer größeren Gruppe auswerten und so sehen, ob sie die Situation besser gemeistert haben als die Projektleiterin im Meeting Container Express. Analysieren Sie dabei Ihre eigenen Fehler, aber versuchen Sie auch die Dinge zu diskutieren, die besonders gut gelaufen sind.

Einführung in ein Thema

Die Projektleiterin im Projekt Container Express hat die Einführung in die Thematik des Auftrages verpasst. Schauen Sie sich die Szene noch einmal an.



An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Weitere_Szenarien_eines_Meetings

Im Verlauf Ihres Studiums haben Sie bereits an verschiedenen Projekten mitgearbeitet. Versuchen Sie sich an eines dieser Projekte zu erinnern und notieren Sie sich die Details, die Ihnen zu diesem Projekt einfallen.

Versetzen Sie sich anschließend in folgende Situation: Sie sind abermals der Projektleiter und haben Ihre Mitarbeiter bereits zu ihrem ersten Meeting begrüßt. Ihre Mitarbeiter haben zwar Informationen über den Auftrag erhalten, es ist aber an Ihnen, das Projekt den Mitarbeitern noch einmal vorzustellen. Geben Sie eine Einführung in Ihr Projekt, erklären Sie den Mitarbeitern worum es sich handelt, wer Ihr Auftraggeber ist und welche Ziele Ihr Meeting verfolgt. Versuchen Sie Ihre Mitarbeiter, so gut wie möglich über den aktuellen Stand zu informieren und so eine gute Grundlage für Ihr Meeting zu schaffen.

Lassen Sie sich auch hierbei wieder filmen und werten Sie die Szene in einer größeren Gruppe aus.

Führen einer Diskussion

Die Diskussion während des Meetings im Projekt Container Express ist denkbar schlecht verlaufen und hat zu einem großen Teil gar nicht stattgefunden. Das Videobeispiel verdeutlicht dies noch einmal.



An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Weitere_Szenarien_eines_Meetings

Eine Aufgabe eines Projektleiters ist es, sein Team gut durch Diskussion innerhalb des Projektes zu führen. Finden Sie sich erneut in Ihrem Team zusammen. Greifen Sie das Projekt wieder auf, zu dem Sie auch Ihre Einführung gemacht haben und versuchen Sie nun, in einer Diskussion mit Ihrem Team über den weiteren Verlauf des Projekts zu entscheiden. Seien Sie sich dabei Ihrer Rolle als Projektleiter bewusst und lassen Sie sich diese nicht aus der Hand nehmen. Vergeben Sie die Aufgaben innerhalb des Projektes gerecht und nach den Fähigkeiten der Mitarbeiter.

Folgende Hinweise sollen Ihnen dabei helfen, das Meeting mit einem positiven Ergebnis zu führen:

- Überlegen Sie sich schon im Vorfeld Fragestellungen, mit denen Sie Ihre Mitarbeiter in das Thema führen können.
- Notieren Sie sich Diskussionshilfen, damit Sie den roten Faden niemals verlieren.
- Versuchen Sie für sich, die einzelnen Charaktere Ihrer Mitarbeiter zu analysieren.
- Versuchen Sie auch stille Mitarbeiter in die Diskussion einzubeziehen - sie haben oft gute Vorschläge, trauen sich nur nicht, diese zu unterbreiten.
- Bleiben Sie ruhig.
- Fallen Sie niemandem ins Wort, aber unterbrechen Sie die Diskussion, wenn sie in eine falsche Richtung läuft.

Besprechen Sie danach mit Ihrem Team, welchen Eindruck Sie als Projektleiter hinterlassen haben, ob offene Fragen geblieben sind oder ob jeder im Team genau weiß, was er zu tun hat und wo seine Rolle innerhalb des Projektes liegt.

8.2 Videos

In diesem Lernstrang werden ausgewählte Themen des Lehrbuches als Videos aufbereitet - der Lehrstoff wird hier also durch reale Personen vermittelt. Dabei gehen die einzelnen Videos über das klassische Konzept einer Vorlesung im Hörsaal hinaus: Theoretische Erläuterungen werden durch Spielszenen ergänzt. Es werden auch Projektleiter aus Softwarefirmen in Mecklenburg-Vorpommern nach ihren Erfahrungen beim Managen von Projekten befragt.

 An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Videos>

Starten Sie hier das Begrüßungsvideo. Die Videos zu den einzelnen Themen werden auf den folgenden Seiten gestartet.

8.2.1 Video - Interview mit einem Projektleiter

Um Ihnen auch einen Einblick in die praktische Arbeit eines Projektleiters zu ermöglichen, haben wir mit Herrn Dr. Czubayko von der SIV AG (www.siv.de), einem Softwareunternehmen aus Mecklenburg-Vorpommern, über seine Arbeit gesprochen.

Sie können sich dieses Interview vollständig (Komplettvideo) oder thematisch aufbereitet (Folgevideos) ansehen.



Das komplette Interview mit Dr. Czubayko, Projektleiter in der SIV AG

Sehen Sie hier das komplette Interview. Fast alle im Lehrbuch besprochenen Themen werden aus Sicht der praktischen Arbeit in diesem Unternehmen beleuchtet.

