

Softwareentwicklung

Mike Scherfner

Allgemeines zur Veranstaltung

Format:

- 9.00 – 10.00
- 10.15 – 12.00
- 13.15 – 15.00

Erster Kontakt

Es geht wesentlich um:

Die Herstellung eines Softwareproduktes: In welchen Phasen entsteht typischerweise ein Computerprogramme (Spiel)? Was kennzeichnet die Phasen und wie hängen diese zusammen?

Methoden der Entwicklung: Welche Hilfsmittel gibt es zur Softwareentwicklung?

Das Projektgeschehen: Was ist ein Projekt? Wie läuft ein Projekt in der IT ab? Welche Aspekte sind bei der Gestaltung zu berücksichtigen?

Erster Kontakt

Die Begriffe Softwaretechnik und Softwareentwicklung werden häufig synonym verwendet.

Notwendigkeit der Disziplin: Aufgrund des hohen Aufwandes zur Erstellung und Wartung komplexer Software erfolgt deren Entwicklung anhand eines strukturierten Planes. Dieser Plan – das sog. Vorgehensmodell – unterteilt den Entwicklungsprozess in überschaubare, zeitlich und inhaltlich begrenzte Phasen.

Erster Kontakt

Die Software wird somit Schritt für Schritt fertiggestellt.

Die Softwaretechnik (engl. Software Engineering) beschäftigt sich mit der Entwicklung von Software, der Organisation und Modellierung der zugehörigen Datenstrukturen und dem Betrieb von Softwaresystemen.

Eine Definition von Helmut Balzert:

Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen.

Erster Kontakt

Die Softwareentwicklung beinhaltet den gesamten Prozess von der Identifizierung eines Bedarfs bis hin zur Inbetriebnahme einer konkreten Erfüllung des Bedarfs (zum Teil auch darüber hinaus).

Aufgabe

Was bedeutet das in Bezug auf Game Development?

SWEBOK

Wichtiges Grundlagenwerk:

Guide to the Software Engineering Body of Knowledge (SWEBOK); ein Dokument der IEEE Computer Society.

Es strukturiert das gesammelte Wissen (englisch body of knowledge) auf dem Gebiet der Softwaretechnik und stellt es der Allgemeinheit zur Verfügung.

SWEBOK (Kernpunkte)

Software requirements: Anforderungsanalyse

Software design: Softwareentwurf

Software construction: Programmierung

Software testing: Softwaretest

Software maintenance: Softwarewartung

Software engineering management: Projektmanagement

Software engineering process: Vorgehensmodell

Software engineering tools and methods: Entwicklungswerkzeuge und -methoden

Software quality: Softwarequalität

Vorgehensmodell

Ein Vorgehensmodell ist wie folgt gekennzeichnet:

1. Ein Vorgehensmodell legt fest, wie Projekte gleicher Art ablaufen.
2. Ein Vorgehensmodell benennt die an den Projekten Beteiligten und beschreibt ihre Aufgaben.
3. Ein Vorgehensmodell stellt Methoden zur Verfügung, die bei der Bewältigung der Aufgaben benutzt werden.

Vorgehensmodell

30 % schneller Fassade dämmen. So easy geht's:



1. Vorbereiten
Easy Kle beschichtete Dämmplatten. Ver-
schlussklappe nach oben, auf
die Schraubnähte aufzuschrauben
und los geht's!

2. Auftragen
Unterlauf und mittig einen ca. 6 cm
starken Strang Klebeschichtung auf
die Dämmplatte aufzutragen.
ca. 6 % der Dämmplatte muss mit ca.
60 % mit Klebeschichtung bedeckt sein.
Zusätzlich 2 Fixierpunkte mit einer
Fixierheber (je ca. 10 g) in den oberen
Ecken der Dämmplatte auftragen.

3. Anbringen
Die Dämmplatten leicht
Fassade drücken und mit
einer Wasserdruckrolle (ca.
10 kg/cm²) ins Lot bringen.

So einfach gehts:

- ▶ die Bearbeitung und das Angebot eines Onlinekredit ist grundsätzlich kostenlos
- ▶ Kredit Antrag online ausfüllen und unterschrieben zurücksenden
- ▶ das Kreditangebot erhalten Sie umgehend (meistens nach einem Tag)
- ▶ Sie entscheiden nun selber, nach Rücksendung der Unterlagen kann bereits ausgezahlt werden
- ▶ die komplette Abwicklung erfolgt auf dem Postweg
- ▶ zum [Kreditantrag](#) für Sofortkredit

The "PLAN" Phase:
■ Strategic & Management Consulting
■ Organizational Consulting
■ Business Consulting (Business Processes)
■ IT Consulting (IT Architectures, Technologies, IT Processes)
■ Definition, Planning & Design

The "BUILD" Phase:
■ Systems Integration (IT Infrastructure, Hard- and Software, WebServ

Vorgehensmodell

Zum ersten Punkt: Die meisten Vorgehensmodelle beschreiben den Ablauf von Projekten mit Hilfe von Phasen. Phasen fassen bestimmte Tätigkeiten eines Gesamtprozesses sinnvoll zusammen.

Noch ein weiterer Aspekt steckt in dem ersten Punkt der Definition: Vorgehensmodelle beziehen sich immer auf Projekte gleicher Art.

Vorgehensmodell

Zum zweiten Punkt: Hier ist die verallgemeinerte Benennung und Aufgabenbeschreibung gemeint. Ein Vorgehensmodell beinhaltet also die Benennung und Definition von Rollen.

Vorgehensmodell

Zum dritten Punkt: Ein Vorgehensmodell überlässt es den Beteiligten nicht, wie sie ihre Aufgaben bearbeiten. Es beinhaltet vielmehr auch Methoden, die bei der Erledigung der Aufgaben angewendet werden.

