4. Software-Projektführung

Da Software ein immaterielles Produkt ist, führt eine ad-hoc-Abwicklung bei Software-Projekten wesentlich schneller ins Desaster als bei konventionellen Projekten. Es ist daher notwendig, Software-Projekte systematisch zu führen. Dazu gehören eine sorgfältige Planung, fortlaufende Kontrolle und nötigenfalls Korrektur des Projektablauf. Ferner muss den Projektrisiken besondere Beachtung geschenkt werden. Wir konzentrieren uns in diesem Kapitel auf Fragen, die spezifisch für Software-Projekte sind. Allgemeine Probleme der Projektführung, beispielsweise die Führung von Projektmitarbeitern, werden nicht behandelt.

4.1 Projektplanung

Die Projektplanung schafft die notwendigen Voraussetzungen für das Gelingen eines Projekts. Neben der Definition der zu erreichenden Ziele geht es vor allem darum, den Weg zum Ziel und die benötigten Mittel festzulegen. Dabei sind im Wesentlichen folgende Aufgaben zu erledigen: die Festlegung des zu verwendenden Prozessmodells und der Organisationsstruktur des Projekts, die Bestimmung der Anzahl der benötigten Personen und die Planung ihres Einsatzes, das Aufstellen eines Terminplan und eines Kostenplans, die Festlegung der zu erstellenden Dokumente und der anzuwendenden Verfahren für Entwicklung, Konfigurationsmanagement und Qualitätsmanagement sowie die Abschätzung und Behandlung der Projektrisiken.

4.1.1 Prozessmodell und Organisationsstruktur

Zu den ersten Planungsschritten gehören die Auswahl des Prozessmodells und der Organisationsstruktur für das Projekt auf der Grundlage einer Analyse des Projektkontextes und -inhalts.

Die prinzipielle Struktur der möglichen Prozessmodelle und ihre Eignung in verschiedenen Situationen wurden in Kapitel 3 diskutiert. In der Planung eines konkreten Projekts wird das Prozessmodell konkretisiert und mit einer personellen Organisationsstruktur verbunden. Die personelle Organisationsstruktur legt die Funktionen, Aufgaben, Verantwortlichkeiten und gegenseitigen Beziehungen der am Projekt beteiligten Personen fest.

Die Organisation von Software-Projekten wird hier nicht näher behandelt, da grundsätzlich die gleichen Regeln gelten wie für andere Projekte. Es werden nur zwei Aspekte kurz diskutiert: die Rolle des Projektleiters und die Beziehungen der Projektbeteiligten untereinander.

Die Projektleiterin ist eine Schlüsselfigur in jedem Software-Projekt. Ihre Kompetenzen und Verantwortlichkeiten müssen klar geregelt werden. Anzustreben ist eine *Führung durch Zielsetzung*, d.h. die Auftraggeber bzw. die Verantwortlichen in der Linienorganisation erteilen der Projektleiterin einen *Auftrag* und statten sie mit den dafür notwendigen personellen und materiellen *Ressourcen* aus. Im Rahmen dieses Auftrags handelt die Projektleiterin *eigenverantwortlich*. Sie berichtet der vorgesetzten Stelle regelmäßig über den Projektablauf und informiert insbesondere über Störungen im Ablauf, die mit projekteigenen Mitteln nicht behoben werden können.

Die Beziehungen der Projektbeteiligten untereinander hängen von der Zahl der Personen ab. Bei wenigen Personen ist eine demokratische Struktur möglich, welche Weinberg (1971) idealtypisch als "egoless team" bezeichnet (Bild 4.1 a). Alle Beteiligten sind gleichberechtigt. Probleme und anstehende Entscheidungen werden ausdiskutiert. Die Verantwortung wird gemeinsam getragen. Der Projektleiter ist in dieser Struktur Vorsitzender, aber nicht Chef.

© 1996, 2004 by Martin Glinz. Alle Rechte vorbehalten. Reproduktion zum nicht kommerziellen Gebrauch mit Quellenangabe gestattet. Reproduktion - auch auszugsweise - zum kommerziellen Gebrauch nur mit schriftlicher Bewilligung des Verfassers gestattet.

Bei mehr als ca. fünf Beteiligten funktionieren demokratische Teams nicht mehr, weil der Kommunikationsbedarf zu stark steigt (vgl. Bild 1.6). Durch hierarchische Strukturen (ein- oder zweistufig je nach Personenzahl) wird der Kommunikationsbedarf kanalisiert. Die Projektleiterin steht an der Spitze dieser Hierarchie, ist den Projektbeteiligten gegenüber weisungsbefugt und trägt die Projektverantwortung allein (Bild 4.1 b).

Bei allen Organisationsformen ist anzustreben, dass möglichst viele der Projektbeteiligten möglichst ausschließlich für dieses Projekt arbeiten. Je mehr Projekte eine Person gleichzeitig bearbeitet, desto höher werden die Umschaltverluste beim Übergang von der Arbeit am Projekt X zu der am Projekt Y.