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Video - Interview mit einem Projektleiter>

Das Unternehmen SIV AG

Gründung • Entwicklung • heutige Standorte • Unternehmensprofil • Produkte • besondere Herausforderungen

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Video - Interview mit einem Projektleiter>

Der Projektleiter

Position und Aufgaben unseres Gesprächspartners • Fähigkeiten eines Projektleiters

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Video - Interview mit einem Projektleiter>

Projekte und Prozessmodelle

Standardprodukt • Entwicklung auf Kundenwunsch • RUP • Phasen • Dokumentation • Analyse • Pflichtenheft • Vertrag • Extreme Programming • Anforderungen • Kommunikation mit Kunden

► An dieser Stelle befindet sich online ein Video.

[https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter
Projektziele](https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter Projektziele)

Aufgabenbeschreibung • Kommunikation Entwickler / Kunde • Priorisierung von Funktionalitäten

- An dieser Stelle befindet sich online ein Video.

[https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter
Projektorganisation](https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter Projektorganisation)

Organisationsformen in Abhängigkeit vom Produkt

- An dieser Stelle befindet sich online ein Video.

[https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter
Projektplanung](https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter Projektplanung)

Aufwandsschätzung • Metriken • Erfahrungen • Pflege von Anforderungen • Priorisierung von Funktionalitäten

- An dieser Stelle befindet sich online ein Video.

[https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter
Projektkontrolle](https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter Projektkontrolle)

Reviews • Terminabweichungen • Übergabetermin • Zusatzpersonal • Priorisierung von Funktionalitäten

- An dieser Stelle befindet sich online ein Video.

[https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter
Die Bedeutung des Projektmanagements in der SIV AG](https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter Die Bedeutung des Projektmanagements in der SIV AG)

Die Bedeutung des Projektmanagements im Unternehmen – für die Geschäftsleitung und für die Mitarbeiter

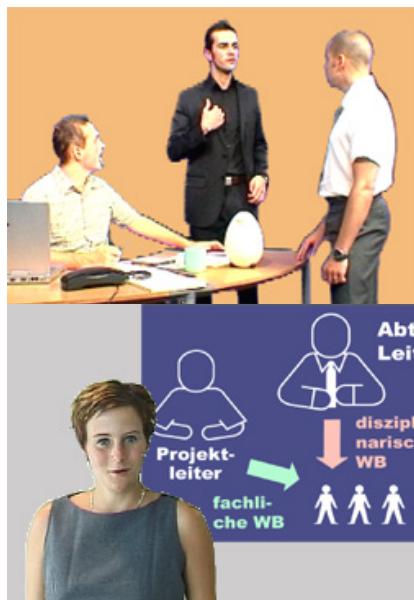
- An dieser Stelle befindet sich online ein Video.

<https://vfhpmpm.eduloop.de/loop/Video - Interview mit einem Projektleiter>

8.2.2 Video - Projektorganisation

Die Projektorganisationsformen erscheinen zunächst sehr logisch und einfach, doch wie so oft sieht es in der Praxis komplizierter aus als in der Theorie.

Schauen wir uns daher beides noch einmal in einer anderen Form an – eine Sprecherin erklärt die theoretischen Grundlagen, und eine Studentengruppe stellt die Organisationsformen als praxisnahe Spielszenen vor.



Projektorganisation: Theoretische Erläuterungen

Allgemeine Unternehmensstruktur – spezielle Organisation im Projekt • Autonome PO

- Einfluss-PO • Matrix-PO • Auftrags-PO • Linien-PO

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Video - Projektorganisation>

Unter "Autonome PO" bis "Linien-PO" werden diese Theorie-Teile im Anschluss an die jeweilige Spielszene wiederholt.

Projektgremien

Wenn von Projektorganisation die Rede ist, denkt man zunächst an die bekannten und auch hier ausführlich vorgestellten Organisationsformen. Diese betreffen jedoch nur die eigentliche Projektgruppe. Darüber hinaus gibt es jedoch noch weitere organisatorische Einheiten, die die Durchführung eines Projektes beeinflussen: die Projektgremien. Ihre Mitglieder sind zwar nicht unmittelbar z.B. an der Erstellung der Produkte beteiligt, haben aber Entscheidungen zu treffen, von denen das Gelingen oder Scheitern eines Projektes direkt abhängt.

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmp.eduloop.de/loop/Video - Projektorganisation>

Projektorganisation: Spielszenen

In diesem von einer Studentengruppe erstellten Video wird das Thema "Projektorganisation" nicht theoretisch, sondern (fast) wie im richtigen Leben dargestellt.

Sehen Sie hier die Szenen zu allen behandelten Organisationsformen im Zusammenhang.

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Unter "Autonome PO" bis "Linien-PO" werden die jeweiligen Spielszenen zusammen mit den entsprechenden Theorie-Teilen angeboten.

Projektorganisation: Geht es auch ohne?

Wenn alle Mitarbeiter gute, erfahrene Fachleute sind, sollte es doch möglich sein, ohne viele hierarchische Strukturen und ähnlichen bürokratischen Ballast einfach seine Arbeit zu machen – oder doch nicht?

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Autonome (auch: Reine) Projektorganisation

Hier werden die Spielszenen und der theoretische Film zur autonomen Projektorganisation thematisch kombiniert (Spielszene – Moderation – Theorie).

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Einfluss-Projektorganisation

Hier werden die Spielszenen und der theoretische Film zur Einfluss-Projektorganisation thematisch kombiniert (Spielszene – Moderation – Theorie).

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Matrix-Projektorganisation

Hier werden die Spielszenen und der theoretische Film zur Matrix-Projektorganisation thematisch kombiniert (Spielszene – Moderation – Theorie).