Vorgehensmodell

Motivation für Vorgehensmodelle:

1. divide et impera – teile und herrsche

Das Erkennen und/oder Definieren von Strukturen zerlegt ein kompliziertes Problem in kleinere, beherrschbare Probleme. Durch die Festlegung von Phasen und Rollen kann eine große Aufgabe besser oder sogar überhaupt erst gelöst werden.

2. Ein Vorgehensmodell vereinfacht die Kommunikation

Wenn die Beteiligten in einem Projekt dieselbe Sprache sprechen und alle dasselbe darunter verstehen (weil es eine saubere Definition der Planungsphase gibt), wird die Gefahr der Missverständnisse geringer.

Vorgehensmodell

3. Ein Vorgehensmodell ist ein Qualitätsmerkmal

Wer ein Modell für sein Handeln hat, der zeigt, dass er über das eigene Handeln nachgedacht hat. Das eigene Handeln zu überdenken und zu optimieren, ist zweifellos eine Maßnahme zu Verbesserung der Qualität der Arbeit.

4. Projekte werden vergleichbar

Wenn Projekte nach einem und demselben Vorgehensmodell gestaltet werden, sind sie besser vergleichbar. Das gilt sowohl für den zahlenmäßigen Vergleich (Wie groß ist der Personalaufwand in den einzelnen Projektabschnitten?), als auch für eine inhaltliche Diskussion.

Basismodell für die Softwareentwicklung

Siehe Tafel

Basismodell für die Softwareentwicklung

Das Projektmanagement leitet und steuert das Projekt.

Der Kernbereich – hier spielt sich der inhaltliche Kern des Projekts ab – ist in mehrere Phasen gegliedert.

Jedes Projekt braucht eine Infrastruktur als Grundlage (zum Beispiel Rechner, auf denen programmiert werden kann).

Eine solche Aufteilung in drei Bereiche findet sich in vielen Modellansätzen – manchmal wird statt von Bereichen auch von Prozessen, genauer von Führungsprozessen, Kernprozessen und Supportprozessen gesprochen.

Basismodell für die Softwareentwicklung

Zuerst muss die Idee geprüft und konkretisiert werden, dies geschieht in der Phase *Auftragsklärung*.

Wenn die Idee umgesetzt werden soll, beginnt das IT-Projekt mit der Phase *Konzeption*, in der die Anforderungen an das System festgelegt werden.

Die anschließende Phase *Design* widmet sich der Frage, wie die Soft- und Hardwarearchitektur des zukünftigen IT-Systems aussehen sollen.

Basismodell für die Softwareentwicklung

In der Phase *Realisierung* wird programmiert.

In der Phase *Einführung* geht es darum, das IT-System direkt beim Kunden zum Laufen zu bringen.

Eine weitere Phase, die den anderen Kernphasen überlagert ist, ist das *Testen*.

Basismodell für die Softwareentwicklung

Die Phasen folgen zeitlich im Prinzip aufeinander. Es gibt an zwei Stellen die Möglichkeit, einen Schritt zurück zu gehen.

Es ist sogar gar nicht selten, dass zum Beispiel in der Phase Design festgestellt wird, dass die Konzeption überarbeitet werden muss.

Das Basismodell sieht nicht vor, bei der Einführung eines Systems plötzlich wieder an der Konzeption zu arbeiten.

Auftragsklärung

Die Phase Auftragsklärung ebnet den Weg von einer vagen Idee zu konkreten IT-Projekten.

Auftragsklärung

Die Ziele dieser Phase bestehen deshalb darin zu prüfen:

- Ob die Umsetzung der Idee aus der Sicht des Geschäfts sinnvoll ist und
- welche IT-Projekte zur Umsetzung der Idee aufgesetzt werden müssen.

Auftragsklärung

Wie kann ein Unternehmen feststellen, ob eine Idee sinnvoll ist?

Auf jeden Fall muss die Wirtschaftlichkeit geprüft werden, zum Beispiel mit der Beantwortung von Fragen wie: Welchen wirtschaftlichen Nutzen brächte die Umsetzung der Idee? Was würde eine Lösung kosten?

Auftragsklärung

Dann ist die Machbarkeit zu klären: Ist die Idee technisch überhaupt umsetzbar? Stehen Ressourcen (Personal, Hard- und Software, Know-how) zur Umsetzung prinzipiell zur Verfügung oder können sie besorgt werden?

Auftragsklärung

Ganz häufig spielt auch die firmeninterne Politik eine große Rolle:

Wer ist Befürworter der Idee?

Wer hat kein Interesse an der Umsetzung?

Welche Konsequenzen hätte die Umsetzung für das Unternehmen insgesamt?

Auftragsklärung

Wohl die wichtigste fachliche Aufgabe in dieser Phase besteht darin, die Idee inhaltlich ganz genau auszuarbeiten und zu präzisieren. Dazu gehört insbesondere:

Eine Einordnung der Idee in das Geschäft des Auftraggebers
Welche Ziele werden mit der Umsetzung der Idee verfolgt? Welche Unternehmensbereiche sind von der Umsetzung der Idee betroffen?

Aufgabe

Formuliert beispielhaft eine Auftragsklärung im Bereich der Spiele.

Konzeption

Die Phase Konzeption klärt die Frage: Was soll das beabsichtigte Produkt leisten?

Die Voraussetzungen zur Beantwortung dieser Frage sind durch die Ergebnisse der Phase Auftragsklärung klar gegeben.

Die Phase Konzeption hat danach noch ein zweites großes Thema zu bearbeiten – die Beschreibung und Bewertung von grundsätzlichen Lösungsalternativen.

Konzeption

Die beste und am häufigsten verwendete Methode zum systematischen Vergleich von Alternativen ist die Nutzwertanalyse. Dabei werden die Alternativen anhand von Kriterien, die unterschiedlich gewichtet sein können, verglichen.

Im Ergebnis erhält jede Alternative genau eine Bewertungszahl – den Nutzwert. Die Alternative mit dem höchsten Nutzwert gewinnt.

