

Einführung in die Informatik, Algorithmen und Datenstrukturen

Teil 1 - Thema 7

Entwicklung und Nutzung von
Software

aktuelle Softwaretechnologien

Business Analysten → IT-Architekten → Programmierer

PLM - Product Life Cycle Management

ALM - Application Life Cycle Management

SCM – Supply Chain Management

ERP – Enterprise Resource Planning

Identity Management

CRM - Customer Relationship Management

EDM – Engineering Data Management

PDM – Product Data Management

Softwareprojekte

„80% aller Softwareprojekte scheitern!“

(Dale Fuller, Chef des Entwicklungsspezialisten
Borland, 2004)

Software

Software: Sammel-Bez. für Programme, die für den Betrieb von Rechensystemen (z. B. Computern) zur Verfügung stehen, einschließlich der dazugehörigen Dokumentation.

/Lexirom 4; Meyers Lexikonverlag/

Software, Computerprogramme. Anweisungen, die dafür sorgen, daß die Hardware, also der Rechner, läuft. Software kann anhand der Art der durchzuführenden Aufgaben in verschiedene Kategorien eingeteilt werden. Die Hauptkategorien umfassen einerseits die Betriebssysteme (Systemsoftware), die die Arbeiten des Computers steuern, und andererseits die Anwendungssoftware, die eine Vielzahl von Funktionen übernimmt, die Menschen von einem Computer erwarten. ...

/"Software," Microsoft® Encarta® 97 Enzyklopädie. © 1993-1996 Microsoft Corporation/

Entwicklung und Nutzung von Software

PROGRAMM:

Ein Programm ist die Darstellung einer Folge von Operationen (arithmetische, logische) über Daten. Diese Folge stellt eine eindeutige und endliche Lösungsvorschrift dar, die aus bestimmten Eingabedaten bestimmte Resultatdaten erzeugt. Die Darstellungsform des Programms muss so sein, dass die Operationsfolge durch eine Rechenanlage abgearbeitet werden kann.

DATEN:

Datum = (Name, Typ, Wert)

Entwicklung und Nutzung von Software

OPERATIONEN:

EINGABEOPERATIONEN stellen die Werte für die Eingabedaten des Programms von einem Eingabemedium bereit.

AUSGABEOPERATIONEN sorgen u.a. für die Darstellung der vom Programm errechneten Resultate auf dem Bildschirm, Drucker oder Speicher.

WERTZUWEISUNGEN übertragen den errechneten Wert eines rechts von ":= " stehenden Ausdrucks der links stehenden Variablen als ihren Wert.

PROGRAMM STEUERENDE OPERATIONEN dienen zur Realisierung von Verzweigungen.

Eigenschaften von Software

- Software ist nicht sinnlich wahrnehmbar
- Software besteht aus Sprache
- Software ist digital
- ***Software ist fehlerhaft***
- große Softwaresysteme sind komplex
- große Softwaresysteme bestehen aus umfangreichen Texten
- große Softwaresysteme verfestigen Sichtweisen

