

## Einführung in die drei Teildisziplinen

---

Version 1.1

### Ziele dieses Kapitels

Nach Abschluss dieses Abschnittes sollten Sie in der Lage sein, ...

- darzustellen, was man unter Software bzw. Software Engineering versteht.
- Motivation und Ziele des Software Engineering zu skizzieren.
- zu erklären, warum der Aspekt der Nutzerfreundlichkeit bei der Entwicklung von Software und Produkten berücksichtigt werden soll.
- zu erklären, was man unter „Usability Engineering“ und „Benutzerfreundlichkeit“, sowie unter einem Projekt bzw. Projektmanagement versteht.
- Erfolgs- und Misserfolgskriterien von Projekten zu benennen.

# Gliederung

Ziele dieses Kapitels.....	I
Gliederung.....	II
1. Software Engineering .....	3
1.1 Was ist Software bzw. Software Engineering? .....	3
1.2 Seit wann gibt es Software Engineering?.....	4
1.3 Ziele des Software Engineering .....	5
2. Usability Engineering.....	7
2.1 Warum Usability Engineering? .....	7
2.2 Was ist Usability Engineering?.....	8
2.3 Faktoren der Benutzerfreundlichkeit .....	9
2.4 Normen und Gesetze .....	10
3. Projektmanagement.....	11
3.1 Warum Projektmanagement?.....	11
3.2 Merkmale von Projekten .....	11
3.3 Wann ist ein Projekt erfolgreich?.....	13
3.4 Was ist Projektmanagement? .....	14
3.5 Hürden und typische Fehler .....	15
4. Zusammenfassung .....	17
5. Literatur und weiterführende Literatur.....	18

# 1. Software Engineering

## 1.1 Was ist Software bzw. Software Engineering?

Software (SW) begegnet uns im Alltag, sobald wir zum Beispiel mit dem Smartphone im Internet surfen oder den Heimcomputer bedienen. Trotz dieser Alltäglichen Präsenz von Software, ist die Definition dieses Begriffes bis heute umstritten. Nach der Definition der IEEE Computer Society umfasst Software demnach alle „**Programme, Abläufe, Regeln, [...] Dokumentationen und Daten, die mit dem Betrieb eines Rechnersystems zu tun haben.**“

*Ludewig & Lichter (2010, S. 34).*

Von der Software wird im Allgemeinen die Hardware (HW) eines Computers unterschieden. Während **Software alle nichtphysischen Bestandteile eines Rechnersystems bezeichnet, sind mit Hardware all seine physischen Bestandteile** gemeint. Software weist einige besondere Merkmale auf, so ist **Software immateriell und bildet in vielen Fällen Vorgänge der analogen Welt/realen Welt ab**. Ebenso wie Hardware wird Software generell als **technisches Produkt** betrachtet, welches von Ingenieuren systematisch entwickelt und durch feststellbare Eigenschaften, wie Funktionalität und Qualität, ausgezeichnet wird.

Auch über die Auslegung des Begriffs „**Software Engineering**“ wird bis heute viel diskutiert. Der IEEE Standard besagt:

“(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).”

*IEEE Std 610.12 (1990).*

Helmut Balzert definiert den Begriff „Software Engineering“ oder „Softwaretechnik“ in seinem Nachschlagewerk wie folgt:

„Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Techniken, Verfahren und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen. Zielorientiert bedeutet die Berücksichtigung von Kosten, Zeit und Qualität.“

*Balzert (2000, S. 36).*

Kurz gesagt: **Software Engineering bezeichnet also jede Aktivität, die die Erstellung und/oder Veränderung einer Software umfasst.** Fast immer werden mit der Software Ziele verfolgt, die über das Softwareprodukt hinausgehen.

## 1.2 Seit wann gibt es Software Engineering?

George Moore, Chef der Firma Intel, bemerkte bereits in den 60er Jahren, dass die Kerngrößen der Hardware, wie z. B. Prozessorleistung und Speicherkapazität, exponentiell wachsen. Er führte zudem auf, dass sie sich sogar alle 18 Monate verdoppeln. Dieser Sachverhalt wird seitdem als *Moore's Law* gelehrt. Als Konsequenz daraus, wurden im Laufe der Jahre die Hardware immer kleiner und billiger, aber auch schneller und zuverlässiger. Bei der Software verhielt sich dies nicht so: **Software wurde weder billiger noch zuverlässiger. Der Hauptgrund hierfür liegt in der zunehmenden Komplexität von Software und der aufwändigen Pflege.** Bemerkenswert ist zudem, dass sich die Kosten für eine Zeile Code im Laufe der Jahrzehnte kaum verändert haben. Allerdings leistet diese Zeile Code in höheren Programmiersprachen viel mehr als in primitiven Programmiersprachen, welche früher weitgehend eingesetzt wurden.