Konzeption

Wenn es um die Konzeption geht, fallen oft auch die Begriffe Lastenheft und Pflichtenheft.

Ein Lastenheft beschreibt die Wünsche und Anforderungen aus Sicht des Auftraggebers. Ein Lastenheft kann als ein Ausschreibungsdokument verwendet werden.

Das Gegenstück zu einem Lastenheft ist das Pflichtenheft. Dieses ist die Antwort auf ein Lastenheft.

Das Pflichtenheft beschreibt detailliert die Leistungen, zu der sich der Entwickler verpflichtet

Konzeption

Lastenheft und Pflichtenheft beleuchten also ein und dieselbe Sache aus unterschiedlichen Perspektiven.

Konzeption

Vollständigkeit:

Das Pflichtenheft stellt alle Anforderungen dar.

Widerspruchsfreiheit:

Inhalte des Pflichtenheftes widersprechen sich nicht; es ist wohlüberlegt.

Konzeption

Allgemeinverständlichkeit:

Nicht selten wird ein Pflichtenheft Teil eines juristischen Vertrages zwischen Kunde und Dienstleister. Beide müssen ein gemeinsames Verständnis von den Anforderungen haben.

Präzision:

Um Klarheit im Projekt zu haben und um juristische Auseinandersetzungen zu vermeiden, sollten Begriffe, über die sich trefflich streiten lässt (alle moderne Plattformen, Multisprachfähigkeit, ...), präzise erläutert werden.

Konzeption

Das Pflichtenheft klärt die Frage

Was soll realisiert werden?

und nicht die Frage:

Wie soll es umgesetzt werden?

Nach Vorlage eines Pflichtenheftes kann man in der Regel noch nicht anfangen zu programmieren.

Aufgabe

Überlege ein Beispiel für ein Lasten- und Pflichtenheft.

Details zur Nutzwertanalyse

Die Nutzwertanalyse ist eine Methode, welche die Entscheidungsfindung bei (komplexeren) Problemen rational unterstützt.

Die Nutzwertanalyse ist ein relativ altes Verfahren, das seine Ursprünge in der Volkswirtschaft hat.

Details zur Nutzwertanalyse

Die Nutzwertanalyse findet u. a. Anwendung im Projektmanagement und in der Volkswirtschaftslehre, also dort, wo eine Beurteilung auf Basis mehrerer quantitativer und qualitativer Kriterien, Zielen oder Bedingungen getroffen werden muss.

Eine Nutzwertanalyse wird häufig erstellt, wenn weiche Kriterien vorliegen, anhand derer zwischen verschiedenen Alternativen eine Entscheidung gefällt werden soll.

Details zur Nutzwertanalyse

Die entscheidungstheoretische Grundlage für die Nutzwertanalyse bildet die sog. additive multiattributive Wertefunktion w .

Diese ordnet jeder Alternative einen Wert in Abhängigkeit von ihren Attributausprägungen zu.

Am Ende wird ein Gesamtwert für jede Alternative aus der gewichteten Summe von Einzelwerten pro Attribut errechnet.

Details zur Nutzwertanalyse

Die additive multiattributive Wertefunktion für die Berechnung des Gesamtwertes einer Alternative a lautet

$$v(a) = \sum_{r=1}^m w_r v_r(x_r).$$

Dabei gilt für alle r , dass $w_r > 0$ ist und es gilt die folgende Bedingung:

$$\sum_{r=1}^m w_r = 1.$$

Ferner liegen die Werte für $v_r(x_r)$ stets im Intervall $[0, 1]$.

Details zur Nutzwertanalyse

Das bedeutet, dass jedes Gewicht w_r größer als 0 sein muss und die Summe aller Gewichte 1 (bzw. 100 Prozent) ist.

Der Term $v_r(x_r)$ ist der Wert (bzw. die Bewertung), welcher der Ausprägung x_r zugeordnet ist.

Details zur Nutzwertanalyse: Beispiel

Es werden drei Stellenangebote miteinander verglichen, wobei zwei Attribute zur Bewertung herangezogen werden, nämlich die Arbeitszeit und das Gehalt: Siehe Tafel.

Aufgabe

Führe eine Nutzwertanalyse bei einem frei gewählten Beispiel aus dem Games-Bereich mit zwei Alternativen und drei Attributen durch.

Design

In der Phase Design geht es um die Frage: Wie soll das Produkt intern aufgebaut sein?

Design

Im Design sind folgende Aufgaben zu erledigen:

1. Die Hard- und Softwarearchitektur müssen geklärt werden.
2. Die einzelnen Baugruppen innerhalb der Softwarearchitektur (auch Komponenten genannt) müssen benannt werden.
3. Die Transformation der Komponenten in die geplante IT-Landschaft (beim Kunden) muss geplant werden.

Design

Zum ersten Punkt: Eine Architektur sagt uns, aus welchen Teilen ein Ganzes besteht und wie diese Teile zusammenhängen. So ist das auch bei der Architektur eines IT-Systems.

Allerdings wird hier noch unterschieden zwischen der Softwarearchitektur und der Hardwarearchitektur.

Eine Softwarearchitektur legt fest, welche Softwarekomponenten es gibt und wie sie miteinander zusammenarbeiten.

Design

Paul Clements beschreibt Softwarearchitektur als Strukturen eines Softwaresystems: Softwareteile, die Beziehungen zwischen diesen und die Eigenschaften der Softwareteile und ihrer Beziehungen.

Im Rahmen der Softwareentwicklung repräsentiert die Softwarearchitektur die früheste Softwaredesign-Entscheidung (Architekturentwurf). Sie wird u. a. durch Softwarequalitätskriterien, also Eigenschaften wie Modifizierbarkeit, Wartbarkeit, Sicherheit oder Performance bestimmt.

Design

Eine einmal eingerichtete Softwarearchitektur ist später nur mit hohem Aufwand abänderbar.