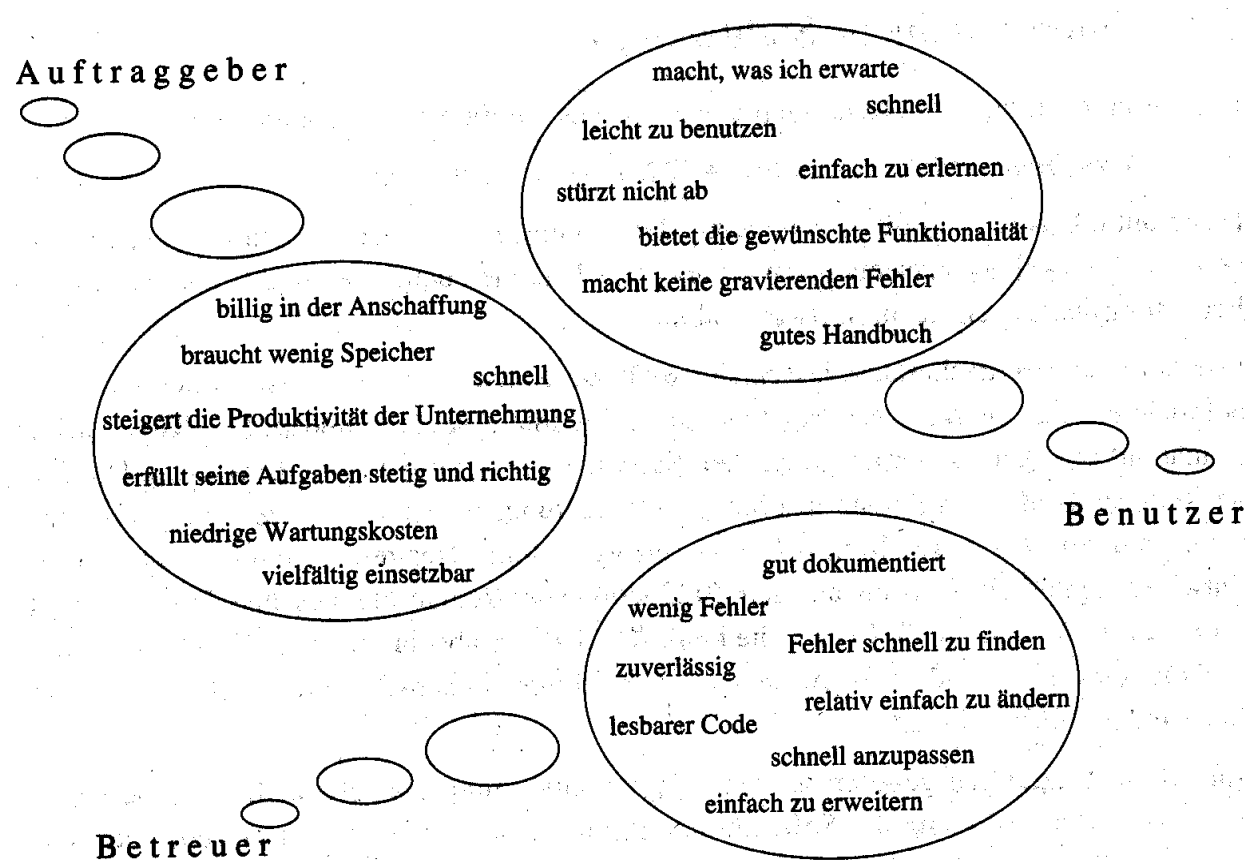
Softwarequalität

„Softwarequalität ist, nach DIN 55350, die Gesamtheit der Eigenschaften oder Merkmale, die Software in Verwendung und (Weiter-) Entwicklung aufweist, um die an sie gestellten Anforderungen zu erfüllen.“

Bewertungskriterien für Softwareprodukte:

Korrektheit, Konsistenz, Vollständigkeit,
Bedienkomfort, Portabilität, Adaptierbarkeit,
Wartbarkeit, Änderbarkeit, Lesbarkeit,
Strukturierung, Modulierung, Hierarchisierung

Softwarequalität aus unterschiedlichen Blickwinkeln



Softwareentwicklung - Programmieren

„**Programmieren** bedeutet das Lösen von Problemen der realen Welt mit Hilfe des Computers. Dazu gehören alle Tätigkeiten, die vom umgangssprachlich formulierten Problem zum präzisen, formalen Programm führen.“

/Pagel, Six; Software

Engineering; 1994/

geringe Programmkomplexität \Rightarrow „Programmieren im Kleinen“

Softwarekrise

Als es noch **keine Rechner** gab, war auch das **Programmieren** noch **kein Problem**,
als es dann ein paar **leistungsschwache Rechner** gab, war das **Programmieren ein kleines Problem** und nun,
wo wir **gigantische Rechner** haben,
ist das **Programmieren zu einem gigantischen Problem** geworden.

In diesem Sinne hat die elektronische Industrie kein einziges Problem gelöst, sondern nur neue geschaffen.

Sie hat das Problem geschaffen, ihre Produkte zu benutzen.

(E.W. Dijkstra 1972)

Umfang von Softwareprojekten

Projekt	Software Lines of Code
NASA Space Flight control	420000(shuttle) + 1,4 Millionen (ground)
Microsoft Windows 3.1	3 Millionen
Microsoft Windows NT	4 Millionen
Sun Solaris	7-8 Millionen
eBay (Q4 2004)	9 Millionen
Microsoft Windows 95	12-15 Millionen
Linux – Redhat 6.2	17 Millionen
Microsoft Windows 98	18 Millionen
Microsoft Windows NT 5.0	20 Millionen
SAP R/3	25 Millionen
Windows Server 2003	40-50 Millionen

Fragen bei der Softwareentwicklung

- Wie konzipiert, analysiert und integriert man Gesamtsysteme, um die gewünschten Eigenschaften zu erhalten?
- Welche Hilfsmittel gibt es und wie werden sie verwendet?
- Wie integriert man Hardware- und Softwarearchitekturen?
- Wie kommuniziert der Mensch über geeignete Graphikoberflächen und Schnittstellen mit dem Gesamtgebilde?
- Welches sind Funktionen, die der Mensch optimal ausführen kann und welche Aufgaben kann das System autonom erledigen?