In den späten 70er Jahren wurde nach und nach deutlich, dass sich manche Softwareprojekte selbst mit immensem Aufwand nicht zufrieden stellend abschließen ließen und entweder zu spät oder mit hohen Kostenüberschreitungen beendet wurden. Nicht alles war machbar und umsetzbar, so wie es die Entwickler erwarteten. Diese Situation nahm die Formen einer **Softwarekrise** an, welche Auslöser für die gezielte Entwicklung von Software waren. Die Krise wurde durch den drastischen Verfall der Hardwarepreise und dem gleichzeitig enormen Anstieg der Softwarekosten ausgelöst.

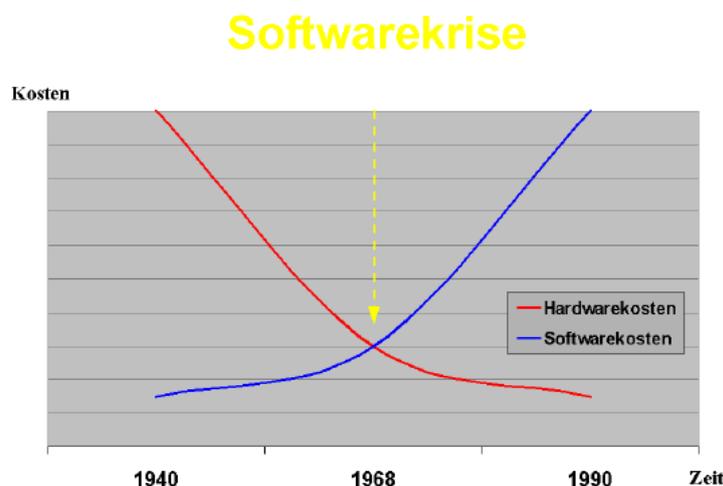


Abbildung 1: Softwarekrise.

Internetpublikation unter: [http://images.slideplayer.org/1/208672/slides/slide\\_19.jpg](http://images.slideplayer.org/1/208672/slides/slide_19.jpg) (Stand: 10.06.2016)

## Einführung in die drei Teildisziplinen

### 1. Software Engineering

F. L. Bauer äußerte hierzu eine passende Aussage:

“The whole trouble comes from the fact that there is so much tinkering with software. It is not made in clean fabrication process, which it could be. What we need is software engineering.”

*Zitat von Friedrich L. Bauer (1968)*

Nach diesem Ereignis entwickelte sich eine Eigendynamik in den 70er Jahren zum Thema Software Engineering.

*Vgl. Ludewig & Lichter (2010, S. 34ff).*

### 1.3 Ziele des Software Engineering

Die **Kosten** für Software steigen stetig an. Gründe hierfür sind die wachsende Komplexität von Software, neue Anforderungen und Änderungen etc. Deshalb ist es wirtschaftlich sinnvoll, Kosten und Qualität der Softwareentwicklung besonders zu berücksichtigen. Vorab muss man sich die Frage stellen: Soll die benötigte Software gekauft oder neu entwickelt werden?

Das Wort „kaufen“ ist in diesem Sinn eigentlich mit „wiederverwenden“ gleichgestellt. Meist endet diese Fragestellung in einer *Neuentwicklung*, da vorhandene Software nicht verändert werden sollte. Das wichtigste Argument einer Neuentwicklung ist die Flexibilität und die präzise Erfüllung aller Anforderungen des Anwenders. Die Entwicklung einer eigenen Software birgt jedoch auch viele Risiken und versteckte Kosten.

Hier spielt speziell der Faktor **Zeit** eine besondere Rolle, denn eine frühzeitige Erkennung und Behebung von Fehlern in IT-Projekten ist von großer Bedeutung! Je später Fehler entdeckt werden, desto höher die Kosten für die Beseitigung des Fehlers!

Auch die **Qualität** ist eng mit den Kosten einer Software verknüpft. Die Qualität einer Software gehört zu den wichtigsten Zielen in einem IT-Projekt.