Die Entscheidung über ihr Design ist somit einer der kritischsten und wichtigsten Punkte im Entwicklungsprozess einer Software.

Design

Die Hardwarearchitektur definiert, welche Hardware (Server, Workstations, Drucker usw.) zum beabsichtigten System gehört und wie die Hardwarekomponenten miteinander verbunden sind.

Design

Zweiter Punkt: Eine Softwarekomponente ist erstens definiert durch ihre Schnittstellen und zweitens durch ihr (Ausgabe-) Verhalten bei bestimmten Eingaben.

Design

Dritter Punkt: Das IT-System soll später unter bestimmten technischen Bedingungen laufen, wie zum Beispiel einer vorgegebenen Betriebssystemumgebung – diese Randbedingungen müssen von vorn herein berücksichtigt werden.

Design

Die Phase Konzeption klärt, was ein Produkt leisten soll:
Funktionalität und sonstige Anforderungen. (Die Kunden als
fachliche Experten sind besonders gefordert.)

Die Phase Design klärt, wie das Produkt aufgebaut ist:
Architektur, Softwarekomponenten, Plattform.

Aufgabe

Finde ein (einfaches) Beispiel für eine Softwarearchitektur.

Gib explizit die vorkommenden Komponenten und ihre Schnittstellen an.

Realisierung

Das Motto der Phase Realisierung lautet: Erschaffe es!

Realisierung

Die Realisierung erfolgt in den Strukturen, die im Design erarbeitet wurden. Jeder Entwickler ist dann einer Komponente zugeordnet.

Eine besondere Rolle spielt in der Realisierung das Testen. Das ist sehr wichtig – so wichtig, dass dem Testen im Basismodell eine eigene Phase gewidmet ist.

In der Phase Realisierung werden die Konzeption und das Design in einer Programmiersprache implementiert. Die Phase wird daher manchmal auch Implementation oder Implementierungsphase genannt.

Einführung

In der Realisierung wird das IT-System erschaffen – das geschieht oft in den Räumen und auf den Rechnern des Dienstleisters. Nun kommt es darauf an, das Produkt beim Kunden ans Laufen zu bringen.

Testen

In der Entwicklung von IT-Systemen gibt es eine Besonderheit: das Testen.

Testen zielt darauf ab, Fehler in einem Produkt zu finden, damit diese beseitigt werden können. Diese eigentlich einfache Aufgabe wird bei großen Produkten relativ schnell recht kompliziert.

Testen

Durch Testen lassen sich Fehler aus dem System fischen. Ein Nachweis, dass ein System fehlerlos ist, ist nicht möglich.

Wenn kein Fisch mehr ins Netz geht, so heißt das nicht, dass kein Fisch mehr da ist. (Freue dich über jeden Fehlerfisch...)

Je eher und intensiver getestet wird, umso besser. Ein Fehler ist umso billiger, je früher er im Projektverlauf auffällt.

Testen

Beispiel:

Für ein System zur Kundenverwaltung könnte in der Konzeption definiert werden, dass zu jedem Neukunden das Geburtsdatum gespeichert werden muss. Ein fachlicher Test würde dann überprüfen, ob es möglich ist, einen neuen Kunden ohne Geburtsdatum zu speichern.

Ein technischer Test für ein solches System würde beispielsweise prüfen, ob das System auch bei gleichzeitigem Zugriff von 20 Anwendern nicht abstürzt.

Testen

Ein Testfall ist die Beschreibung eines bestimmten Tests. Dabei enthält der Testfall im Wesentlichen Aussagen darüber:

- was getestet werden soll (Ziel des Tests),
- eine Beschreibung dessen, was zu tun ist (Ablauf und Daten des Tests) und
- welche Ergebnisse erwartet werden (damit man sehen kann, ob das Produkt richtig funktioniert).

Testen

Für das Beispiel eines Kundenverwaltungssystems könnte ein Testfall also so aussehen:

— Ziel:

Neukunden können nur mit der Angabe des Geburtsdatums aufgenommen werden.

— Beschreibung:

- Lege einen neuen Kunden im System an
- Lasse das Feld für das Geburtsdatum leer
- Speichere den Datensatz

— Ergebnis:

Das System weist den Speicherversuch mit einer Fehlermeldung zurück.

Testen

Ein Testplan beschreibt, wie in einem Projekt das Testen gestaltet werden soll. Er ist sozusagen der Projektplan für das Teilprojekt des Testens.

Der Testplan enthält zum einen Testfälle und zum anderen einen Plan, wann wer welche Testfälle durchführt.

Er soll das Vorgehen im Testen insgesamt festlegen.

Aufgabe

Betrachte die schwache KI Eliza:

<http://www.med-ai.com/models/eliza.html.de>

Wie könnte ein Testfall dafür aussehen, dass Eliza einer menschlichen Intelligenz entspricht?

Wie könnte ein Testfall für ein Navigationssystem gestaltet werden?

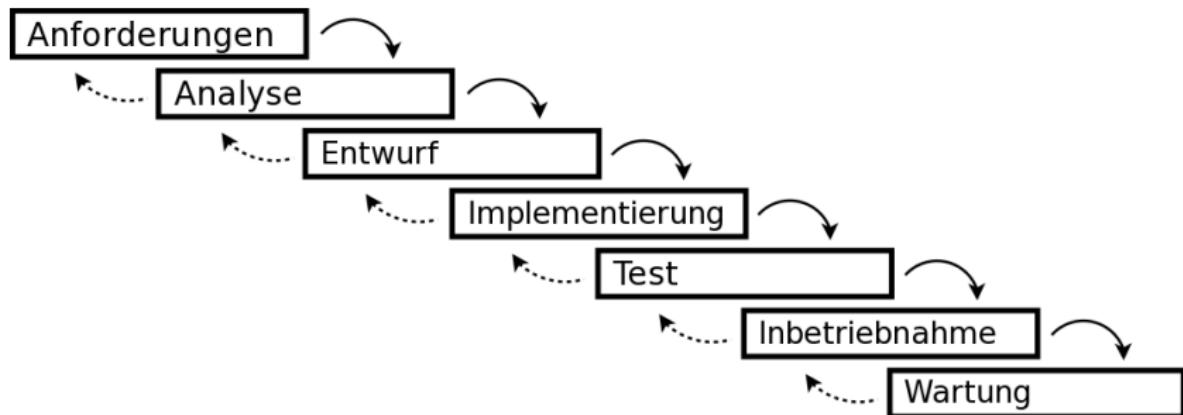
Rollen bei der Entwicklung

Siehe Tafel.