/Paul Levi, Ulrich Rembold; Einführung in die Informatik; Carl Hanser Verlag München Wien 2003/

Softwareentwicklung - Projekte

„Die Erstellung von Software erfolgt in Projekten. Der Begriff Projekt signalisiert,

- dass ein **vorgegebenes Arbeitsziel**
- in einer **vorgegebenen Zeit**
- unter **Einsatz** von vorhandenen, meist **beschränkten Mitteln** erarbeitet werden muss.

/Forbig, Kerner; Lehr- und Übungsbuch Softwareentwicklung, Fachbuchverlag Leipzig; 2004/

Software Engineering

Software Engineering bezeichnet das geplante, systematische Vorgehen bei der Softwareentwicklung unter Anwendung von Methoden, Verfahren und Softwarewerkzeugen mit dem Ziel, qualitativ hochwertige Softwareprodukte wirtschaftlich herzustellen und zu nutzen.

/Horn97/

(**Engineering**: Transformation einer Spezifikation in ein funktionales System)

Software Engineering

Aufgaben des Softwareentwicklers:

Umsetzung einer Idee in ein reales Produkt mit Hilfe von wissenschaftlichen Erkenntnissen und unter Verwendung von Werkzeugen der Informationsverarbeitung.

→ Der Softwareentwickler bzw. das Entwicklungsteam muss über Kompetenzen auf dem Gebiet der Konstruktionslehre, Produktionstechnik, Qualitätssicherung und Projektorganisation für Software verfügen.

Softwareentwicklung ist (bisher) nicht automatisierbar → Softwareentwicklung erfordert Kreativität

Software Lebenszyklus

„**Softwarelebenszyklus:**

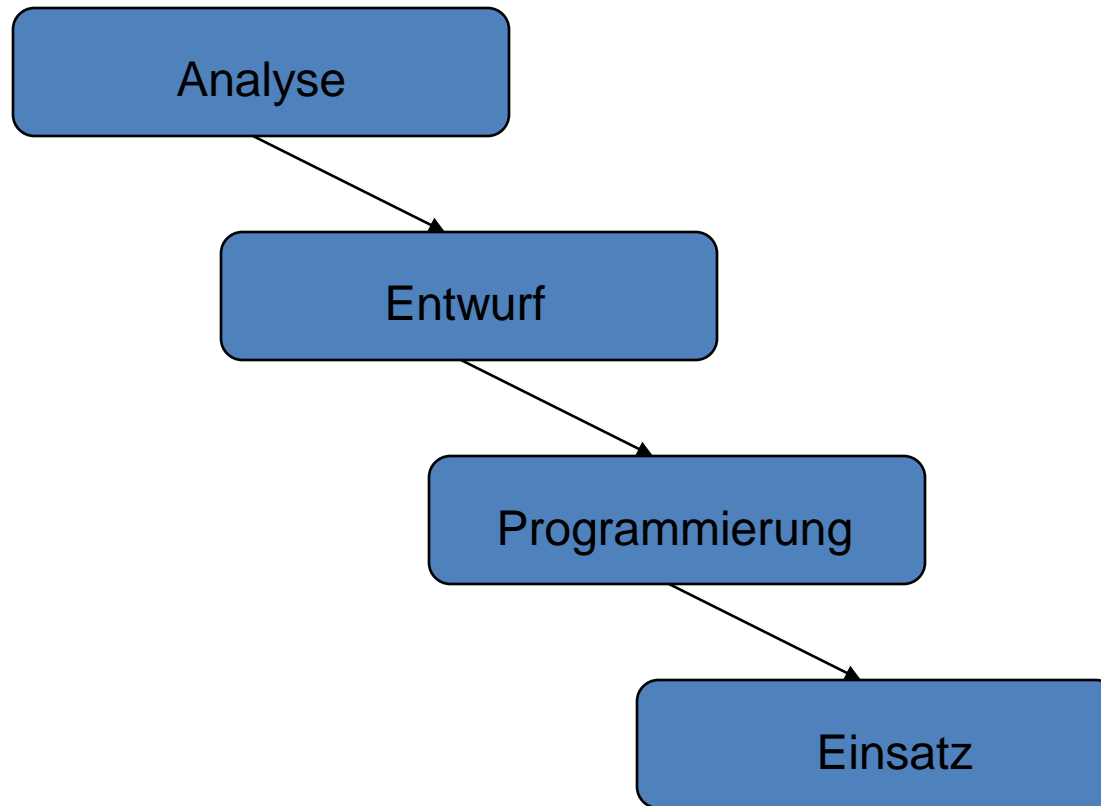
Der Softwarelebenszyklus umfasst den Lebenslauf eines Softwaresystems vom Projektbeginn über dessen Nutzung und Betreuung bis zur Außerbetriebnahme.