Die DIN 55350-11:1995-08 definiert den Begriff „Qualität“ wie folgt:

„Gesamtheit von Eigenschaften und Merkmalen eines Produktes oder einer Tätigkeit, die sich auf die Eignung zur Erfüllung gegebener Erfordernisse beziehen.“

*Deutsches Institut für Normung (1995).*

## Einführung in die drei Teildisziplinen

### 1. Software Engineering

Eine Software hat alle geforderten Qualitätsmerkmale, wenn sie in ihrer Funktionalität korrekt ist und in diesem Zusammenhang alle geforderten Eigenschaften besitzt. Zu diesen zählen die geforderte Wartbarkeit, Effizienz und Verfügbarkeit zum geforderten Termin.

Nun stellt man sich natürlich auch die Frage, weshalb es in IT-Projekten häufig zu Überschreitungen der geplanten Kosten und Termine kommt. Abbildung 2 zeigt Gründe auf, wieso Projekte letztendlich scheitern:

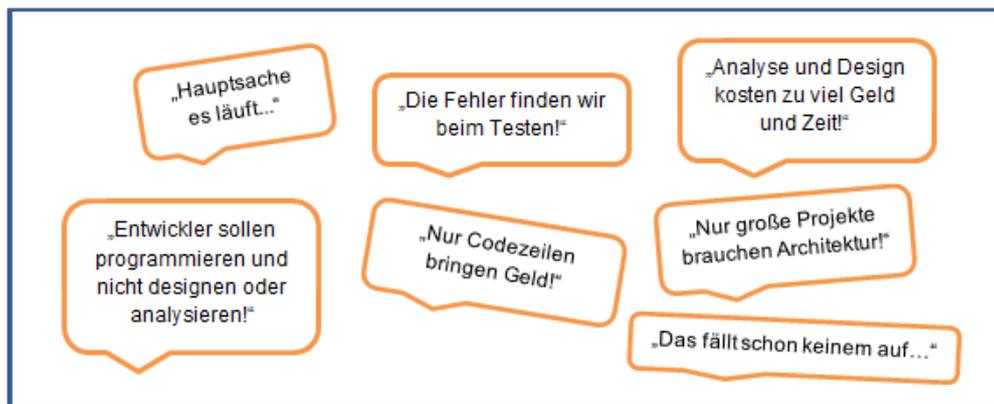


Abbildung 2: Gründe für gescheiterte Softwareprojekte.

Offensichtlich liegt der **Hauptgrund für das Scheitern eines IT-Projekts an der fehlenden oder fehlerhaften Planung und Konzeption, sowohl bei der Entwicklung und insbesondere bei der Wartung von IT-Lösungen.**

## 2. Usability Engineering

In diesem Kapitel lernen Sie einen weiteren Bereich der drei Teildisziplinen dieses Kurses kennen, den Bereich Usability Engineering.

### 2.1 Warum Usability Engineering?

**Usability steht für die Benutzerfreundlichkeit und Gebrauchstauglichkeit eines Produkts** und bleibt im besten Fall von der Nutzerin unbemerkt. Eine nutzerfreundliche Software macht den Anwendern Spaß, gesuchte Elemente werden schnell gefunden und die Bedienung des Systems ist angenehm.

Durch die benutzerfreundliche Gestaltung von Software wird den Nutzern die Arbeit am System erleichtert. Zusätzlich haben Usability-Aspekte Einfluss auf die Produktentwicklungskosten, die Umsatzsteigerung, Support- und Schulungskosten und den Produktlebenszyklus.

**Merkmale für schlechte Usability** wären zum Beispiel:

- Mitarbeiter arbeiten mit dem System nicht so schnell wie erhofft.
- Die Einarbeitung und die Schulung von Benutzerinnen nimmt viel Zeit in Anspruch.
- Die Qualität der geleisteten Arbeit sinkt merklich.
- Die Motivation der Mitarbeiter fällt ab, sie minimieren die Tätigkeit am System.
- Prozessvorgaben werden umgangen und Sicherheitsmaßnahmen ignoriert.
- Es gibt immer wieder Fälle, in welchen „Benutzerfehler“ die Ursache von Schäden (Unfälle, Datenverluste, kommerzielle Schäden) sind.

*Vgl. Richter & Flückiger (2010, S. 7)*



Das Video [“Beispiel von schlechter Usability”](#) zeigt - auf humorvolle Art - mögliche Auswirkungen von unzureichender Usability.