Weitere Modelle

Es gibt nicht nur das Basismodell, sondern weitere.

Wasserfallmodell



Wasserfallmodell

- Es ist ein Phasenmodell.
- Die Phasen folgen strikt nacheinander. Eine Phase muss erst vollständig abgeschlossen sein, bevor die nächste Phase beginnen kann. Es gibt in der klassischen Variante keine Rücksprungmöglichkeit in eine vorherige Phase.
- Das Modell ist sehr einfach, daher gut verständlich.
- Das Modell behandelt nur das, was wir Kernbereich nennen – das Projektmanagement und die Projektinfrastruktur werden ausgeblendet.
- Das Modell sagt nichts über Rollen und Methoden.

V-Modell

Siehe Tafel.

V-Modell

Der Modultest überprüft die Programmierung der kleinsten Einheiten, nämlich der Programmmodulen.

Die einzelnen Module werden zu Komponenten zusammengefasst (man sagt auch integriert), die dann im Integrationstest getestet werden.

Alle Komponenten zusammen ergeben das System, das im Systemtest getestet wird.

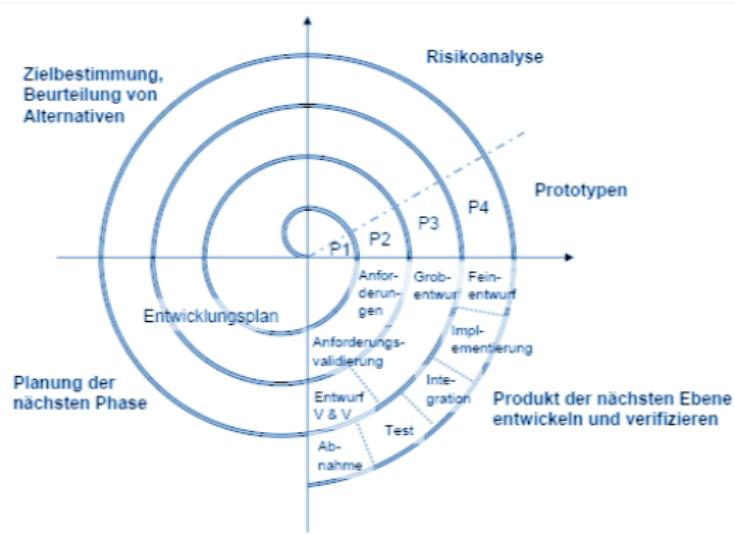
Schließlich gibt es den Abnahmetest, der das Produkt im Hinblick auf seine ursprünglichen fachlichen Anforderungen überprüft.

V-Modell

Das V-Modell ist als verbindliches Vorgehensmodell zu verwenden, wenn der Auftraggeber ein Bundesministerium oder eine ähnliche staatliche Stelle ist.

Man sagt deshalb auch, dass das V-Modell das Vorgehensmodell des Bundes sei.

Spiralmodell



Spiralmodell

Das Spiralmodell basiert auf der Idee, während des Projektfortschritts immer wieder gleichartige Zyklen zu durchlaufen.

Das Spiralmodell ist geprägt von den vier Quadranten, in denen die Zyklen ablaufen.

Spiralmodell

1. Es beginnt mit der Festlegung der Ziele und der Rahmenbedingungen sowie der Identifikation der möglichen Alternativen für die IT-Lösung.

Bei Projektbeginn (also im ersten Zyklus) stehen typischerweise die groben Gesamtziele, Rahmenbedingungen und Lösungen im Fokus, während bei einem späteren Zyklus detaillierte Ziele für einzelne Systemkomponenten beschlossen werden.

Spiralmodell

2. Die Alternativen werden näher ausgestaltet und bewertet. Am Ende ist klar, welche der Alternativen umgesetzt werden soll. Auch hier gilt: Im ersten Durchlauf wird es um die Bewertung prinzipieller Lösungsansätze für das beabsichtigte System gehen, während in einem späteren Zyklus Details bewertet werden.

Spiralmodell

3. Die beste Alternative wird umgesetzt, getestet und abgenommen. Im ersten Durchgang ist das beispielsweise die Abnahme von Konzeptionsdokumenten, später getestete Software.

Spiralmodell

4. Den Abschluss eines Zyklus bildet die Planung der nächsten Runde.

In diesem Modell gibt es eine besondere Betonung der Risikoanalyse. Dabei wird in jedem Zyklus im zweiten Schritt geschaut, welche Risiken die Lösungsalternativen bergen. Wenn eine brandneue Technologie eingesetzt werden soll, ist der Programmieraufwand möglicherweise schwer abschätzbar. Damit besteht das Risiko der Kostenexplosion. Die Risikobetrachtung als festgelegte Aufgabe in jedem Zyklus ist eine Stärke des Modells.