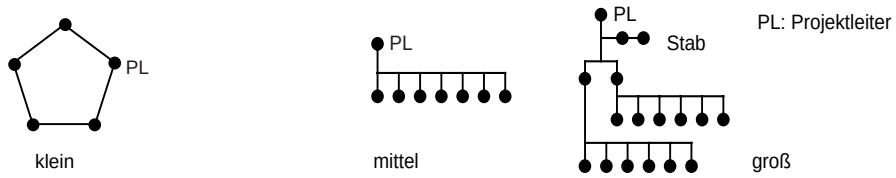


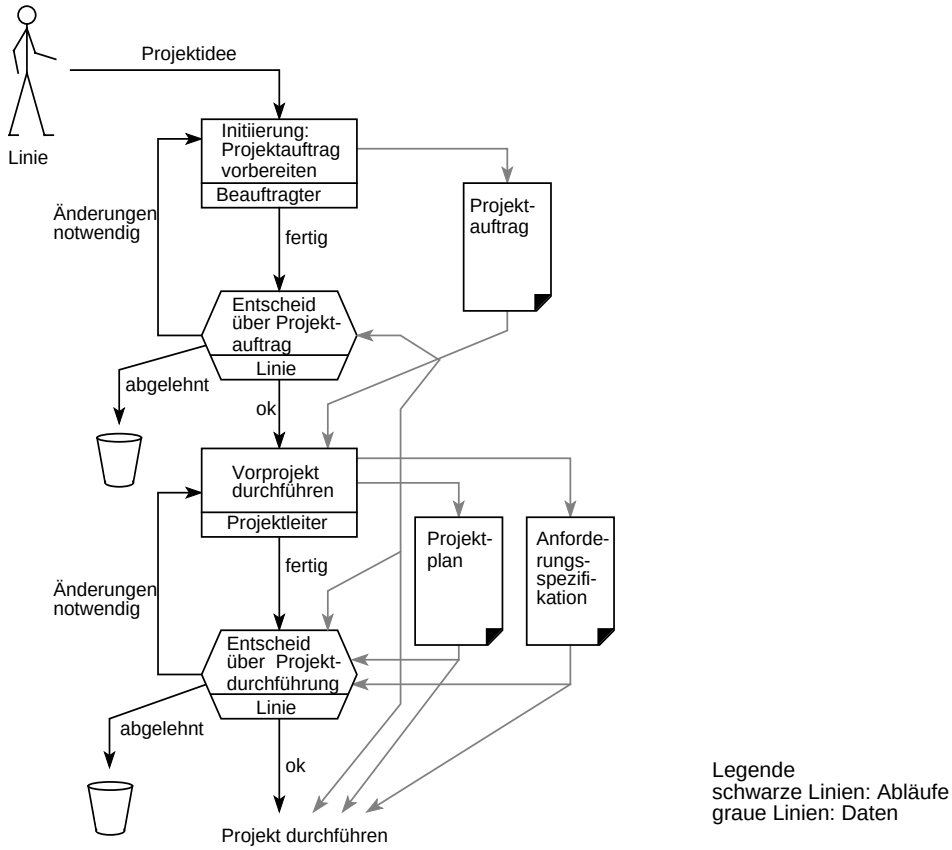
BILD 4.1 A. Demokratisches Team **B.** Hierarchisch organisiertes Team

4.1.2 Der Projektstart

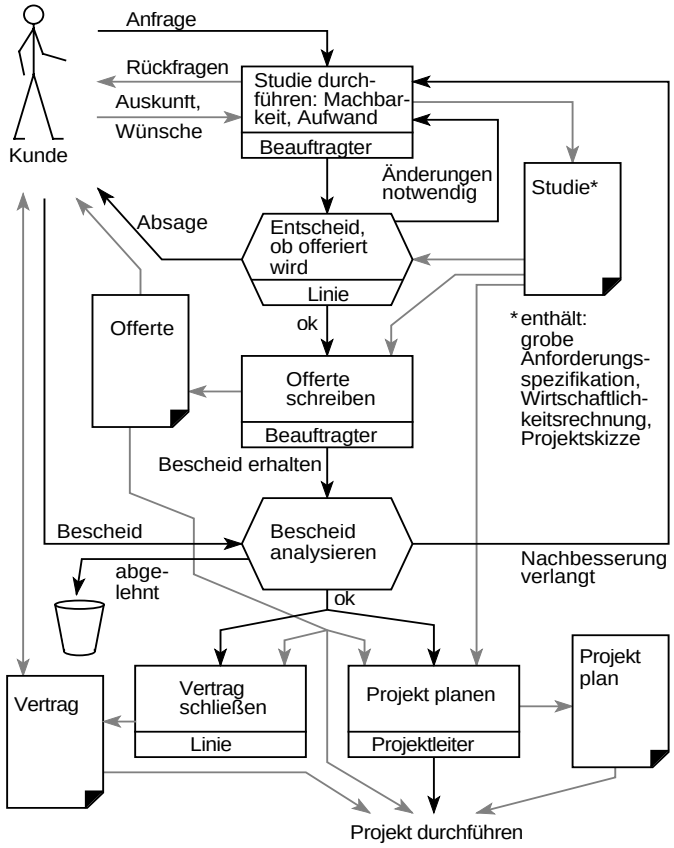
Der Projektstart ist eine der kritischen Phasen eines Projekts. Hier werden die Weichen gestellt, die das Projekt auf einen guten Weg oder auf den Holzweg führen.

Es empfiehlt sich, jedem Projekt ein *Vorprojekt* vorzuschalten, in dem das Projekt aufgesetzt wird und wichtige Definitions- und Planungsarbeiten ablaufen. Die Muster 4.1 bis 4.4 zeigen mögliche Modelle für ein Vorprojekt für vier typische Entwicklungssituationen. An Hand dieser Muster wird gezeigt, wo welche Planungsaufgaben anfallen.

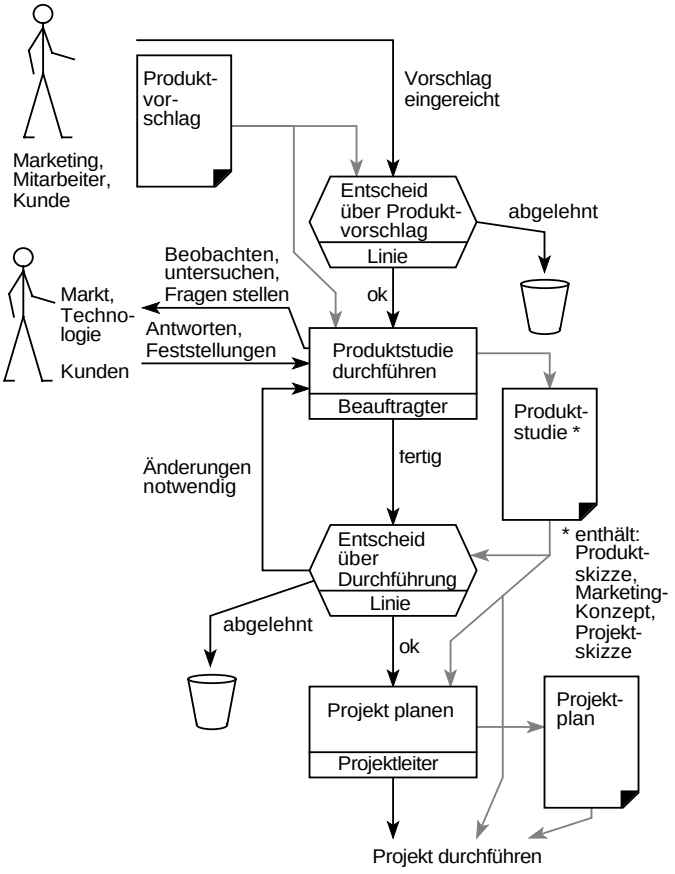
MUSTER 4.1. Start eines unternehmensinternen Softwareprojekts