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Auftrags-Projektorganisation

Hier werden die Spielszenen und der theoretische Film zur Auftrags-Projektorganisation thematisch kombiniert (Spielszene – Moderation – Theorie).

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_Projektorganisation

Projektorganisation in Linie

Hier werden die Spielszenen und der theoretische Film zur Projektorganisation in Linie thematisch kombiniert (Spielszene – Moderation – Theorie).

- ▶ An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_-_Projektorganisation

Projektorganisation: Fazit

Welche praktische Schlussfolgerungen lassen sich ziehen?

- ▶ An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_-_Projektorganisation

8.2.3 Video - Erhebung der Projektanforderungen

Wenn Sie noch einmal einen Blick in das Lehrbuch werfen, werden Sie einen Begriff finden, der sich durch alle Kapitel zieht: Anforderungen. Sie wissen, dass es projektentscheidend ist, die richtigen Anforderungen festzulegen, um genau die Software zu entwickeln, die der Kunde tatsächlich benötigt. Sie wissen auch, dass iterative Prozessmodelle den Vorteil haben, eventuelle Missverständnisse früh zu entdecken und auszuräumen.

Aber egal wie früh Sie falsche Anforderungen entdecken – woher weiß man, welche die richtigen sind, und wie findet man diese heraus?

Verfolgen Sie die Verhandlungen unserer Projektgruppe aus der erdachten Firma "Software Today" mit ihrem Kunden, dem Chef des PC-Versandes "PC Seller".



Erhebung der Projektanforderungen

Vorstellung der Projektgruppe • Erstes Zusammentreffen mit dem Kunden • Auswertung der ersten Verhandlungsrunde • Zweites Kundengespräch und dessen Analyse • Fazit

- ▶ An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video_-_Erhebung_der_Projektanforderungen

Im Folgenden können Sie die genannten Abschnitte auch einzeln abspielen.

Die Projektgruppe

Vorstellung der Projektgruppe und des Kunden

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video - Erhebung_der_Projektanforderungen

Das erste Kundengespräch

Das erste Kundengespräch der Projektgruppe: "Was stellen Sie sich eigentlich vor, Herr Vanderberg?" – Fragen und Antworten

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video - Erhebung_der_Projektanforderungen

Auswertung des ersten Treffens

Auswertung der ersten Verhandlungsrounde: Analyse der Fragen und Antworten

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video - Erhebung_der_Projektanforderungen

Das zweite Treffen und eine Analyse

Zweites Kundengespräch und dessen Analyse: Strategie und Techniken eines Interviews

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video - Erhebung_der_Projektanforderungen

Fazit

Was hat die Projektgruppe in den Verhandlungen erreicht und wie? Wie geht es weiter?

► An dieser Stelle befindet sich online ein Video.

https://vfhpmp.eduloop.de/loop/Video - Erhebung_der_Projektanforderungen

8.2.4 Video - Führungsstile

Eine weitere Studentengruppe hat sich mit den drei Führungsstilen autoritär, laissez-faire und kooperativ auseinandergesetzt und stellt die wesentlichen Merkmale in den folgenden Szenen dar.



Sie können sich hier die Szene komplett ansehen oder im Folgenden die Führungsstile einzeln abrufen.

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmpm.eduloop.de/loop/Video - Führungsstile>

Autoritärer Führungsstil

Das Projekt steht auf der Kippe und droht zu scheitern. Als Projektleiter wird ein knallharter Krisenmanager eingesetzt.

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmpm.eduloop.de/loop/Video - Führungsstile>

"Laisser-faire" - Stil

Jetzt sehen Sie den Vorgänger des autoritären Projektleiters und erfahren, wie das Projekt mit ihm in die Krise geriet ...

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmpm.eduloop.de/loop/Video - Führungsstile>

Kooperativer Führungsstil

Ein dritter Projektleiter versucht es schließlich auf kooperative Weise. Ob er damit mehr Erfolg haben wird als seine Vorgänger?

► An dieser Stelle befindet sich online ein Video.

<https://vfhpmpm.eduloop.de/loop/Video - Führungsstile>

I Literaturverzeichnis

- Ansbacher, u.a. (Hrsg.): Alfred Adlers Individualpsychologie, Reinhardt Verlag, 1995
- Armstrong, M.: A Handbook of Human Resource Management Practice, Kogan Page, 1999
- Balzert, H.: Objektorientierung in 7 Tagen, Spektrum Verlag, 2000
- Beck, K.: eXtreme Programming explained, Addison Wesley, 2000
- Boehm, B.: Software Cost Estimation with COCOMO II, Prentice Hall PTR, 2000
- Boehm, B.W.: A Spiral Model of Software Development and Enhancement, IEEE Computer 21(5): 61-72 (1988)
- Boehm, u.a.: Identifying Quality-Requirement Conflicts, IEEE Software 13(2): 25-35 (1996)
- Booch, G.: Objektorientierte Analyse und Design, Addison Wesley, 1995
- Buhl, u.a.: XML-basiertes Projektmanagement, Java Spektrum, 2/2002
- Buhl, A.: Grundkurs Software-Projektmanagement: Einführung in das Management objektorientierter Projekte, Carl Hanser Verlag, 2004
- Burghardt, M.: Projektmanagement, Publicis MCD Verlag, 1995
- ohne Namen: Bei Projekten entscheidet wieder die Kostenfrage, Computer Zeitung Nr. 48/ 29. November 2001
- DeMarco, T.: Bärentango: Mit Risikomanagement Projekte zum Erfolg führen, Hanser Fachbuch, 2003
- DeMarco, T.: Der Termin, Hanser Verlag, Wien, 1998
- Forrester Research, December 1999, Collaboration beyond email
- Fowler, M.: Refactoring, Addison Wesley, 2000
- Fowler, M.: UML konzentriert, Addison Wesley, 1998
- Frühauf, u.a.: Software-Prüfung, vdf Hochschulverlag an der ETH Zürich, 2000
- Gamma, E.: Design Patterns, Addison Wesley, 1995
- Huber-Jahn, I.C.: Projektmanagement für kleine und mittlere Unternehmen im europäischen Binnenmarkt, Dissertation Uni Bremen, 1993
- Hürten, R.: Function Point Analysis - Theorie und Praxis, Expert Vlg., Renningen, 1999
- Kellner, H.: Leadership-Programm auf der Advanced Developers Conference, Hannover, 2001
- Kocher, H.: Projekte steuern in "Internet Time", Vortrag auf der Advanced Developers Conference, 2000