## 2.2 Was ist Usability Engineering?

Ganz allgemein kann man sagen, dass Usability ein Qualitätsmerkmal eines Produkts ist. Sie wird zusammenfassend wie folgt definiert:

„Die Gebrauchstauglichkeit (Usability) ist das Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch **bestimmte Benutzer** in einem **bestimmten Nutzungskontext** genutzt werden kann, um **festgelegte Ziele effektiv, effizient und zufriedenstellend** zu erreichen.“

*Deutsches Institut für Normung (2011), S. 7*

Effektivität bezieht sich hier auf den Grad der Zielerreichung, Effizienz auf den Aufwand, um diesen Effekt zu erreichen. Dieser kann sich beispielsweise auf erforderliche Zeit oder Anstrengung beziehen.

Usability Engineering bezeichnet somit den **Prozess, d.h. ein systematisches, ingenieurmäßig betriebenes Vorgehen**, das bei der Erstellung von interaktiven Systemen eingesetzt wird, um auf eine möglichst hohe Gebrauchstauglichkeit als Merkmal der Interaktion Nutzer-Produkt hinzuarbeiten.

Der **Benutzer (Endanwender) ist das zentrale Element** im Usability Engineering. Dieser darf nicht mit dem Experten/ den Entwicklerinnen der Software gleichgesetzt werden, die über andere Herangehensweisen und anderes Hintergrundwissen verfügt. Um eine Software benutzerfreundlich gestalten zu können, muss die **Sichtweise des Endanwenders** in die Gestaltung und den Entwicklungsprozess des Produkts miteinbezogen werden.

*Vgl. Richter & Flückiger (2010), S. 2.*



Siehe hierzu auch das Video „[Der Benutzer](#)“.

#### 2.3 Faktoren der Benutzerfreundlichkeit

Eine Software muss folgende Eigenschaften aufweisen, um als nutzerfreundlich bezeichnet werden zu können:

**Nützlichkeit:** Die Software muss mit all ihren Funktionen die **Bedürfnisse der Benutzerinnen befriedigen**. Um die Bedürfnisse der Benutzer beurteilen zu können, werden die Anforderungen in einer Analyse erhoben.

**Erlernbarkeit:** Je nach Anwendungsfall sollte die **Bedienung einer Software für die Benutzerinnen möglichst schnell und einfach zu erlernen sein**. Für spezielle Produkte, bei denen beispielsweise eine effiziente Bedienung im Vordergrund steht, kann eine umfangreiche Lernphase jedoch auch vorteilhaft sein, um diese Effizienz zu gewährleisten (z. B. Zehn-Finger-Schreiben).

**Effizienz:** Aufgaben sollen vom Benutzer **schnell und ohne Fehler** erledigt werden können. Bei der Softwareentwicklung werden häufig Prototypen als Hilfsmittel herangezogen, um den Faktor Effizienz bestmöglich optimieren zu können.

**Zufriedenheit:** Die Software muss den **Erwartungen der Benutzerinnen entsprechen**. Dabei können die Erwartungen der Benutzer in drei Bereiche geteilt werden: Erwartungen an den Kauf und die Informationen vor dem Einsatz der Software, an die Anwendung der Software und an den Support nach Installation des Produkts.

*Vgl. Groner et. al. (2008), S. 430ff.*



Im angefügten Video „[Faktoren der Benutzerfreundlichkeit](#)“ werden Ihnen die Faktoren der Benutzerfreundlichkeit noch einmal genauer erklärt.

## 2.4 Normen und Gesetze

Bei der benutzerzentrierten Gestaltung der Interaktion zwischen Menschen und Computern müssen einige Gesetze, Verordnungen und Normen berücksichtigt werden.

Allgemein gelten Gesetze, die in jedem Unternehmen eingehalten werden müssen. Dies wären beispielsweise das Betriebsverfassungsgesetz, das Arbeitsschutzgesetz, die Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten oder das Behindertengleichstellungsgesetz.

Zusätzlich gibt es eine Vielzahl von Normen und Richtlinien, die bei der Gestaltung von benutzerfreundlicher Software wichtige Hinweise liefern können.