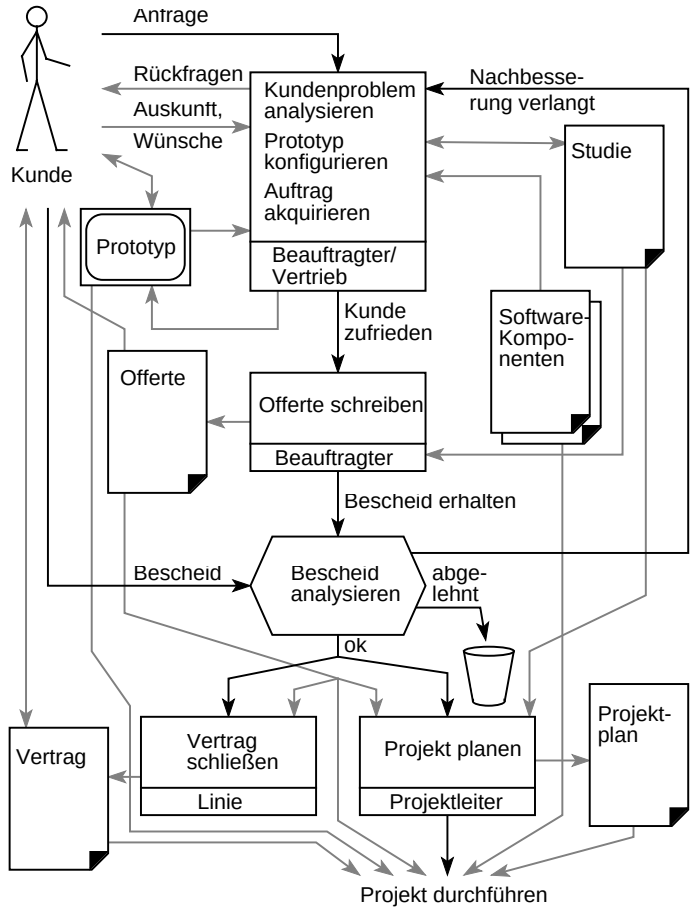
MUSTER 4.2. Start eines externen Auftragsprojekts für Software



MUSTER 4.3. Start eines Produktentwicklungsprojekts für Software



MUSTER 4.4. Start eines Software-Konfigurationsprojekts



4.1.3 Der Projektplan

Im Projektplan werden alle Ergebnisse der Planung sowie alle Planungsfortschreibungen dokumentiert. Er muss Antworten auf die folgenden sechs W-Fragen geben:

- WARUM: Veranlassung und Projektziele
- WAS: die zu liefernden Resultate (Produktziele)
- WANN: die geplanten Termine
- DURCH WEN: Personen und ihre Verantwortlichkeiten
- WOMIT: die zur Verfügung stehenden Mittel (Geld, Geräte, Software...)
- WIE: die Vorgehensweise und die Maßnahmen zur Sicherstellung des Projekterfolgs.

Muster 4.5 zeigt eine mögliche Gliederung eines Projektplans.

MUSTER 4.5. Mögliche Gliederung eines Projektplans

1. Einleitung
Projektname, Anlass, Veranlasser, etc.
2. Ziele
Projektziele (Verweis auf die Anforderungsspezifikation für die Produktziele)
3. Arbeitsplan
3.1 Arbeitspakete
Was zu tun ist und wie die Arbeit gegliedert wird
3.2 Lieferung und Abnahme
Was zu liefern ist, wie und von wem es abgenommen wird
3.3 Risiken
Liste der Risiken und der geplanten Maßnahmen zu ihrer Steuerung

4. Terminplan
Was wann fertiggestellt sein soll
5. Personalplan
5.1 Projekt-Organigramm
wie das Projekt personell strukturiert ist
5.2 Personal-Einsatzplan
welche Person welche Aufgabe aus dem Arbeitsplan bearbeitet
6. Kostenplan
geplante Kosten und ihre Verteilung über die Projektdauer
7. Übrige Ressourcen
sonstige benötigte Hilfsmittel (Entwicklungsmaschinen, Software, etc.)
8. Vorgehen
Prozessmodell, Einsatz von Methoden und Werkzeugen, Qualitäts- und Konfigurationsmanagement, zu erstellende Dokumente...
(existiert eine Entwicklungsrichtlinie und/oder ein Q-Plan, so kann das meiste durch Verweis auf diese Dokumente erledigt werden)

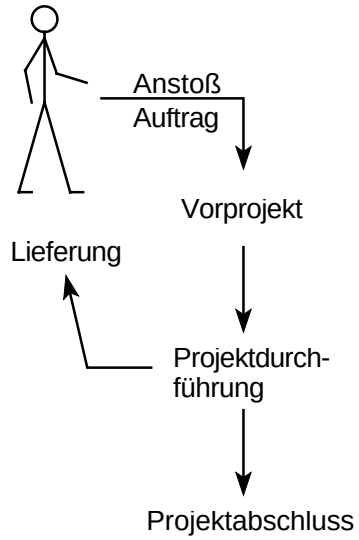
4.1.4 Planung bei linearen und bei inkrementellen Prozessen

Bei linearen Prozessmodellen, zum Beispiel dem ergebnisorientierten Phasenmodell, erfolgt die Planung im Vorprojekt. Danach wird die Planung nur noch fortgeschrieben (vgl. 4.2).

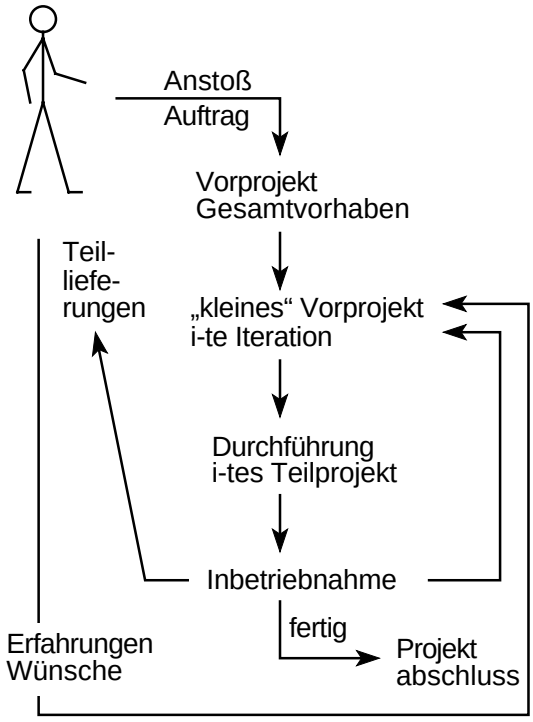
Bei Wachstumsmodellen gibt es mehrere Planungsabschnitte. Hier muss zunächst in einem Vorprojekt eine Grobplanung erfolgen, welche insbesondere die Erstellung des Lieferplans umfasst. Jedes Inkrement wird in einem „kleinen“ Vorprojekt dann geplant, wenn es zur Ausführung ansteht. Die Gesamtplanung wird revidiert, wenn es aufgrund der realisierten Inkremente zu Änderungen im Projekt kommt oder wenn unkorrigierbare Abweichungen von der Planung auftreten.