Softwareentwicklungsprozess:

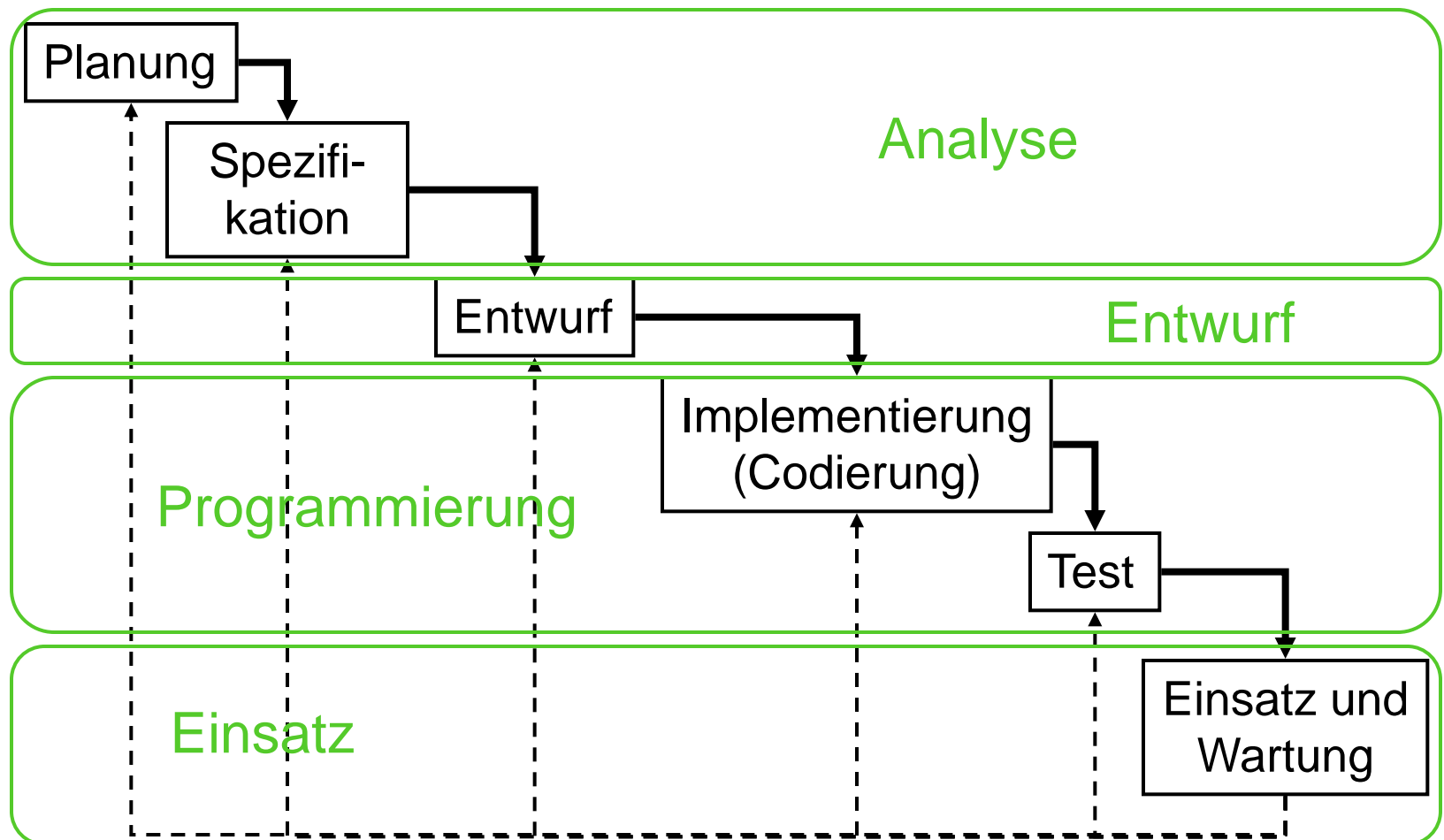
Ein Softwareentwicklungsprozess ist der Handlungsablauf zur Erstellung von Softwaresystemen bei der Umsetzung betrieblicher Tatbestände, geformt vom Vorgehensmodell und vom Projektmodell.“