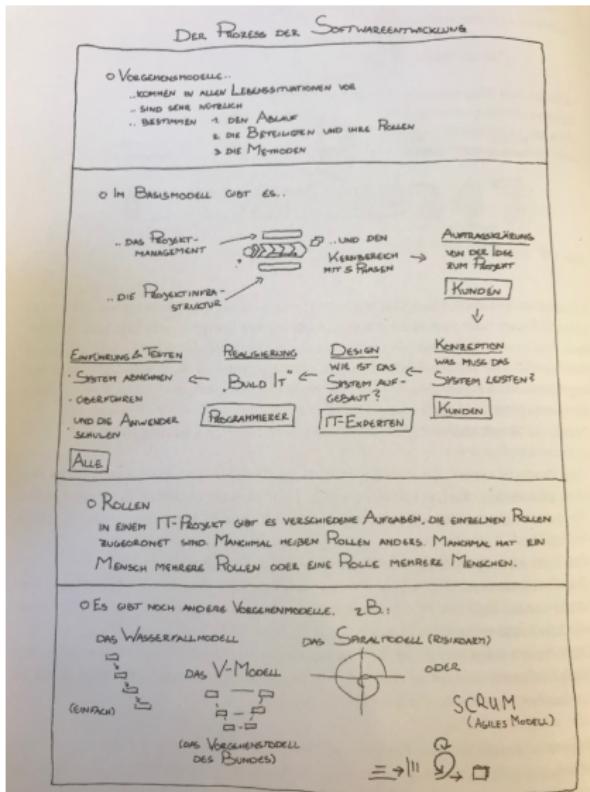
Aufgabe

Finde ein konkretes Beispiel, bei dem Du das Spiralmodell anwendest.

Weitere Modelle

Es gibt weitere Modelle, wie z. B. das der extremen Programmierung oder Scrum (bereits bekannt). Für den praktischen Teil soll eines der zuvor behandelten Modelle verwendet werden.

Zusammenfassung



Fortlauf der Veranstaltung

1. Behandlung der Methoden der Softwareentwicklung I (heute).
2. Partner suchen, der das Lasten- und Pflichtenheft abnimmt und Euch bei der Arbeit begleitet (nach der Mittagspause).
3. Pitchen des Projektes, das in der praktischen Phase umgesetzt und DOKUMENTIERT wird (voraussichtlich nach der Mittagspause).
4. Diskussion der jeweiligen Vorschläge (voraussichtlich nach der Mittagspause).
5. Behandlung der Methoden der Softwareentwicklung II (Kurzvorträge, nächster Termin). 6. Start der Praxisphase (nächster Termin).

Methoden in der Softwareentwicklung

Nachdem wir im vorhergehenden Teil einen Überblick über den Ablauf eines Softwareprojektes gegeben haben, sehen wir uns jetzt Werkzeuge/Methoden an.

Methoden in der Softwareentwicklung

Die nachfolgend vorgestellten Diagramme gehören zusammen – sie bilden die sog. Unified Modelling Language (UML).

Sie entstand in den 90er Jahren des letzten Jahrhunderts durch das Zusammenfassen von verschiedenen Ansätzen.

Bitte beachtet die folgende Übersicht an der Tafel.

Prozesshierarchiediagramm

Im Prozesshierarchiediagramm, geht es wesentlich um betriebliche Prozesse.

Das Prozesshierarchiediagramm hilft damit dem IT-Dienstleister beim Verständnis des Geschäfts des Kunden. Erst im Gespräch zwischen Dienstleister und Auftraggeber erfährt ersterer, wie der Betrieb organisiert ist, welche Geschäftsprozesse es gibt und an welcher Stelle eine neue IT-Lösung hilfreich wirken soll.

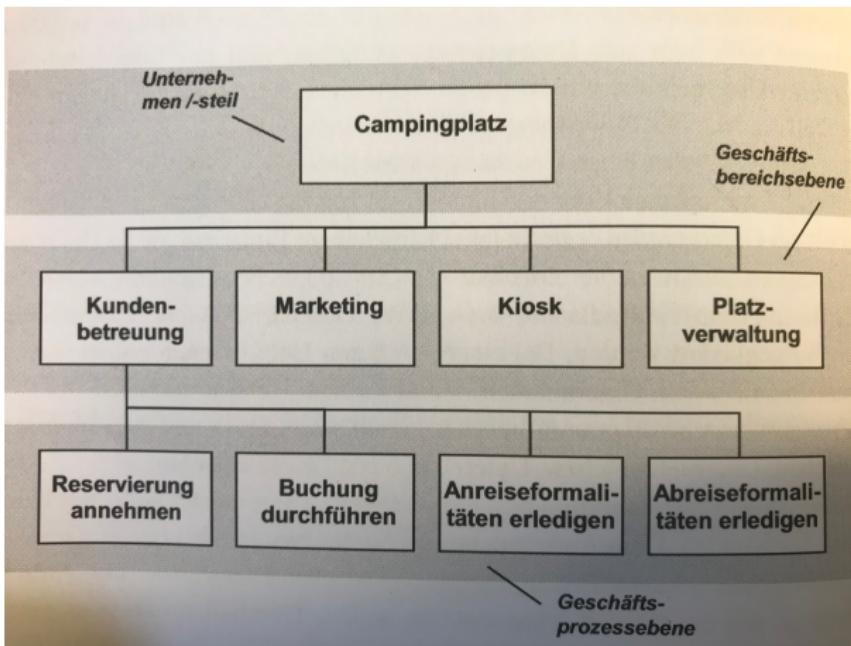
Es ist sinnvoll, wenn der Dienstleister in einem solchen Kundendialog eine Grafik erstellt, die das Umfeld und damit auch die Einbettung der IT-Lösung deutlich macht. Und genau dafür ist ein Prozesshierarchiediagramm ideal.

Prozesshierarchiediagramm

Die Methode dient dem Grundverständnis und ist somit eine Technik der ersten Stunden.

Sinnvoll ist daher die Erstellung und Verwendung von einem oder mehreren Prozesshierarchiediagrammen in der Phase der Konzeption.

Prozesshierarchiediagramm: Beispiel



Prozesshierarchiediagramm

Das Beispiel zeigt Geschäftsprozesse auf einem Campingplatz.