Dabei spielt im Usability-Umfeld die **DIN EN ISO 9241 Normenreihe** mit dem Titel „Ergonomie der Mensch-Computer-Interaktion“ eine wichtige Rolle. Sie beschreibt **Mindestanforderungen an die Konzeption, Gestaltung und Bewertung von Software, Hardware und Arbeitsumgebung und gilt als internationaler Standard.**

Insbesondere drei Teile der DIN EN ISO 9241 werden häufig zur benutzerfreundlichen Gestaltung von Software herangezogen:

- DIN EN ISO 9241-11
- DIN EN ISO 9241-110
- DIN EN ISO 9241-210

Diese Norm wird durch die DIN EN ISO 14915 „Software-Ergonomie für Multimedia-Benutzerschnittstellen“ ergänzt, welche in drei Teile gegliedert ist:

- Teil 1: Gestaltungsgrundsätze und Rahmenbedingungen
- Teil 2: Multimedia-Navigation und Steuerung
- Teil 3: Auswahl und Kombination relevanter Medien

## 3. Projektmanagement

Die Arbeit an und in Projekten in und für Unternehmen/Organisationen ist heute eine häufige Aufgabe vieler Arbeitnehmer, soll z.B. in einem Unternehmen eine neue Software entwickelt und eingeführt werden, kann dies als Projekt und mithilfe von Methoden des Projektmanagements umgesetzt werden. Nachfolgend werden einige Grundlagen zum Thema Projektmanagement vorgestellt.

### 3.1 Warum Projektmanagement?

Veränderungsgeschwindigkeit und Komplexität haben in den letzten Jahren in allen Bereichen der Wirtschaft sehr zugenommen. **Für eine zunehmende Zahl von neuartigen Problemstellungen bzw. Aufgaben gibt es keine Verantwortlichkeiten in der hierarchischen Linienorganisation eines Unternehmens.** Die erforderliche flexible, bereichsübergreifende Zusammenarbeit und die Notwendigkeit rascher Entscheidungen sind in der Linienorganisation schwer umsetzbar.

Deshalb sind **neue Organisationsformen und Strukturen notwendig**, um effizientere Führungs- und Kommunikationswege zu ermöglichen. Diese sind in der Lage den wechselnden Anforderungen der jeweiligen Aufgabenstellung gerecht zu werden.

Besonders in der Raumfahrt und im Anlagebau wurde Projektmanagement schon in den 50er Jahren eingeführt. Für Projekte in diesen Branchen wurden spezielle Methoden entwickelt, welche Sie in späteren Kapiteln dieses Kurses noch kennenlernen werden. Nicht nur bei technischen Aufgabenstellungen werden diese Methoden zur Lösung komplexer Aufgaben angewandt. In Problem- und Krisensituationen werden diese Instrumente in allen Funktionen des Managements, wie z. B. Marketing, Personalwesen und Finanzen eines Unternehmens sowie in öffentlichen Verwaltungen, eingesetzt.

*Vgl. Kuster (2011, S.3).*

### 3.2 Merkmale von Projekten

Ähnlich wie in den Bereichen Software Engineering und Usability Engineering konnte sich bislang keine allgemeingültige Definition des Begriffs „Projekt“ etablieren.

Die DIN 69901-5 definiert den Begriff „**Projekt**“ wie folgt:

„Ein Projekt ist ein Vorhaben, das im Wesentlichen durch **Einmaligkeit der Bedingungen in ihrer Gesamtheit** gekennzeichnet ist. Es zeichnet sich aus durch ...

... eine Zielvorgabe,

... eine zeitliche, finanzielle, personelle oder andere Begrenzung

... und eine projektspezifische Organisation“

*Deutsches Institut für Normung (2009).*

Zwei grundsätzliche Charakteristika von Projekten sind:

- 1) Jedes Projekt ist **zeitlich begrenzt** durch einen Projektstart und ein Projektende. Dabei muss das Projektende klar definiert sein, sodass alle Projektbeteiligten eine Vorstellung davon haben, wann das Projekt abgeschlossen ist.
- 2) Jedes Projekt erzeugt ein **einmaliges Ergebnis**. Zwar können einige Projektliefergegenstände (erarbeitete Projektergebnisse, z.B. ein Lastenheft, ein Marketingkonzept) und/oder –aktivitäten sich wiederholende Elemente enthalten, der einmalige Charakter der Projektarbeit bleibt dennoch bewahrt. (Bsp: der Bau von Bürogebäuden erfolgt durch dasselbe Team mit gleichen Materialien, das Bauprojekt ist dennoch einmalig)

Kurz: **Ein Projekt ist ein zeitlich begrenztes Vorhaben und hat immer das Ziel, ein einmaliges Produkt oder eine einmalige Dienstleistung zu schaffen.** Das Ergebnis dabei kann materiell oder immateriell sein.