/Forbig, Kerner; Lehr- und Übungsbuch Softwareentwicklung, Fachbuchverlag Leipzig; 2004/

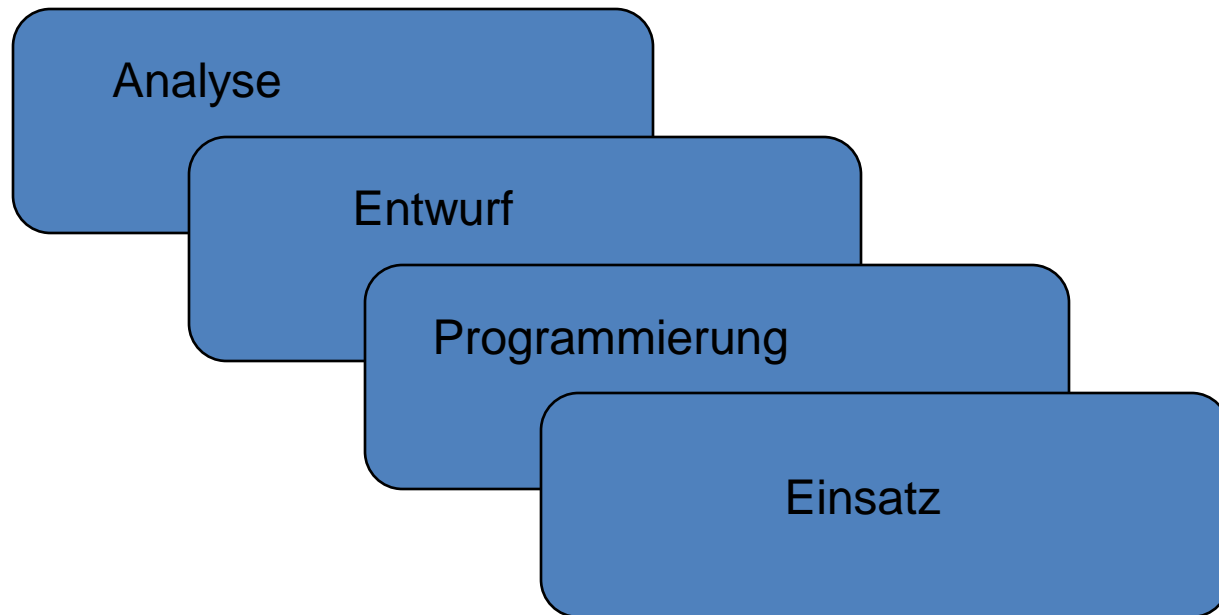
Softwarelebenszyklus - klassisches Wasserfallmodell



Softwarelebenszyklus (klassisches Wasserfallmodell)



Softwarelebenszyklus – heutiges Wasserfallmodell –(ansatzunabhängig)



Analysephase - Systemanalyse

„**Systemanalyse** geht traditionell von der Einsicht aus, dass eine Anwendung nur dann sinnvoll entwickelt werden kann, wenn die **Aufgabe inhaltlich voll verstanden ist**. Also ist es erforderlich, dass die Entwickler sämtliche verfügbaren Unterlagen gründlich studieren und vor allem die Anwender befragen.“

aber: die Aufgaben sind komplex und es gibt viele Akteure mit verschiedenen Sichtweisen

→ Nur die **Abschnitte des Anwendungsbereiches werden analysiert**, die für das Softwaresystem relevant sind.

Analysephase – Grundlegende Aufgaben der Systemanalyse

Die **Aufgabe muss verstanden werden**. Das damit verbundene Erkenntnisproblem ist überlagert durch die unterschiedlichen Erfahrungswelten der beteiligten Personen.