Kruchten, P.: Wie Sie ein Projekt mit dem RUP garantiert in den Sand setzen, Objektspektrum, 4, 2001

Kupper, H.: Zur Kunst der Projektsteuerung, Oldenbourg, 1996

Lorenz, u.a.: Object oriented Software Metrics, Prentice Hall, 1994

Ludewig, u.a.: Software Projektmanagement und Qualitätssicherung, B.G. Teubner Verlag, 1991

McConnell, St.: Software Project Survival Guide, Microsoft Press, 1998

Mummert & Partner, 8/2000

Myers, G.J.: Methodisches Testen von Programmen, Oldenbourg, 1995, 5. Auflage

Recklies, D.: Überblick über grundlegende Motivationstheorien, September 2001, <nowiki><http://www.themanagement.de/HumanResources/Motivationstheorien.htm></nowiki>

Reißing, R.: Extremes Programmieren, Informatik Spektrum, 23.04.2000

Sprenger, R.K.: Mythos Motivation, 20. aktualisierte Auflage, Campus Verlag, 2014

Thaller, G.E.: Software-Qualität, VDE Verlag, 2000

Versteegen, G. (Hrsg.): Software Management - Beherrschung des Lifecycles, Springer, 2002, S. 32-44

Weitere Literatur

Antons, K.: Praxis der Gruppendynamik: Übungen und Techniken, Hogrefe Verlag, 2011, 9. aktualisierte Auflage

Beermann, S.: Workshops - vorbereiten, durchführen, nachbereiten, Haufe-Lexware, 2009

Birkenbihl, V.F.: Psychologisch richtig verhandeln: Professionelle Verhandlungstechniken mit Experimenten und Übungen, mvg Verlag, 2007, 19. Auflage

Blanchard, u.a.: Führungsstile, rororo, 2002

Bohinc, To.: Projektmanagement: Soft Skills für Projektleiter, GABAL Verlag GmbH, 2011, 4. Auflage

Borbonus, R.: Die Kunst der Präsentation: Überzeugend präsentieren und begeistern - 91 Antworten für eine eindrucksvolle Präsentation ohne Show-Business, Junfermann, 2007

Bäumer, H.-P.: Programmieren mit Fortran 90, Vieweg & Sohn Verlagsgesellschaft, 1997

Deutscher Manager-Verband e.V.: Handbuch Soft Skills - Band I: Soziale Kompetenz, Vdf Hochschulverlag, 2003

Deutscher Manager-Verband e.V.: Handbuch Soft Skills - Band II: Psychologische Kompetenz, Vdf Hochschulverlag, 2004

- van Dick, u.a.: Teamwork, Teamdiagnose, Teamentwicklung: Praxis der Personalpsychologie, Hogrefe Verlag, 2005
- Edmüller, u.a.: Konfliktmanagement, Haufe-Lexware, 2010, 3. Auflage
- Engelmann, L.: Kleiner Leitfaden - Informatik und ihre Anwendungen, Paetec Gesellschaft für Bildung und Technik mbH, Berlin, 2000
- Forghani, F.: Tanz um die Macht: Geheimnisse der Verhandlungsführung, AT Edition, 2012, 2. Auflage
- Funcke, u.a.: Moderations-Tools: Anschauliche, aktivierende und klärende Methoden für die Moderations-Praxis, managerSeminare Verlags GmbH, 2011, 2. Auflage
- Glasl, F.: Selbsthilfe in Konflikten: Konzepte, Übungen, Praktische Methoden, Freies Geistesleben, 2007, 6. überarbeitete und erweiterte Auflage
- Graebig, u.a.: Wie aus Ideen Präsentationen werden - Planung, Plot und Technik für professionelles Chart-Design mit PowerPoint, Gabler Verlag, 2011
- Haberleitner, u.a.: Führen, Fördern, Coachen: So entwickeln Sie die Potenziale Ihrer Mitarbeiter, Piper, 2009, 4. Auflage
- Haeske, U.: Team- und Konfliktmanagement: Teams erfolgreich leiten - Konflikte konstruktiv lösen, Cornelsen, 2008, 3. Auflage
- Haller, R.: Mitarbeiterführung kompakt - Grundlagen, Praxistipps, Werkzeuge, Midas Management Verlag, 2009
- Hitzges, u.a.: Projektmanagement bei der Entwicklung multimedialer Anwendungen, Technical Report, Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart, 1995.
- Hofbauer, u.a.: Einstieg in die Führungsrolle: Praxisbuch für die ersten 100 Tage, Carl Hanser Verlag, 2012, 4. aktualisierte und erweiterte Auflage
- Huber, u.a.: Projektmanagement: Erfolgreicher Umgang mit Soft Factors, Vdf Hochschulverlag, 2011
- Function Point Counting Practices: Manual Release 4.0, International Function Point Users' Group, Blendonview Office Park, 5008-28 Pine Creek Drive, Westerville, OH 43001-4899, IFPUG 1994
- Kano, N.: Attractive Quality and Must-be Quality, Journal of the Japanese Society for Quality Control, H. 4, S. 39-48, 1984
- Kruchten, P.: The Rational Unified Process - An Introduction, Addison Wesley, 1999
- Krüger, W.: Teams führen, Haufe-Lexware, 2009, 5. aktualisierte Auflage