Die Muster 4.6 und 4.7 zeigen die beiden Planungsstile.

MUSTER 4.6. Lineare Planung



MUSTER 4.7. Inkrementelle Planung



4.1.5 Planungs-Hilfsmittel

Arbeits-, Termin-, Kosten- und Personaleinsatzpläne sollten graphisch in Diagrammen dargestellt werden. Sie werden so angelegt, dass im Projektablauf ein SOLL/IST-Vergleich geführt werden kann.

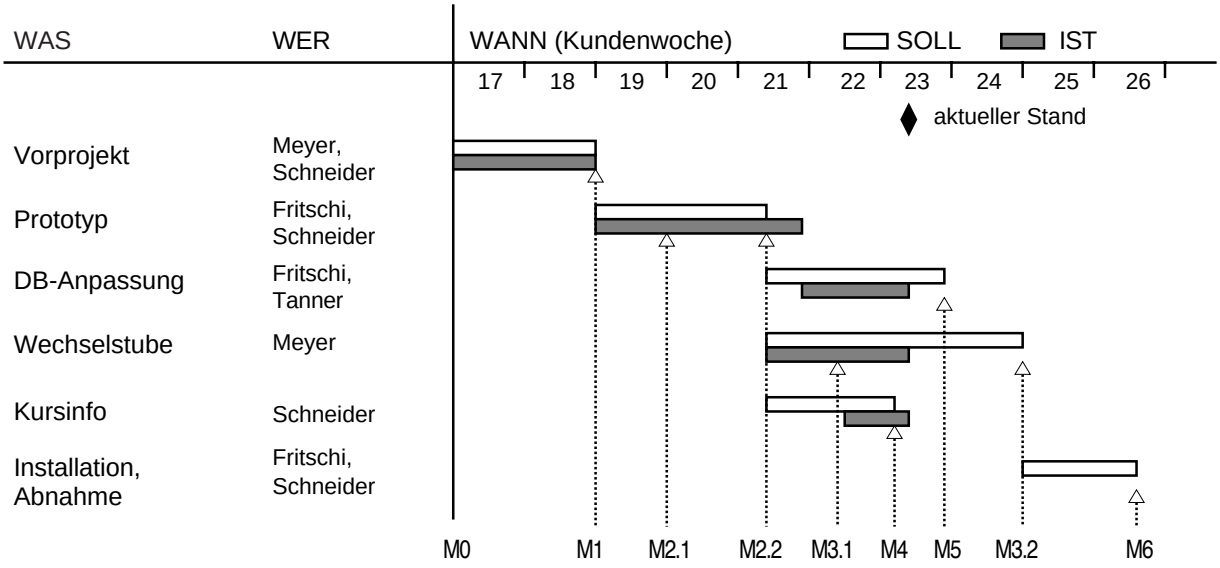


BILD 4.2. Beispiel eines kombinierten Arbeits-/Personaleinsatz-/Terminplans für ein kleines Projekt

Es gibt heute eine Vielzahl von Projektführungs-Werkzeugen für Rechner zu kaufen, mit denen man solche Diagramme erstellen und nachführen sowie SOLL/IST-Vergleiche durchführen kann.

Es ist außerordentlich wichtig, dass die geplanten und die tatsächlichen Aufwendungen möglichst genau erfasst und dokumentiert werden. Denn nur auf der Grundlage verlässlicher Zahlen über abgewickelte Projekte sind gute Aufwandsabschätzungen für neue Projekte möglich.

4.2 Projektkontrolle und -lenkung

Planung ist nur sinnvoll, wenn auch kontrolliert wird, ob der Arbeitsfortschritt der Planung entspricht. Fortschrittskontrollen müssen messbare Ergebnisse haben, wenn sie nützlich sein sollen. Andernfalls gerät man rasch in die Falle des 90%-Syndroms (Bild 4.3). Die wichtigsten Kontroll-Maßnahmen sind Termin-, Sachziel-, Kosten- und Risikoverfolgung.

Terminverfolgung geschieht mit Hilfe von Meilensteinen. In der Planung hat jeder Meilenstein einen Soll-Termin. Jedesmal, wenn ein Meilenstein erreicht ist, kann durch Vergleich zwischen Soll- und Ist-Termin eine *gesicherte quantitative Aussage* über die Terminsituation gemacht werden. Jedesmal, wenn ein Solltermin für einen Meilenstein verstreicht, ohne dass der Meilenstein erreicht ist, kann durch eine Schätzung des verbleibenden Aufwands bis zur Erreichung des Meilensteins eine *Schätzung* über die Terminsituation abgegeben werden.

Sachzielverfolgung geschieht ebenfalls mit Hilfe von Meilensteinen, indem jeder erreichte Meilenstein ein fertiggestelltes Zwischenergebnis dokumentiert.