→ interdisziplinäre Zusammenarbeit mit geeigneten Kommunikationsmitteln

Es muss ein **Modell** erarbeitet werden, das die **neugestaltete Arbeitswelt** in der Sprache des Anwenders **beschreibt**, gleichzeitig aber präzise und detailliert genug ist, um die Systemkonstruktion abzuleiten.

→ Entwicklung im Rahmen eines Autor-Kritiker-Zyklus

Analysephase - Systembeschreibung

→ **Lastenheft** (Aufgabenstellung)

mit:

- Beschreibung des zukünftigen Systems,
- Beschreibung der Zielsetzung,
- Beschreibung der Umweltschnittstellen sowie Randbedingungen.

Entwurfsphase

Probleme:

- komplexe Aufgabenstellungen müssen in überschaubare Teilaufgaben zerlegt werden
- Trennung von Anwender – Programmierer – Entwickler
- Beachtung der psychologischen, sozialen und rechtlichen Einflüsse

Entwurfsphase – ausgewählte Aspekte

Grundlegende Aspekte, die die Systemarchitektur beeinflussen:

- Sollen Programmteile parallel ausführbar sein?
- Ist das System für einen oder mehrere Nutzer gleichzeitig auszulegen?
- Sollen Teilsysteme auf verschiedene Rechner verteilt werden?
- Sind Echtzeitanforderungen zu erfüllen?
- Ist das System interaktiv oder läuft es im Stapelbetrieb?
- Werden Datenstrukturen im Hauptspeicher, in Dateien oder in einer Datenbank gehalten?
- Haben die geforderten Qualitätsmerkmale Auswirkungen auf die Systemarchitektur?

Entwurfsphase - Projektvorbereitung

→ **Pflichtenheft**

juristisch verbindliche Formulierung des Vertrages zwischen dem Auftraggeber und dem Softwareentwicklungsunternehmen zur Herstellung und Abnahme des Softwaresystems

- mit detaillierter Planung des Projektumfangs (Zeitdauer, Kosten und Personal- und Materialbedarf)
- Haftung
- Infrastruktur

Entwurfsphase - Entwurf von Algorithmen

Verfahren der schrittweisen Verfeinerung:

- **grobe Spezifikation des Algorithmus** (durch Vor- und Nachbedingung)
- **Definition der algorithmischen Aktionen**, die zur Erfüllung der Nachbedingung auszuführen sind, Spezifikation der Zwischenzustände, Deklaration der zugehörigen Daten
- **interaktive Verfeinerung der Beschreibung von Aktionen**, bis diese durch entsprechende Konstrukte einer Programmiersprache ausgedrückt werden können

Codierung/Implementierung

Die Codierung ist die Umsetzung der Algorithmen in eine *geeignete* oder *geforderte* Programmiersprache.

Werkzeuge der Codierung:

- Entwicklungsumgebungen,
- Compiler,
- Prüf- und Fehlersuchprogramme,
- Editoren.

Codierung/Implementierung - Test

Die Testphase hat die Aufgabe, die **Zuverlässigkeit** und **Korrektheit** der Software nachzuweisen.

Dies wird durch Fehlerbeseitigung und Vergleich von Soll- und Ist-Eigenschaften realisiert. Es wird die Verifikation und Validierung durchgeführt.

Korrektheit:

Ein Programm gilt als **korrekt**, wenn es die **vorgegebene Aufgabenstellung erfüllt**.

→ eine fehlerhafte Aufgabenstellung führt zu einem korrekten, aber unbrauchbaren Programm

Qualitätssicherung – Verifikation und Validierung

„Verifikation:

Ein Kalkül zum Führen eines formalen Beweises, der zeigen soll, dass ein Programm P eine Spezifikation S erfüllt.

Testen:

Überprüfung eines Programms P mit einer Testmenge T , ob P für T die Spezifikation S erfüllt.