- Laufer, H.: Grundlagen erfolgreicher Mitarbeiterführung: Führungspersönlichkeit - Führungsmethoden - Führungsinstrumente, Gabal Verlag, 2012
- Lipp, u.a.: Das große Workshop-Buch: Konzeption, Inszenierung und Moderation von Klausuren, Besprechungen und Seminaren, Beltz, 2008, 8. überarbeitete und erweiterte Auflage
- Majer, u.a.: Social Competence im Projektmanagement - Projektteams führen, entwickeln, motivieren, Goldegg, 2010
- Meier, D.: Wege zur erfolgreichen Teamentwicklung, Books on Demand, 2004, 2. überarbeitete Auflage
- Moesslang, M.: Besser präsentieren - mehr erreichen: 52 Tipps für wirkungsvolle Präsentationen, Books on Demand, 2008
- Moritz, u.a.: Soft Skills für Young Professionals: Alles, was Sie für Ihre Karriere brauchen, GABAL Verlag, 2006, 3. Auflage
- Niermeyer, R.: Soft Skills: Das Kienbaum Trainingsprogramm, Haufe-Lexware GmbH, 2006, 2. Auflage
- Panse, u.a.: Erfolgsfaktor Emotionen: Ziele sicher erreichen mit Soft Skills, Redline Verlag, 2010
- Paschen, u.a.: Psychologie der Menschenführung: Wie Sie Führungsstärke und Autorität entwickeln, Springer Verlag, 2012
- Patzak, u.a.: Projektmanagement, Linde, 1998, S.434
- Peter-Kühlinger, G.: Soft Skills, Haufe Lexware GmbH, 2008, 2. Auflage
- Portner, J.: Besser verhandeln - Das Trainingsbuch, Gabal, 2010, 2. Auflage
- Reichert, O.: Netzplantechnik, Vieweg Verlag, 1994
- Rothery, u.a.: Der Leitfaden zur ISO 9000: Mit QM-Musterhandbuch und Erläuterungen, Hanser Fachbuch, 2000
- Sawhney, M.: Entwicklung eines Vorgehensmodells für die Multimedia-Anwendungsentwicklung am Beispiel eines Informations- und Orientierungssystems für eine Universität, Diplomarbeit, Universität Osnabrück, Fachgebiet Wirtschaftsinformatik, 1995
- Schmidt, Th.: Konfliktmanagement-Trainings erfolgreich leiten: Der Seminarfahrplan, Managerseminare Verlag, 2009, 3. überarbeitete Auflage
- Schmidt, M.: 25 Tipps für erfolgreiches Team-Management, Books on Demand, 2010
- Schulz, R.: Toolbox zur Konfliktlösung - Konflikte schnell erkennen und erfolgreich bewältigen, Stark Verlagsgesellschaft, 2010
- Seifert, J.W.: Besprechungen erfolgreich moderieren, Gabal Verlag, 2008, 13. Auflage

- Seifert, J.W.: Visualisieren, Präsentieren, Moderieren - Das Standardwerk, Gabal Verlag, 2009, 23. Auflage
- Seifert, J.W.: Moderation und Kommunikation: Gruppendynamik und Konfliktmanagement in moderierten Gruppen, Gabal Verlag, 1999, 6. Auflage
- Siwon, P.: Die menschliche Seite des Projekterfolgs: Was Softwerker über (verborgene) Denkautomatismen und -modelle in der Projektarbeit wissen müssen, dpunkt.verlag, 2010
- Vigenshow, u.a.: Soft Skills für Softwareentwickler: Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle, dpunkt.verlag, 2010, 2. überarbeitete und erweiterte Auflage
- Vigenshow, u.a.: Soft Skills für IT-Führungskräfte und Projektleiter: Softwareentwickler führen und coachen, Hochleistungsteams aufbauen, dpunkt.verlag, 2011, 2. aktualisierte und ergänzte Auflage
- Voltz, H.: Menschen und Computer; Markt und Technik, 1993
- Weidenmann, B.: 100 Tipps & Tricks für Pinnwand und Flipchart, Beltz, 2008, 4. überarbeitete und erweiterte Auflage
- Will, H.: Mini-Handbuch Vortrag und Präsentation: Für Ihren nächsten Auftritt vor Publikum, Beltz, 2007, 7. Auflage