Zunächst besitzt das Campingplatzunternehmen eine Reihe von Geschäftsbereichen. Es wird beschrieben, was in dem Geschäftsbereich Kundenbetreuung zu tun ist.

Die Geschäftsbereiche Marketing, Kiosk und Platzverwaltung haben eigene Prozesshierarchiediagramme, dargestellt wird nur die Kundenbetreuung, mit folgenden vier Geschäftsprozessen:

Prozesshierarchediagramm

Reservierung annehmen:

Per Telefon oder auf anderem Wege meldet sich ein Kunde und möchte eine Campingplatzparzelle für einen bestimmten Zeitraum reservieren.

Prozesshierarchiediagramm

Buchung durchführen:

Der Kunde will in einer bestimmten Zeit eine Campingplatzparzelle mieten. Der Kontakt zwischen Kunde und Campingplatz erfolgt beispielsweise über Telefon, per Mail oder durch das persönliche Gespräch. Es kann sein, dass ein Kunde mit seinem Zelt oder Wohnwagen beim Campingplatz eingetroffen ist und jetzt sofort einen Stellplatz buchen will.

Prozesshierarchiediagramm

Anreiseformalitäten erledigen:

Erst wenn ein Kunde beim Campingplatz eintrifft, müssen Daten ausgetauscht und behördliche Auflagenerfüllt werden. Außerdem müssen dem Kunden Informationen über den Platz, wie zum Beispiel die Platzordnung, bekannt gemacht werden.

Prozesshierarchiediagramm

Abreiseformalitäten erledigen:

Wenn ein Kunde den Campingplatz verlässt, muss u.a. die Rechnung bezahlt werden. Es kann auch sein, dass noch einige andere Dinge geklärt werden müssen. Zum Beispiel, ob Meldebescheinigungen den korrekten Abreisezeitpunkt beinhalten oder ob die genutzte Campingplatzparzelle ordentlich hinterlassen wurde.

Prozesshierarchiediagramm

Notationselemente:



Prozesshierarchiediagramm



Übung

Erstellt ein PHD für ein Start Up im Bereich Games.

Schnittstellendiagramm

Ein Firmenchef könnte bei der Vorstellung seines Betriebes formulieren: Unsere Schnittstelle zum Kunden ist unsere Vertriebsabteilung.

Ein anderer bei demselben Anlass: Die wichtigste Schnittstelle zum Kunden ist das Internet. Von dort bekommen wir 85 aller Aufträge.

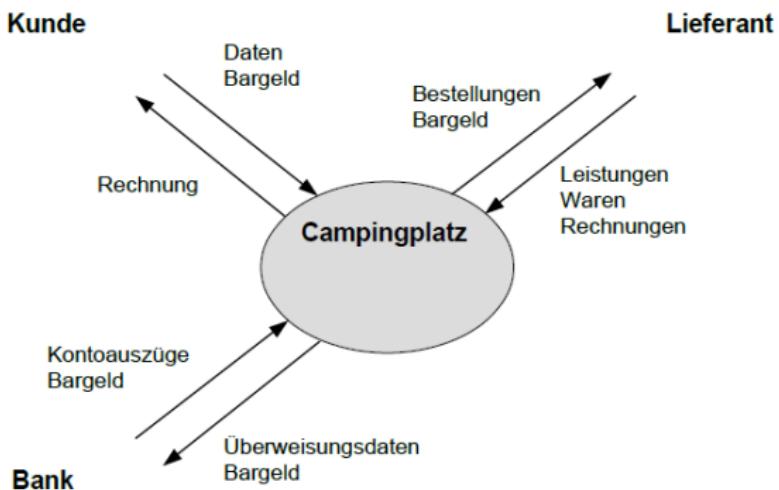
Daraus ergibt sich:

- Eine Schnittstelle kann festlegen, welche Informationen oder auch Waren zwischen zwei Systemen oder Betrieben ausgetauscht werden.
- Eine Schnittstelle wird manchmal sehr präzise (Datenaustausch) und manchmal etwas allgemeiner (Geschäftsbeziehungen) beschrieben.

Schnittstellendiagramm

Ein Schnittstellendiagramm (SSD) zeigt die Schnittstellen zwischen einem Geschäft oder einem IT-System und seinen Partnern.

Schnittstellendiagramm: Beispiel



Schnittstellendiagramm

Das Beispiel zeigt Schnittstellen im Campingplatzgeschäft und damit verbundene Waren-, Geld- und Informationsflüsse. Was genau ausgetauscht wird und in welche Richtung es fließt, kann man an der Beschriftung und Richtung der Pfeile erkennen.
Folgendes lässt sich z. B. interpretieren:

Es gibt eine Schnittstelle vom Campingplatz zum Kunden. Ein Kunde des Campingplatzes wird zahlungspflichtig. Er bekommt eine Rechnung, gibt dem Campingplatz seine persönlichen Daten und bezahlt mit Bargeld.

Schnittstellendiagramm: Beispiel

Nun folgt ein weiteres Beispiel, bei dem aber nicht das Geschäft des Campingplatzes, sondern ein geplantes IT-System für dieses Geschäft im Mittelpunkt steht:

Schnittstellendiagramm: Beispiel

Siehe Tafel.

Schnittstellendiagramm

Geschäftssicht:

Wenn man für einen großen Verbrauchermarkt darstellt, welche Waren, Formulare und Geldmengen das Geschäft durchströmen, ergibt dies die Geschäftssicht.