*Vgl. Projekt Management Institute (PMI®). (2013, S.3)*

### Unterschied Projekt und Linie

Ein Projekt wird im Projektmanagement von der sog. Linie unterschieden. Zur Letzteren zählen laufende Prozesse wie beispielsweise die tägliche Kundenberatung einer Bank oder Fertigung eines Produkts. **Diese Prozesse produzieren ähnliche/identische Produkte/Ergebnisse** (z.B. das Auto einer bestimmten Baureihe), **finden permanent statt** und haben damit kein genaues Ende definiert. Aufgrund dessen können in der Linie Organisationseinheiten mit eindeutigen, permanenten Verantwortlichkeiten für diese Aufgaben gebildet werden. **Der Betrieb ist also fortlaufend und wiederholt, während Projekte zeitlich begrenzt und einmalig sind.**

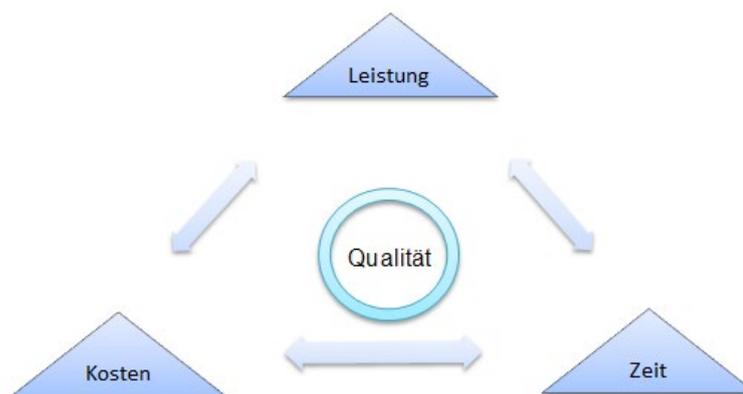
*Vgl. Verzuh (2012, S. 17ff).*

#### 3.3 Wann ist ein Projekt erfolgreich?

Die drei wichtigsten Parameter eines Projektes werden im sogenannten „**Magischen Dreieck**“ dargestellt: **Leistung, Zeit und Kosten**. Dabei ist Leistung hinsichtlich Quantität und Qualität zu sehen. Obwohl der Faktor Qualität ein Merkmal der Leistung ist, wird er im Magischen Dreieck oftmals gesondert beigefügt.

Die drei Parameter haben im gesamten Projektverlauf **wechselseitige, konkurrierende Beziehungen**, deshalb spricht man vom „Magischen Dreieck“.

Basis für den Erfolg eines Projekts ist die Einhaltung der vertraglich definierten Parameter des Magischen Dreiecks. Bei der Definition eines Projektes ist es eine sehr anspruchsvolle Aufgabe diese aufeinander abgestimmt festzulegen. Das Verhältnis und die Priorität der einzelnen Parameter zueinander sind projektbezogen und dementsprechend nur situativ bestimmbar. Wird eine dieser drei Variablen geändert, so müssen alle weiteren Variablen auch angepasst werden.



**Abbildung 3. Das Magische Dreieck im Projektmanagement.**  
Vgl. GPM Deutsche Gesellschaft für Projektmanagement e. V. (2012, S. 57)

**Zeit:** Das Produkt wird zeitlich nach Plan geliefert. Einige Projekte sind im Wesentlichen wertlos, wenn sie nicht pünktlich abgeschlossen werden.

**Kosten:** Das Projekt entspricht den prognostizierten Kostenschätzungen.

**Leistung/Qualität:** Das Projektergebnis muss die Kundenerwartungen erfüllen.

#### Voraussetzung für den Erfolg eines Projektes:

Die Erwartungen der relevanten Interessengruppen an die Ergebnisse des Projekts müssen bekannt und berücksichtigt werden. Sollten Zielsituationen im Konflikt zu anderen Zielsituationen stehen, muss ein entsprechender Kompromiss gefunden werden.

**Ein Projekt ist nur dann erfolgreich, wenn alle Ziele, welche vom Auftraggeber, den Beteiligten und Betroffenen als relevant eingestuft worden sind, am Projektende erreicht wurden.**

*Vgl. GPM Deutsche Gesellschaft für Projektmanagement e. V. (2012, S. 57ff)*

### 3.4 Was ist Projektmanagement?

Die DIN 69901-5 definiert Projektmanagement wie folgt:

**„Gesamtheit von Führungsaufgaben, -organisation, -techniken und –mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten.“**

*Deutsches Institut für Normung (2009).*

Demnach umfasst das Leiten eines Projekts folgende Aufgaben:

- Anforderungen ermitteln
- Unterschiedliche Bedürfnisse, Bedenken und Erwartungen der Projektinteressenten bei der Planung und Ausführung des Projekts berücksichtigen
- Einrichtung, Pflege und Ausführung von aktiver, effektiver und von Zusammenarbeit geprägter Kommunikation zwischen Projektinteressentinnen
- Management der Projektinteressenten dahingehend, dass Anforderungen an das Projekt erfüllt und die Projektliefergegenstände geschaffen werden
- Ausgleichen der konkurrierenden Projekteinschränkungen:
  - o Inhalt und Umfang
  - o Qualität
  - o Terminplanung

### 3. Projektmanagement

- Budget
- Ressourcen
- Risiken

*Vgl. Projekt Management Institute (PMI®). (2013, S. 6)*

Bei Änderung eines Faktors wird mindestens ein weiterer Faktor beeinflusst. Wenn bspw. der Terminplan gekürzt wird, muss häufig auch das Budget angepasst werden. Eine Änderung der Projektanforderungen oder -ziele kann zusätzliche Risiken hervorrufen. Um ein Projekt erfolgreich zum Abschluss zu bringen, ist es demnach erforderlich, die eingetretene Situation erfolgreich einzuschätzen, sie aktiv zu gestalten und eine gezielte Kommunikation mit den Projektinteressenten zu pflegen. Ein Projektleiter muss stets versuchen, den Projektablauf aktiv zu gestalten. Dies ist die wesentliche Voraussetzung für eine effiziente Projektarbeit. Gerät er in eine Situation, in der er reagieren muss, sind Planbarkeit und Steuerbarkeit der Projektarbeit eingeschränkt.

*Vgl. GPM Deutsche Gesellschaft für Projektmanagement e. V. (2012, S. 29ff)*

### 3.5 Hürden und typische Fehler

Aus zahlreichen Studien wurde ersichtlich, dass generell der Faktor Mensch über den Erfolg eines Projektes entscheidet. Er ist für den erfolgreichen Abschluss oder das Scheitern eines Projekts verantwortlich. Somit scheitern Projekte an Menschen und können auch nur durch Menschen zum Erfolg geführt werden.

Es gilt fünf wesentliche **Hürden im Projektmanagement** zu meistern:

- 1) Einigkeit über die Ziele des Projekts zwischen dem Projektteam, den Kunden und dem Management zu erlangen.
- 2) Ein Plan, der den gesamten Projektablauf und klare Verantwortlichkeiten aufzeigt und der verwendet werden kann, um Fortschritte zu messen, festzulegen.
- 3) Eine konstante, effektive Kommunikation zwischen allen Projektbeteiligten zu gewährleisten und zu fördern.
- 4) Einen kontrollierten Projektumfang festzulegen.
- 5) Unterstützung von Seiten des Managements zu gewährleisten.

*Vgl. Verzuh (2012, S. 7ff).*

## Einführung in die drei Teildisziplinen

### 3. Projektmanagement

#### Misserfolgskfaktoren in Projekten durch **typische Fehler der Unternehmensleitung:**

- Unterstützung durch Unternehmensleitung zu gering
- Verzögerung von Entscheidungen
- Befugnisse der Projektleitung zu gering
- Bestellung von inkompetenten Projektleitern

#### Misserfolgskfaktoren in Projekten durch **typische Fehler der Projektleitung:**

- Ziele sind unklar oder zu anspruchsvoll definiert
- Fehler aus alten Projekten werden wiederholt
- Konflikte im Team werden ignoriert und ausgesessen
- Termine werden ohne die Beteiligten geplant

#### Misserfolgskfaktoren in Projekten durch **typische Fehler der Projektmitarbeiter:**

- Es wird zu optimistisch über den Projektverlauf berichtet
- Teammitglieder möchten nur eigene Karrieremöglichkeiten aufbessern
- Eigene Schwächen und Fehler werden verdeckt

#### Misserfolgskfaktoren in Projekten durch **typische Fehler der Auftraggeber/Kunden:**

- Kunde übermittelt seine Ziele und Probleme nicht verständlich
- Durch verdeckte und veränderte Ansprüche seitens des Kunden wird das Projektteam beeinflusst

*Vgl. GPM Deutsche Gesellschaft für Projektmanagement e. V. (2012, S. 59ff)*

## 4. Zusammenfassung

Software begegnet uns heute überall. Software umfasst alle Programme, Abläufe, Regeln, [...] Dokumentationen und Daten, die mit dem Betrieb eines Rechnersystems zu tun haben. Während die Hardware im Laufe der Zeit immer kostengünstiger wurde, stiegen die Kosten für Software teilweise erheblich an. Die ingenieurmäßige Entwicklung von Software - das Software Engineering - wurde daher immer wichtiger. Unter Software Engineering versteht man die zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Techniken, Verfahren und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen. Zielorientiert bedeutet die Berücksichtigung von Kosten, Zeit und Qualität.