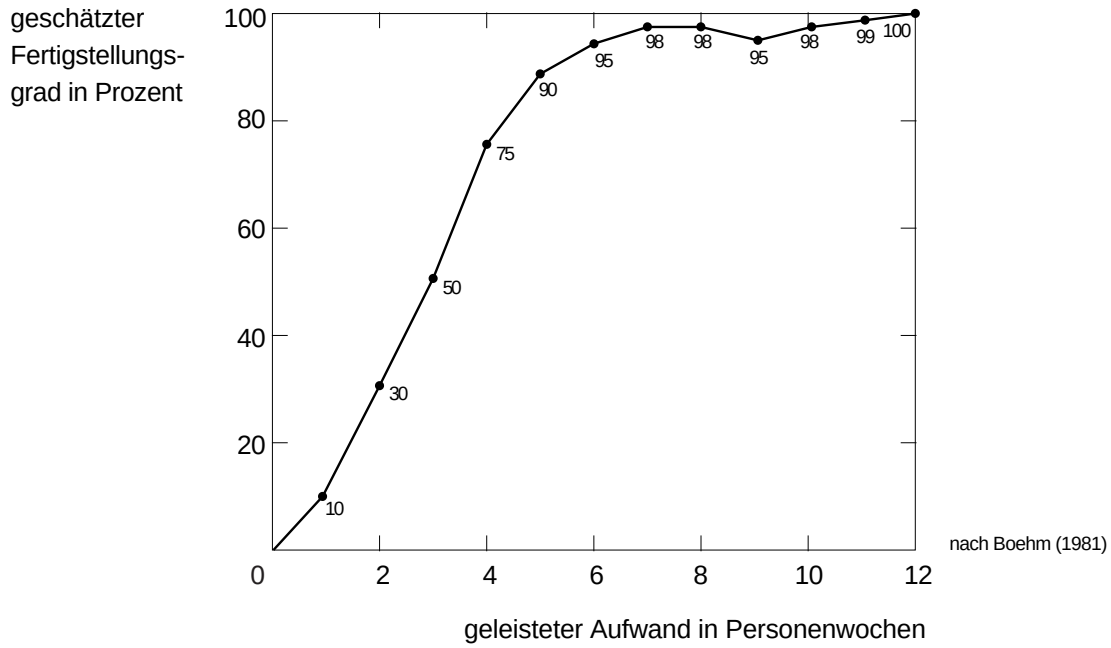


BILD 4.3. Das 90%-Syndrom in der Projektkontrolle

Die scheinbar einfachste Art der *Kostenverfolgung* ist das Aufzeichnen von budgetierten und tatsächlichen Kosten über der Zeit. Bei näherem Hinschauen stellt man aber fest, dass eine solche Graphik für die Projektkontrolle wertlos ist, da der Projektfortschritt nicht berücksichtigt wird. Die gleiche Kurve kann bei einem termingerechten Projekt eine Kostenunterschreitung und bei einem in Verzug geratenen Projekt eine massive Kostenüberschreitung bedeuten. Stattdessen muss entweder der Projektfortschritt als Gewinn bewertet und mit in die Kurve der tatsächlichen Kosten einbezogen werden, oder die Kosten müssen über dem Projektfortschritt statt über der Zeit aufgetragen werden. Ersteres ist aufwendig und wird daher nur selten angewendet. Einfacher ist es, wieder die Meilensteine heranzuziehen und die Kosten über den Meilensteinen aufzutragen. Bei den Ist-Kosten werden dabei die Kosten zum Zeitpunkt der Erreichung des Meilensteins genommen.

Risikoverfolgung ist in Kapitel 4.4 beschrieben.

Ein nützliches Hilfsmittel in der Kontrolle vor allem größerer Software-Projekte ist ein *organisiertes Berichtswesen* (z.B. standardisierte Wochen- und Monats-Fortschrittsberichte). Diese haben im Wesentlichen eine Frühwarnfunktion, indem sie Probleme und Störungen melden, bevor sich diese in der verspäteten Erreichung von Meilensteinen manifestieren.

Bei großen Projekten mit vielen beteiligten Personen haben sich sogenannte *Arbeitspaket-Ordner* (engl. Unit Development Folder) bewährt. In diesen werden alle Ergebnisse sowie alle sonstigen Papiere zu jeweils einem Arbeitspaket eingeheftet. Jeder Ordner enthält ein Leitblatt, auf dem die Termine, Verantwortlichkeiten und Arbeitsfortschritte für das zugehörige Arbeitspaket festgehalten sind. Die Arbeitspaket-Ordner können physisch als Ordner oder logisch als Dateien auf einem Rechner geführt werden.

Werden bei der Projektkontrolle größere Abweichungen erkannt, so müssen die Ursachen dafür ergründet und entsprechende *Lenkungsmaßnahmen* ergriffen werden. Wird beispielsweise festgestellt, dass ein Projekt aus dem Terminplan läuft, so können folgende Lenkungsmaßnahmen in Betracht gezogen werden:

- die Bereitstellung zusätzlicher Ressourcen
- die Befreiung von Projektmitarbeitern von Verpflichtungen außerhalb des Projekts
- die Anordnung von Überstunden

- die Vergabe von Teilaufträgen an Dritte
- Abstriche bei den zu erreichenden Sachzielen
- Etappierung der Sachziele (nach der Art eines Wachstumsmodells).

Wenn erkannt wird, dass der Projektplan trotz Gegenmaßnahmen nicht mehr einhaltbar ist, so muss der Projektplan den neuen Gegebenheiten angepasst werden. Nach unrealistischen Plänen zu arbeiten ist demotivierend und wenig effektiv.

Bild 4.4 zeigt das Prinzip eines gelenkten Prozesses. Allerdings muss man sich darüber klar sein, dass ein Software-Projekt ein komplexes nichtlineares System ist. Das bedeutet, dass es in der Regel keine einfachen proportionalen Zusammenhänge zwischen einer Lenkungsmaßnahme und der beabsichtigten Wirkung gibt. Insbesondere muss bei fast jeder Lenkungsmaßnahme in Betracht gezogen werden, welche unbeabsichtigten Nebenwirkungen sie erzeugt.