Validierung:

Überprüfung einer Spezifikation S , ob diese den Anforderungen A tatsächlich entspricht.“

/Peter Forbrig, Immo O. Kerner, Lehr- und Übungsbuch Softwareentwicklung, Fachbuchverlag Leipzig, 2004/

Dokumentation - Systemdokumentation

Bestandteile der Systemdokumentation

- Spezifikation (vollständige Ergebnisse der Phase Analyse)
- Systemarchitektur (vollständige Ergebnisse der Phase Entwurf)
- Programmentwurf (Entwurfsunterlagen wie Ablaufpläne, Struktogramme, Datenflusspläne)
- kommentierte Modulquelltexte
- Testplan und Ergebnisprotokolle
- Inhalt und Ergebnis des Abnahmetests

Dokumentation - Anwendungsbeschreibung

Bestandteile der Anwenderdokumentation

- Bedienungsanleitung (kurze Beschreibung der Funktion des Systems für den Endnutzer)
 - Installationsanleitung (Vorgehensweise bei der Installation des Systems auf der gewünschten Hardware)
 - Referenzhandbuch (ausführliche Anleitung für die Benutzung des Systems, detaillierte Informationen über Bedingungen und Abläufe im System)
 - Administrationshandbuch (notwendige Unterlagen für den Systemverantwortlichen, z.B. Verwaltung der Nutzer und Zugriffsrechte, Inhalte und Verwendung von Logfiles)
- ➔ Viele Dokumentationsteile werden heute in Hilfesysteme integriert.

Softwarelebenszyklus – V-Modell

Das **V-Modell** ist eine abstrakte, umfassende Projektmanagement-Struktur für die IT-Systementwicklung.

Der Name bezieht sich auf die V-förmige Darstellung der Projektelemente wie IT-Systemdefinitionen und Tests, gegliedert nach ihrer groben zeitlichen Position und ihrer Detailtiefe.

Das V-Modell wird seit 1986 entwickelt und kommt ursprünglich aus dem militärischen Bereich. Die aktuelle Version ist das V-Modell XT (V1.2 2/2006).

V-Modell

Das V-Modell ist als Leitfaden zum Planen und Durchführen von Entwicklungsprojekten unter Berücksichtigung des gesamten Softwarelebenszyklus konzipiert.

Das V-Modell regelt detailliert, „Wer“ „Wann“ „Was“ in einem Projekt zu tun hat.

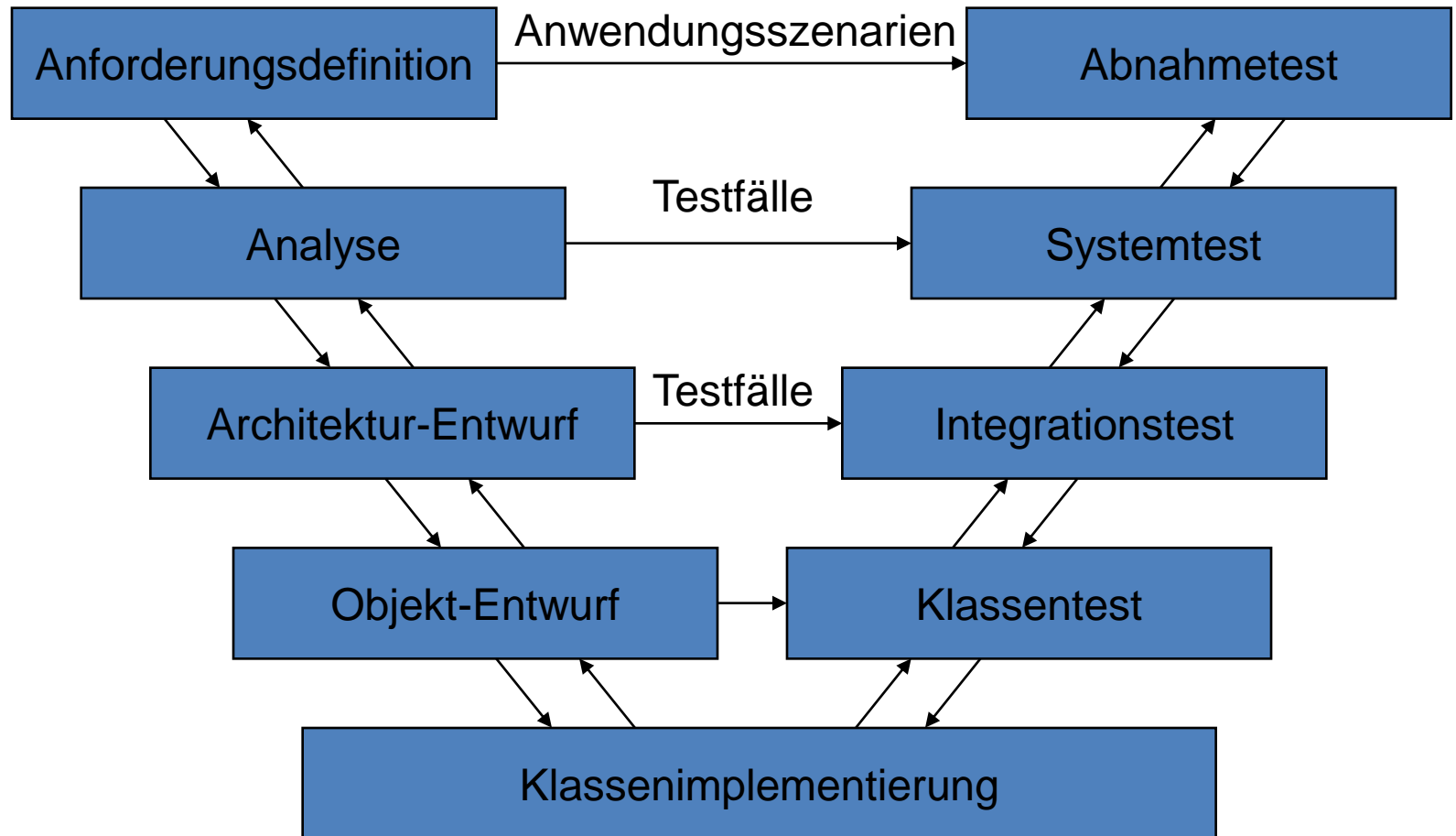
1997 wurde mit der Veröffentlichung des Entwicklungsstandards für IT-Systeme des Bundes das V-Modell 97 als Vorgabe für den Einsatz im gesamten zivilen und militärischen Bereich gültig.