II Abbildungsverzeichnis

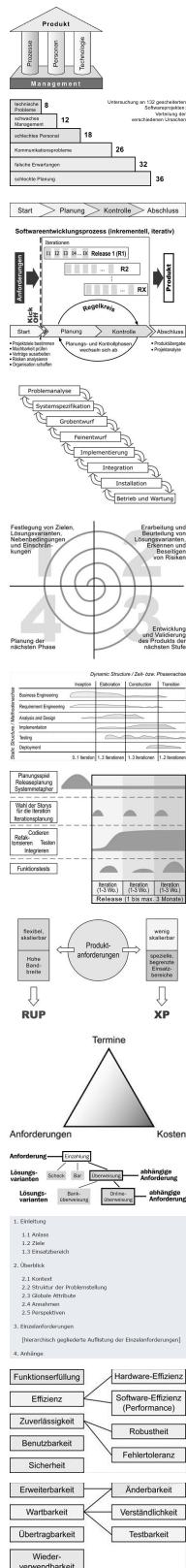


Abbildung 1.1.1: Zusammenhang zwischen Prozessen, Personen und Technologien.....9

Abbildung 1.1.2: Ursachen fuer das Scheitern von Softwareprojekten.....9

Abbildung 1.3.1.1: Projektphasen.....16

Abbildung 1.3.1.2: Zusammenhang zwischen Projektmanagement und Softwareentwicklungsprozess.....16

Abbildung 1.3.3.1: Wasserfalllebenszyklus nach BARRY BOEHM.....19

Abbildung 1.3.4.1: Spirallebenszyklus nach BARRY BOEHM.....20

Abbildung 1.3.6.1: Rational Unified Process (RUP), von der Rational Co.....22

Abbildung 1.3.7.1: eXtreme Programming (XP).....24

Abbildung 1.3.8.1: Einsatz RUP oder XP.....26

Abbildung 2.1.1.1: Das magische Dreieck des Projektmanagements.....30

Abbildung 2.1.1.2: Abhängigkeit der Anforderungen von den Lösungen.....33

Abbildung 2.1.1.3: Gliederung der Software Requirements Specification nach der IEEE 830-1993.....34

Abbildung 2.1.3.1: Qualitätsmerkmale für Software aus Benutzersicht nach BARRY BOEHM.....36

Abbildung 2.1.3.2: Qualitätsmerkmale aus Entwicklersicht nach BARRY BOEHM.....36

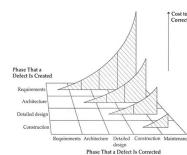


Abbildung 2.2.1.1: Verteilung der Kosten für die Fehlerkorrektur..... 42

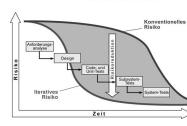


Abbildung 2.2.1.2: Risikoverteilung in Abhängigkeit vom Prozessmodell..... 43



Abbildung 2.3.2.1: Reine Projektorganisation nach BURGHARDT..... 48

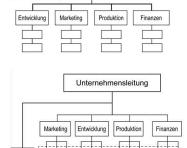


Abbildung 2.3.2.2: Einfluss-Projektorganisation nach BURGHARDT..... 49



Abbildung 2.3.2.3: Matrix-Projektorganisation..... 49

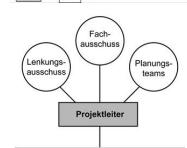


Abbildung 2.3.2.4: Auftragsprojektorganisation aus BURGHARDT..... 50

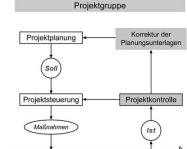


Abbildung 2.3.3.1: Projektgremien nach BURGHARDT..... 52



Abbildung 3.1.1.1: Planung als Regelkreis..... 58

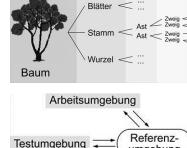


Abbildung 3.2.1.1: Planungsreihenfolge nach Kupper..... 63

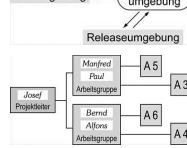


Abbildung 3.2.2.2: Hierarchieebenen am Beispiel eines Baumes..... 65

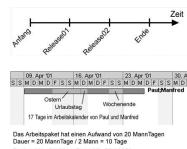


Abbildung 3.2.2.4: Zentrale Rolle der Referenzumgebung..... 67

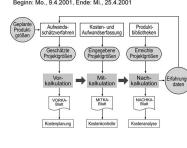


Abbildung 3.2.2.6: Projektstruktur..... 68

Das Arbeitspaket hat einen Aufwand von 20 Monaten
Dauer = 20 Monate / 2 Mann = 10 Tage
Beginn Mo. 9.4.2001, Ende Mi. 29.4.2001

Abbildung 3.2.4.1: Darstellung von Hauptetappen auf der Zeitachse innerhalb der Grobplanung..... 72

Auszug aus der Detailplanung unter Verwendung eines Balkendiagramms

Das Arbeitspaket hat einen Aufwand von 20 Monaten
Dauer = 20 Monate / 2 Mann = 10 Tage
Beginn Mo. 9.4.2001, Ende Mi. 29.4.2001

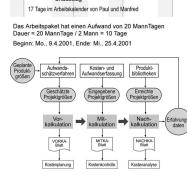


Abbildung 3.2.4.2: Auszug aus der Detailplanung unter Verwendung eines Balkendiagramms..... 72

Das Arbeitspaket hat einen Aufwand von 20 Monaten
Dauer = 20 Monate / 2 Mann = 10 Tage
Beginn Mo. 9.4.2001, Ende Mi. 29.4.2001