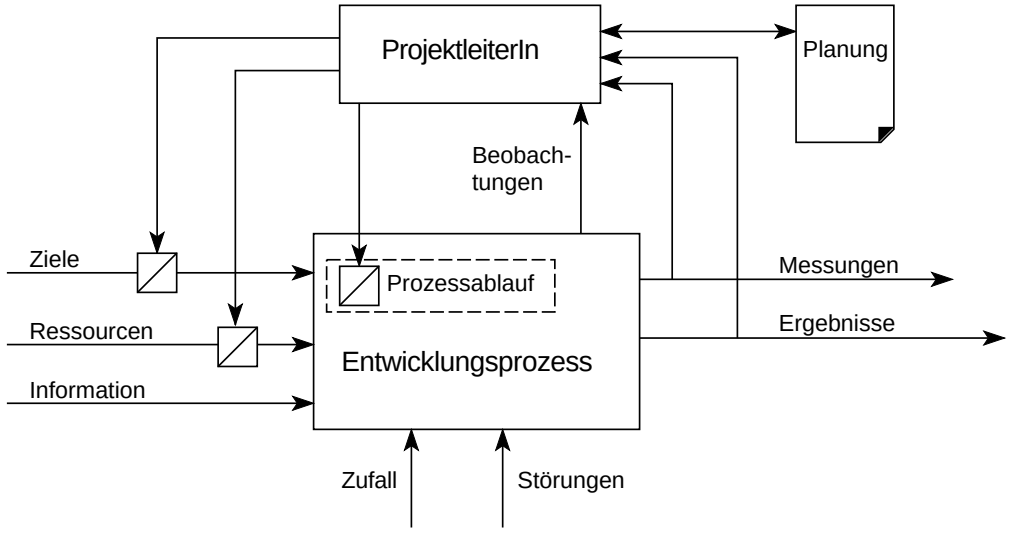


BILD 4.4. Lenkung eines Software-Entwicklungsprojekts

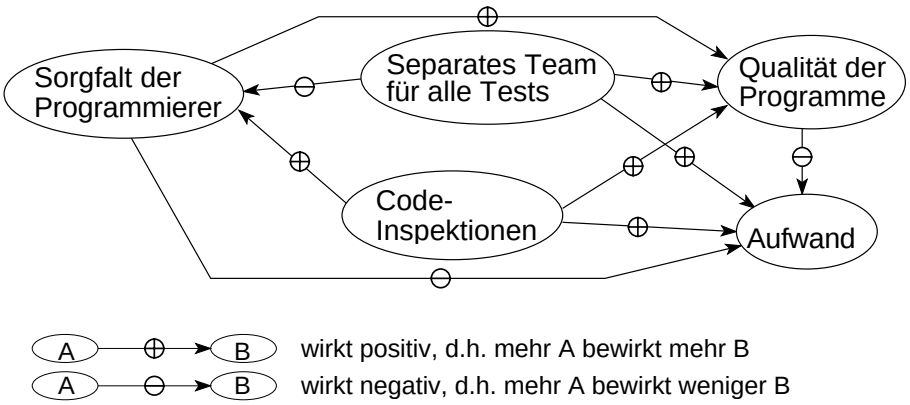


BILD 4.5. Ausschnitt aus dem Systemmodell eines Projekts, welcher Wirkungen und Nebenwirkungen möglicher Lenkungsmaßnahmen in einer gegebenen Projektsituation illustriert (vgl. Text).

Bild 4.5 zeigt ein Beispiel. Angenommen wird ein Projekt, in welchem die Projektleiterin durch entsprechende Messungen festgestellt hat, dass die erzeugte Software qualitativ zu schlecht ist, d.h. zu viele Fehler enthält. Bisher war es in diesem Projekt so, dass die Programmierer auch für das Testen der Software zuständig waren. Teilweise wurden Programme auch durch Inspektionen (vgl. Kapitel 9) geprüft. Die Projektleiterin erwägt nun als Lenkungsmaßnahmen eine Intensivierung der Inspektionen und/oder die Einführung einer von den Program-

mierern unabhängigen Gruppe von Testern, welche alle Tests übernimmt. Bild 4.5 zeigt einen Ausschnitt aus einem Systemmodell des Projekts, an Hand dessen die Wirkung der intendierten Maßnahmen abgeschätzt werden kann. Wichtig ist insbesondere die Untersuchung negativer Wirkungen. Die Einführung einer Testgruppe, welche alle Tests übernimmt, wirkt zwar (wie beabsichtigt) positiv auf die Qualität der Software, aber auch (unbeabsichtigt) negativ auf die Sorgfalt der Programmierer, da diese sich vermehrt darauf verlassen, die Tester würden dann die Probleme in ihren Programmen schon finden. Weniger Sorgfalt jedoch bewirkt weniger Qualität und höheren Aufwand. Man beachte ferner, dass beide Qualitätsmaßnahmen einerseits positiv auf den Aufwand wirken (die positive Wirkung ist hier unerwünscht, weil sie eine Erhöhung der Kosten bedeutet), gleichzeitig aber indirekt über die höhere Programmqualität den Aufwand reduzieren. Dementsprechend muss abgeschätzt werden, wie stark welche Wirkungen sind, wenn das Resultat insgesamt den Absichten entsprechen soll.

4.3 Der Projektabschluss

Der Projektabschluss ist fast so wichtig wie der Projektstart. Es geht darum, das entstandene Produkt geordnet in die Phase der Pflege überzuleiten und das entstandene Projektwissen zu sichern. Die während des Projekts entstandenen Dokumente und Messungen werden abgeschlossen. Aus den Messwerten werden interessierende Gesamtgrößen bestimmt, zum Beispiel Gesamtaufwand und totale Durchlaufzeit des Projekts. Eine kurze Projektgeschichte hält SOLL und IST bezüglich Sache, Terminen, Kosten und Personalaufwand fest und berichtet über wichtige Erfahrungen, die in zukünftigen Projekten von Nutzen sein könnten.

Alle für die Pflege der Software, für den Nachweis qualitätsrelevanter Eigenschaften oder für eine spätere Auswertung notwendigen Dokumente und Messungen werden archiviert; der Rest wird vernichtet.

4.4 Software-Risikoführung

Projektrisiken sind Ereignisse, welche den sachlichen oder wirtschaftlichen Erfolg eines Projekts bedrohen. Gut geführte Projekte zeichnen sich dadurch aus, dass in ihnen die Risiken nicht dem Zufall überlassen werden, sondern dass die Projektleiterin oder der Projektleiter eine explizite *Risikoführung* betreibt (Boehm 1989, 1991, Charette 1989). Bild 4.6 zeigt die Aufgaben der Risikoführung.

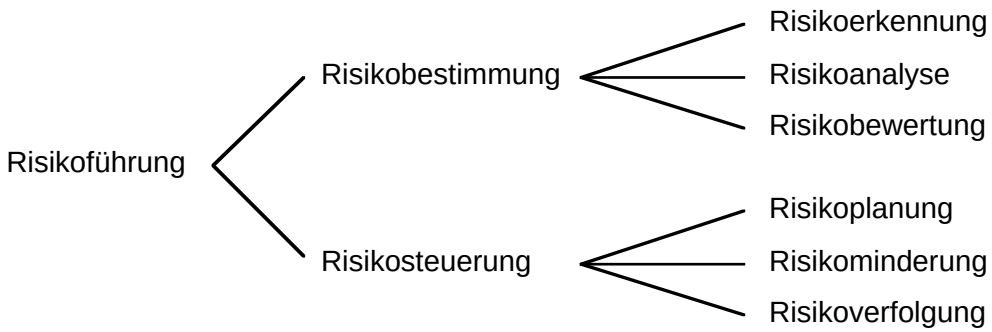


BILD 4.6. Aufgaben der Risikoführung

4.4.1 Risikobestimmung

Als erstes müssen die Risiken eines Projekts *erkannt* werden. Eine Checkliste oder zumindest das Durchgehen der Liste der häufigsten Software-Projektrisiken (Tabelle 4.1) ist dabei hilfreich.