IT-Sicht:

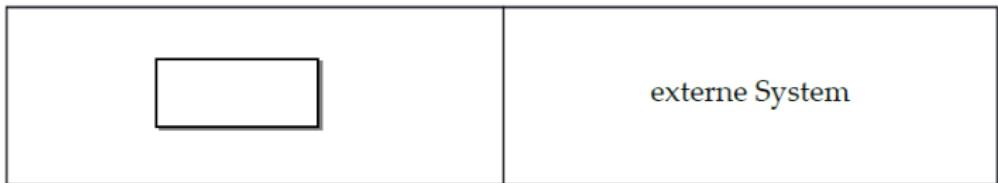
Der Verbrauchermarkt arbeitet vielleicht mit einem zentralen IT-System. Dieses arbeitet nicht mit Waren, wohl aber mit den Daten der Waren. Eine Aufgabe des zentralen IT-Systems ist es, in Abhängigkeit von der Anzahl der verkauften Artikel, beim Lieferanten automatisch einen Bestellvorgang auszulösen. Diese Datenfüsse vom eigenen zu einem IT-System des Lieferanten gehören zur IT-Sicht.

Schnittstellendiagramm

Notationselemente:



Schnittstellendiagramm



Schnittstellendiagramm



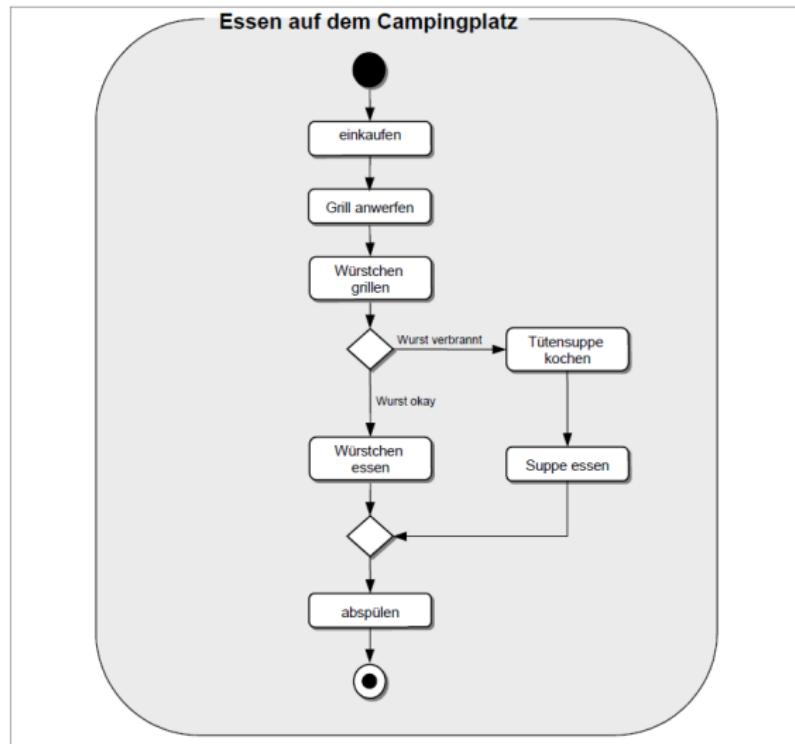
Aktivitätsdiagramm

Das Aktivitätsdiagramm gehört zu der Familie der UML-Diagramme. Im Prozess der Softwareentwicklung wird es zu verschiedenen Zwecken genutzt:

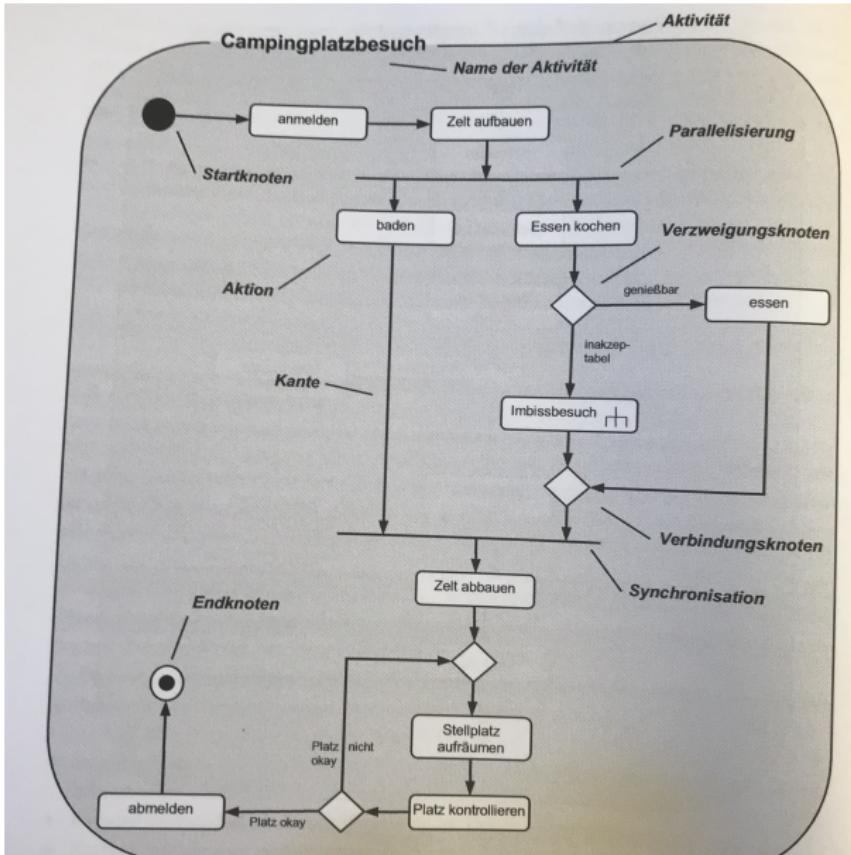
- In einer frühen Projektphase dient es dazu, Geschäftsabläufe in der Übersicht zu beschreiben.
- Es wird verwendet, um Geschäftsabläufe im Detail darzustellen.
- Softwarearchitekten und Programmiererinnen können mit diesem Diagramm Abläufe in der Software visualisieren.

Das Diagramm ist sowohl für Auftraggeber als auch für IT-Experten interessant.

Aktivitätsdiagramm: Beispiel



Aktivitätsdiagramm: Beispiel



Weitere Themen in Kurzvorträgen