Abbildung 3.2.5.1: Kalkulation der Projektkosten..... 73

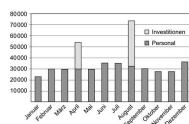


Abbildung 3.2.5.2: Monatlicher Kostenplan, grafisch..... 74

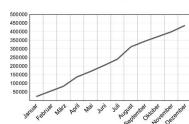


Abbildung 3.2.5.3: Kostensummenlinie..... 75

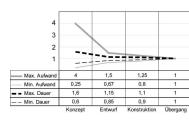


Abbildung 3.3.1.1: Abhängigkeit des Aufwandes von den Prozessphasen..... 77



Abbildung 3.3.2.1: kVASy®-Maske Kaufmännische Kundendaten..... 79

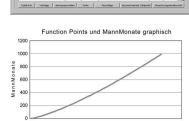


Abbildung 3.3.2.4: Function Points und Maßnahmen graphisch..... 82



Abbildung 3.3.4.1: Netzplantechnik, grafische Darstellung eines Knotens..... 87



Abbildung 3.3.4.2: Netzplantechnik, grafische Elemente..... 89

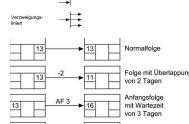


Abbildung 3.3.4.4: Netzplantechnik: Besonderheiten bei der Darstellung von Folgen..... 90

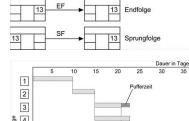


Abbildung 3.3.5.1: Balkendiagramm, Vorgänge über der Dauer..... 91

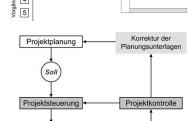


Abbildung 4.1.1: Kontrolle als Element im Regelkreis der Projektplanung..... 95



Abbildung 4.1.2: Rechtzeitiges Erkennen von Planabweichungen..... 96

Arbeitsstunden	
Start	Ende
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00

Abbildung 4.1.1.1.1: Erfassung der Stunden für geplante Arbeitspakete..... 98

Ungeduldige Tätigkeiten	
Start	Ende
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00

Abbildung 4.1.1.1.2: Erfassung der Stunden für ungeplante Aufgaben..... 99

Ressourcen	
Erreichbar	Durchsetz.
Erreichbar	Erreichbar
Erreichbar	Erreichbar
Erreichbar	Erreichbar

Abbildung 4.1.1.1.3: Erfassung der verwendeten Ressourcen..... 99

Ausfallzeiten	
Start	Ende
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00
Montag 20.01.2011 08:00	Montag 20.01.2011 16:00

Abbildung 4.1.1.1.4: Erfassung der Ausfallzeiten..... 100

Problemübersicht	
Bereich	Erreichbar
Entwicklungs	Erreichbar
Test	Erreichbar
Technik	Erreichbar
Logistik	Erreichbar

Abbildung 4.1.1.1.5: Erfassung von Problemen und Lösungsansätzen..... 100

Wir schätzen der Mitarbeiter die Projektqualität ein!	
Checkliste	Checkliste hier Projektmanagementkommunikation
Bewertung	sehr gut C C C C C sehr schlecht
Bemerkung	Projektmanagement ist geprägt durch eine schlechte Kommunikation
Markierung:	
Checkliste	Checkliste hier Anlagenprojektmanagement
Bewertung	sehr gut C C C C C sehr schlecht
Bemerkung	Technikpraktiken Soll nicht Kritisch
Markierung:	
Checkliste	Checkliste hier Risikomanagement
Bewertung	sehr gut C C C C C sehr schlecht
Bemerkung	Frühestens Ende August kann Risiko eingeschätzt werden
Markierung:	
Checkliste	Checkliste hier Projektmanagement
Bewertung	sehr gut C C C C C sehr schlecht
Bemerkung	Frühestens Ende August kann Risiko eingeschätzt werden
Markierung:	

Abbildung 4.1.1.2.1: Ausschnitt aus einem computergestützten Reviewprotokoll.....102

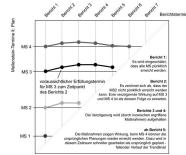


Abbildung 4.2.1.1: Meilensteinentrendanalyse, normale Verläufe von MS 3 und 4, MS 2 weist einen ansteigenden Verlauf auf.....105

Abbildung 4.2.3.1: Kostenvergleichsdiagramm.....108

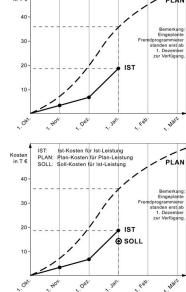


Abbildung 4.2.3.2: Kostenvergleichsdiagramm unter Berücksichtigung der Soll-Kosten für die erbrachten Leistungen zum Stichtag.....108

Abbildung 4.2.4.2: Grafische Auswertung der Fertigstellungsgrade (Werte aus Tabelle 4.2.4.1).....111

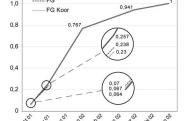


Abbildung 4.2.4.3: Aufbau eines Projektfortschrittsberichts.....111



Abbildung 4.2.5.2: Übersicht über Qualitätssicherungsmaßnahmen.....114



Abbildung 4.2.5.3: Aufbau eines Projektqualitätsberichts.....116



Abbildung 5.1.1.1: Schematische Darstellung eines Black-Box-Tests.....120



Abbildung 6.1.1.2: Erhöhung der Motivation durch auf den Mitarbeiter abgestimmte Anreize 134



Abbildung 6.1.1.3: Faktoren, die das Verhalten von Menschen beeinflussen..... 135