TABELLE 4.1. Die 10 häufigsten Software-Risiken (nach Boehm 1991)

Risiko	**mögliche Maßnahmen**
zuwenig Leute	gute Leute einstellen, vorhandene Leute ausbilden, Motivation und Arbeitsklima fördern, den richtigen Leuten die richtigen Aufgaben geben
unrealistische Kosten- und Terminpläne	sorgfältige Aufwandschätzung, Entwicklung mit Wachstumsmodell, Anforderungen reduzieren, kostenorientierte Entwicklung
falsche Funktionalität	quantifizierte Ziele, sorgfältige Spezifikation, Prototypen, Beteiligung des Auftraggebers
falsche Benutzerschnittstelle	Prototypen, Einbezug der Endbenutzer (oft nicht identisch mit den Auftraggebern!)
Vergoldung (überflüssiger Luxus)	Kosten-Nutzen-Analyse, Setzen von Prioritäten in den Zielen, kostenorientierte Entwicklung
sich ständig verändernde Anforderungen	Setzen von Wichtigkeits-Schwellwerten (unterhalb derer nicht geändert wird), änderungsfreundlicher Entwurf (z.B. durch Information Hiding), Entwicklung mit Wachstumsmodell
Probleme mit zugekauften Komponenten	sorgfältige Auswahl (z.B. mit Benchmarks), Eingangs-Qualitätskontrolle
Probleme mit extern vergebenen Aufträgen	Überprüfung des Auftragnehmers vor Auftragsvergabe, klar formulierte Aufträge, Zwischeninspektionen während der Abwicklung, Abnahmeinspektion, Aufträge mit Erfolgshonorar
Nichterreichen der verlangten Leistungen (z.B. Reaktionszeit)	Abschätzung in Review, Simulationen, Prototypen, Messung und Optimierung
Überforderung der Mitarbeiter in Bezug auf ihr softwaretechnisches Können	Aufgaben-Analyse, Ausbildung, Reduktion der Anforderungen, Entwicklung mit Wachstumsmodell

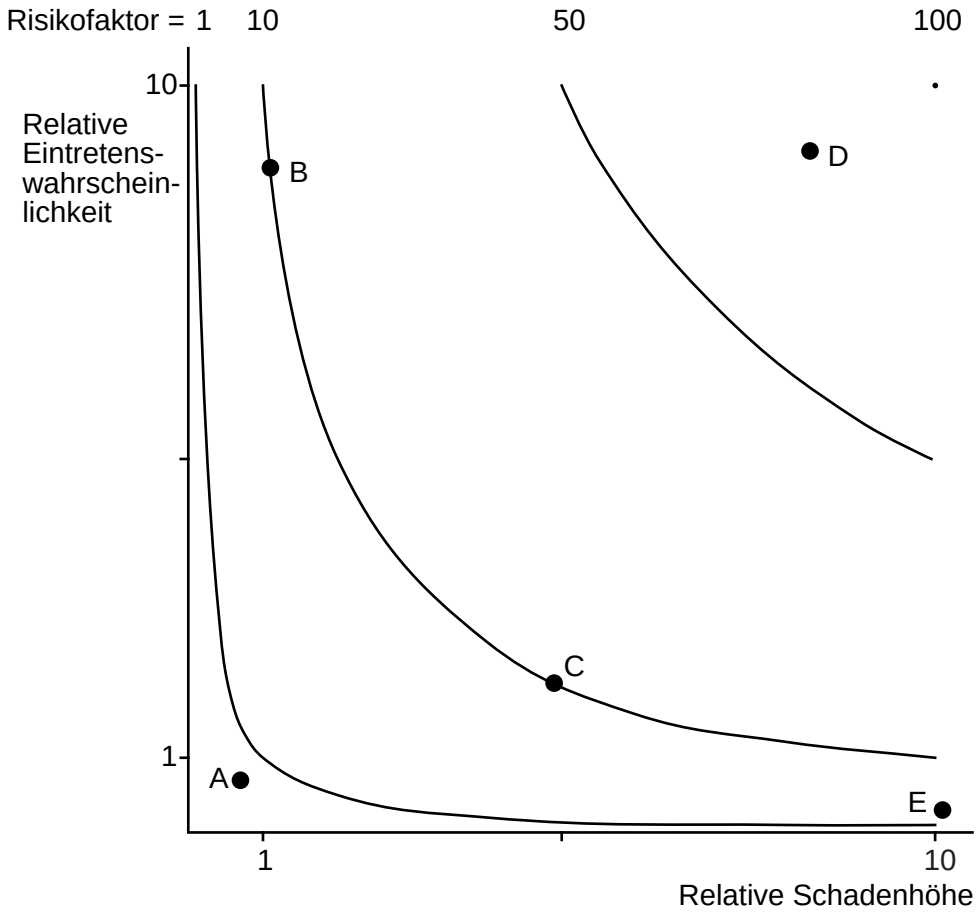
In der *Risikoanalyse* wird die Gefährlichkeit der einzelnen Risiken bestimmt und werden Probleme untersucht, die sich aus dem gleichzeitigen Eintreten mehrerer Risiken ergeben.

Die Gefährlichkeit eines Risikos ist abhängig von zwei Größen: der Eintretenswahrscheinlichkeit und der Schadenhöhe. Es ist daher sinnvoll, Risiken nach ihrem *Risikofaktor* zu beurteilen. Dieser ist das Produkt aus Eintretenswahrscheinlichkeit und Schadenhöhe (Bild 4.7). Üblicherweise werden relative Skalen verwendet, bei einfachen Abschätzungen nur mit den Werten gering, mittel und hoch, bei genauerem Arbeiten mit Werten von 0 bis 10.

Das Arbeiten mit Risikofaktoren hat allerdings einen Schönheitsfehler: Risiken mit sehr hohem Schaden bei gleichzeitig sehr geringer Eintretenswahrscheinlichkeit (sogenannte *Restrisiken*) können einen sehr kleinen Risikofaktor haben. Hier kommen andere Überlegungen zum Zug, zum Beispiel ob die Firma einen solchen Schaden finanziell überlebt oder ob der Schaden sozial noch tolerierbar ist.

Bei Risikokombinationen können die resultierenden Risikofaktoren anhand eines Entscheidungsbaums bestimmt werden.

In der *Risikobewertung* werden die Konsequenzen aus der Risikoanalyse gezogen. Es wird eine Rangliste der Risiken erstellt und es werden Maßnahmen zur Risikominderung geplant und eingeleitet.



- A ist ein harmloses Risiko
- Die Risiken B und C haben den gleichen Risikofaktor und sind somit etwa gleich gefährlich
- D ist ein sehr gefährliches Risiko, welches unbedingt unter Kontrolle gebracht werden muss
- E ist ein sogenanntes Restrisiko, welches anders beurteilt werden muss (siehe Text)

BILD 4.7. Beurteilung von Risiken

4.4.2 Risikosteuerung

Um die erkannten Risiken in den Griff zu bekommen, braucht es für jedes größere Risiko einen *Plan*, wie das Risiko beherrscht werden soll. Grundsätzlich ist bei jedem Risiko zu überlegen
- ob das Risiko vermeidbar ist
- welche Maßnahmen zur Risikominderung getroffen werden sollen
- wie das Risiko im Projekt verfolgt werden soll
- ob das Risiko auf Dritte abgewälzt werden kann.