Die Gebrauchstauglichkeit von Software und Produkten allgemein ist ein wesentliches Kriterium für deren Erfolg. Die Gebrauchstauglichkeit (Usability) ist das Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um festgelegte Ziele effektiv, effizient und zufriedenstellend zu erreichen. Usability Engineering bezeichnet den Prozess, d.h. ein systematisches, ingenieurmäßig betriebenes Vorgehen, das bei der Erstellung von interaktiven Systemen eingesetzt wird, um auf eine möglichst hohe Benutzerfreundlichkeit als Produktmerkmal hinzuarbeiten. Gegenstand dieses Prozesses ist die Untersuchung und Gestaltung der Interaktion zwischen den Faktoren „Benutzer“, „Aufgaben“, „System“ und „Umgebung“. Zum Einsatz kommt dabei eine Vielzahl an Methoden. Eine Software muss folgende Eigenschaften vorweisen, um als nutzerfreundlich zu gelten: Nützlichkeit, Erlernbarkeit, Effizienz und Zufriedenheit.

Aufgrund der zunehmenden Veränderungsgeschwindigkeit und Komplexität sind Organisationen zunehmend mit Aufgaben konfrontiert, für die es in der klassischen Linienorganisation keine Verantwortlichkeiten gibt. Diese Aufgaben werden von Projekten bearbeitet. Ein Projekt ist ein Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist. Es zeichnet sich aus durch eine Zielvorgabe, eine zeitliche, finanzielle, personelle oder andere Begrenzung sowie eine projektspezifische Organisation. Projekte sind also zeitlich begrenzt und erzeugen ein einmaliges Ergebnis. Unter Projektmanagement versteht man die Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten. Ein Projekt/Projektmanagement ist dann erfolgreich, wenn das Produkt mit der geforderten Leistung/Qualität in der vorgegebenen Zeit zu den geplanten Kosten erstellt wird.

## 5. Literatur und weiterführende Literatur

Bauer, F.L. (1993). Software Engineering – wie es begann. In: Informatik Spektrum, Band 16, Heft 5, Berlin: Springer Verlag.

Deutsches Institut für Normung (1995). Begriffe zu Qualitätsmanagement und Statistik - Teil 11: Begriffe des Qualitätsmanagements (DIN 55350-11:1995-08). Berlin: Beuth.

Deutsches Institut für Normung (2009). Projektmanagement; Projektmanagementsysteme (DIN 69901). Berlin: Beuth.

Deutsches Institut für Normung (2011). Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010); Deutsche Fassung EN ISO 9241-210:2010. Berlin: Beuth.

GPM Deutsche Gesellschaft für Projektmanagement e. V. & Gessler, M. (2012). Kompetenzbasiertes Projektmanagement (PM3), Band 2, 5. Auflage, Nürnberg.

Groner, R. & Raess, S. & Sury, P. (2008). Usability: Systematische Gestaltung und Optimierung von Benutzerschnittstellen. In: Batinic, B. & Appel, M.: Medienpsychologie. Berlin, Heidelberg: Springer Verlag.

Kuster, J. & Huber, E. & Lippmann, R. & Schmid, A. & Schneider, E. & Witschi, U. & Wüst, R. (2011). Handbuch Projektmanagement. Berlin, Heidelberg: Springer Verlag.

Ludewig, J. & Lichter, H. (2010). Software Engineering: Grundlagen, Menschen, Prozesse, Techniken. Berlin, Heidelberg: dpunkt.Verlag.

Projekt Management Institute (PMI®). (2013). A Guide To The Project Management Body of Knowledge (PMBOK® Guide). USA : Project Management Institute (PMI®).

Richter, M. & Flückiger, M.D. (2010). Usability Engineering kompakt. Benutzbare Software gezielt entwickeln. Heidelberg: Spektrum.

Verzuh, E. (2012). The Fast Forward MBA in Project Management. New Jersey: John Wiley & Sons.

Wandke, H. (2004). Usability-Testing. In: Mangold, R., Vorderer, P. & Bente, G. (Hrsg.): Lehrbuch der Medienpsychologie. Göttingen: Hogrefe, 325-354.