Abbildung 7.5.1..... 160

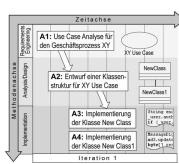


Abbildung 7.7.1: Arbeitspakete A1 bis A4 unter Berücksichtigung von Produkt und Prozess 162

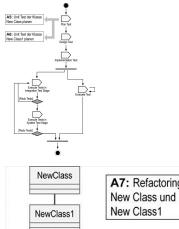


Abbildung 7.7.3: Arbeitspakete A7 und A8 in Folge von Refactoring.....165

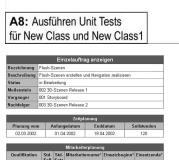


Abbildung 7.7.4: Computergestütztes Formular für einen Einzelauftrag.....165



Abbildung 7.8.1: Projektplan für das eXtreme Programming (1) 166

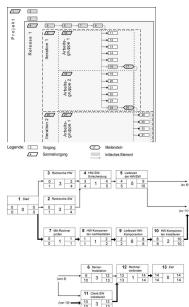


Abbildung 7.8.3: Netzplanauszug mit Vorgangsnummern aus dem Projektplan lt. Abbildung 7.8.1 und Abbildung 7.8.2.....168

Abbildung 7.9.2: Netzplan in graphischer Darstellung; der kritische Pfad ist mit fetter Linie dargestellt..... 170

III Tabellenverzeichnis

Tabelle 1.2.3.1: Weiche und harte Faktoren des Projektmanagements.....	14
Tabelle 3.2.2.1: Strukturplanung.....	64
Tabelle 3.2.2.3: Tabelle für die Versionierung von Dokumenten.....	66
Tabelle 3.2.2.5: Ablaufplan der Arbeitspakete A1 bis A6.....	68
Tabelle 3.3.2.2: Bewertungsmatrix Externe Eingabe.....	80
Tabelle 3.3.2.3: Function Point Methode, Tabelle zur Berechnung der MannMonate.....	81
Tabelle 3.3.3.1: Umrechnungsverhältnis von UFP in SLOC nach [Boe00].....	84
Tabelle 3.3.4.3: Netzplantechnik, tabellarische Darstellung.....	90
Tabelle 4.2.4.1: Beispielrechnung zur Bestimmung der unterschiedlichen Fertigstellungsgrade.....	110
Tabelle 4.2.4.4: Bestimmung des Stabilitätsindex.....	112
Tabelle 4.2.5.1: Auftreten von Softwarefehlern.....	114
Tabelle 7.9.1: Vorgangsliste.....	169

IV Aufgabenverzeichnis

Aufgabe 1.2.4-1.....	14
Aufgabe 1.2.4-2.....	14
Aufgabe 1.2.4-3.....	15
Aufgabe 1.3.9-1.....	27
Aufgabe 1.3.9-2.....	27
Aufgabe 1.3.9-3.....	27
Aufgabe 1.3.9-4.....	27
Aufgabe 1.3.9-5.....	28
Aufgabe 2.1.5-1.....	40
Aufgabe 2.1.5-2.....	40
Aufgabe 2.1.5-3.....	40
Aufgabe 2.1.5-4.....	40
Aufgabe 2.1.5-5.....	40
Aufgabe 2.1.5-6.....	41
Aufgabe 2.2.3-1.....	45
Aufgabe 2.2.3-2.....	45
Aufgabe 2.2.3-3.....	46
Aufgabe 2.2.3-4.....	46
Aufgabe 2.2.3-5.....	46
Aufgabe 2.2.3-6.....	46
Aufgabe 2.3.4-1.....	55
Aufgabe 2.3.4-2.....	55
Aufgabe 2.3.4-3.....	55
Aufgabe 2.3.4-4.....	56
Aufgabe 2.3.4-5.....	56
Aufgabe 3.1.4-1.....	61
Aufgabe 3.1.4-2.....	61
Aufgabe 3.1.4-3.....	62
Aufgabe 3.2.6-1.....	75
Aufgabe 3.2.6-2.....	75
Aufgabe 3.2.6-3.....	76
Aufgabe 3.2.6-4.....	76
Aufgabe 2.2.2.6-1.....	92
Aufgabe 2.2.2.6-2.....	92
Aufgabe 2.2.2.6-3.....	92
Aufgabe 2.2.2.6-4.....	93
Aufgabe 2.2.2.6-5.....	93
Aufgabe 2.2.2.6-6.....	94
Aufgabe 4.1.2-1.....	103
Aufgabe 4.1.2-2.....	103
Aufgabe 4.1.2-3.....	103
Aufgabe 4.2.6-1.....	117
Aufgabe 4.2.6-2.....	117
Aufgabe 4.2.6-3.....	117
Aufgabe 5.1.3-1.....	124
Aufgabe 5.1.3-2.....	124
Aufgabe 5.1.3-3.....	125
Aufgabe 5.2.3-1.....	131
Aufgabe 5.2.3-2.....	131
Aufgabe 5.2.3-3.....	131
Aufgabe 5.2.3-4.....	131
Aufgabe 6.1.4-1.....	142
Aufgabe 6.1.4-2.....	142
Aufgabe 6.1.4-3.....	142
Aufgabe 6.1.4-4.....	142

Aufgabe 6.2.4-1.....	153
Aufgabe 6.2.4-2.....	154
Aufgabe 6.2.4-3.....	154
Aufgabe 6.2.4-4.....	154