Tabelle 4.1 zeigt eine Reihe von Maßnahmen zur *Risikominderung*. Sie lassen sich mit folgenden Stichworten zusammenfassen: Informationsbeschaffung, Personalpolitik, Prototypen, Simulationen, Vereinfachung.

Bei der *Risikoverfolgung* geht es darum, die wichtigsten Risiken und ihre Entwicklung während der gesamten Projektabwicklung im Auge zu behalten. Boehm zum Beispiel empfiehlt, in jedem Projekt eine Rangliste der 10 wichtigsten Risiken zu führen. In den in regelmäßigen Abständen stattfindenden Sitzungen der Projektverantwortlichen mit ihren Vorgesetzten wird diese Liste besprochen und aktualisiert. Die dynamische Entwicklung der Rangliste bildet die Grund-

lage, um Maßnahmen zur Risikominderung zu treffen oder getroffene Maßnahmen zu überprüfen und gegebenenfalls anzupassen.

4.4.3 Wirtschaftlichkeit der Risikoführung

Eine sorgfältige Risikoführung kann mit erheblichen Kosten verbunden sein. Fragt man sich nach der Wirtschaftlichkeit dieser Aufwendungen, so ist folgendes zu beachten. Risikoführungskosten amortisieren sich bei jedem Projekt, bei dem die Risikoführung das Eintreten eines größeren Risikos verhindert oder dessen Schaden erheblich mindert. Einen entsprechenden Nachweis zu führen, ist im Einzelfall allerdings oft schwierig. In Projekten, in denen es rückblickend keine Probleme mit Risiken gab, sind die Risikoführungskosten ähnlich zu betrachten wie Versicherungsprämien: Sie sind, in vernünftigem Umfang eingesetzt, im statistischen Mittel wirtschaftlich und sie vermindern drastisch die Wahrscheinlichkeit, dass der Risikofall zur finanziellen Katastrophe wird.

Aufgaben

4.1 Gegeben sei das Problem aus Aufgabe 2.5 (System zur statistischen Auswertung von Betriebsdaten). Zum Umfeld sei folgendes zusätzlich bekannt: Sie verfügen als ProjektleiterIn über eine erfahrene Gruppe von Entwicklerinnen und Entwicklern. Allerdings hat die Informatikerin, welche bei der Entwicklung des Vorgängersystems die Konzepte erstellt und die Hauptarbeit geleistet hatte, vor einem Monat das Unternehmen verlassen. Wegen Sparmaßnahmen in Ihrer Firma hat Ihre Chefin die Stelle bisher nicht wiederbesetzen dürfen. Einen Ihrer Projektmitarbeiter müssen Sie für ein strategisches Projekt an eine andere Abteilung ausleihen. Sie verfügen damit über rund 30% weniger Arbeitskapazität, als Sie für die Abwicklung Ihres Projekts eigentlich brauchen. Eine Analyse ergibt folgende Aufwandschätzung für das Projekt:

- Portierung der vorhandenen Software mit gleichzeitiger Trennung in einen maschinenabhängigen Infrastrukturteil und einen maschinenunabhängigen Applikationsteil: 12 Personenwochen
- Parametrierung des verarbeitbaren Datenvolumens: 3 Personenwochen
- Erstellung der graphischen Auswertungen auf der Basis einer gekauften Graphik-Bibliothek: 12 Personenwochen
- Abnahme und Projektabschluss: 1 Personenwoche.

Zur Problemanalyse, Planung und Erstellung eines Rahmenkonzepts für die Lösung haben Sie bisher zwei Wochen aufgewendet; eine weitere Woche werden Sie noch brauchen. Der vereinbarte Liefertermin ist heute in neun Wochen. Sie können (sich selbst eingerechnet) maximal drei Leute mit voller Arbeitskapazität für das Projekt einsetzen.

Entwerfen Sie einen Aufgaben- und Terminplan für dieses Projekt im Stil von Bild 4.2.

4.2 Begründen Sie, warum das Auftragen von budgetierten und tatsächlichen Kosten über der Zeit ein unbrauchbares Mittel für die Kostenverfolgung in Projekten ist.

4.3 Gegeben sei das Problem aus Aufgabe 4.1; es ist aber 10 Tage früher. Sie haben soeben mit der Problemanalyse begonnen und wollen nun die Risiken des Projekts abschätzen.

a) Welches sind Ihrer Ansicht nach die größten Risiken, die den Erfolg Ihres Projektes gefährden können?

b) Welche Maßnahmen können Sie ergreifen, um diese Risiken zu steuern und/oder zu verkleinern?

Ergänzende und vertiefende Literatur

Es gibt eine Fülle von Literatur über Software-Projektmanagement. Stellvertretend seien hier die folgenden zwei Bücher herausgegriffen.

Frühauf, Ludewig und Sandmayr (1999) geben eine kurz gefasste Einführung in die Probleme der Software-Projektführung.

Weinberg (1992) gehört zur Pflichtlektüre für angehende Software-Manager und -Projektleiter.

Glinz (1999) enthält eine Standortbestimmung sowie weitere Literaturverweise.

Boehm (1991) gibt einen Überblick über die Problematik der Risikoführung. Charette (1989) ist ein Lehrbuch über Risikoführung. Boehm (1989) ist ein Tutorium, das Artikel verschiedener Autoren zur Risikoproblematik enthält.

Zitierte Literatur

Boehm, B.W. (1989). *Software Risk Management*. Los Alamitos, Ca.: IEEE CS Press.

Boehm, B.W. (1991). Software Risk Management: Principles and Practice. *IEEE Software* **8**, 1 (Jan. 1991), 32-41.

Charette, R.N. (1989). *Software Engineering Risk Analysis and Management*. New York, etc.: McGraw-Hill.

Frühauf, K., Ludewig, J., Sandmayr, H. (1999). *Software-Projektmanagement und -Qualitätssicherung*. Dritte, überarbeitete Auflage. Zürich: vdf.

Glinz, M. (Hrsg.) (1999). Software-Projektmanagement. *Informatik/Informatique* **6**, 5 (Okt 1999).

Weinberg, G.M. (1971). *The Psychology of Computer Programming*. New York: Van Nostrand Reinhold.

Weinberg, G.M. (1992). *Quality Software Management: Volume 1, Systems Thinking*. New York: Dorset House.