2004 wurde das V-Modell überarbeitet und zum **V-Modell XT** weiterentwickelt.

V-Modell - Zielsetzung

- Minimierung der Projektrisiken
- Verbesserung und Gewährleistung der Qualität
- Eindämmung der Gesamtkosten über den gesamten Projekt- und Systemlebenszyklus
- Verbesserung der Kommunikation zwischen allen Beteiligten

Softwarelebenszyklus – V-Modell

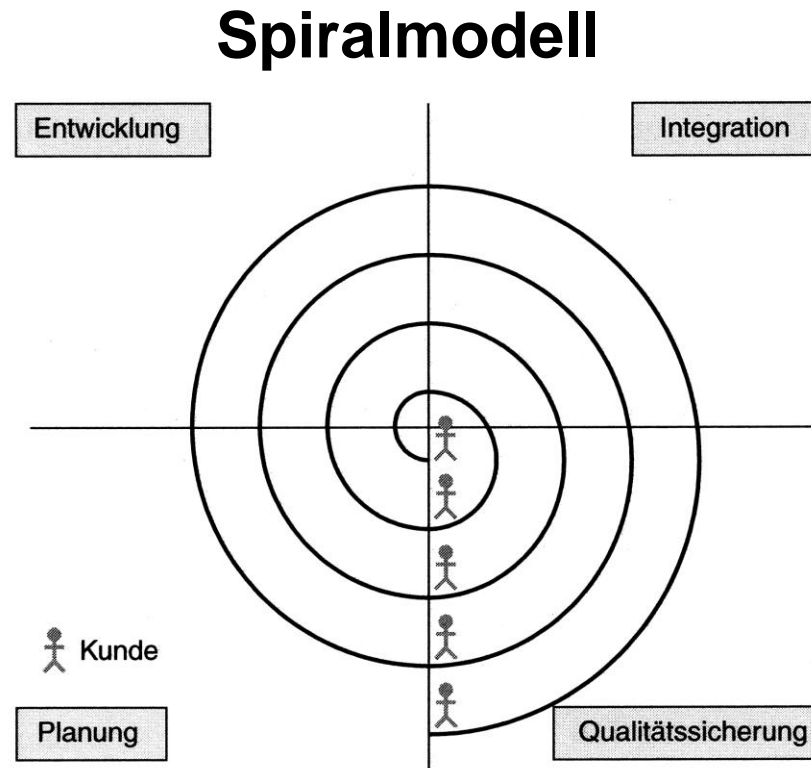


Inkrementelle Softwareentwicklung

Die **inkrementelle Softwareentwicklung** basiert auf dem Spiralmodell. Zuerst wird für den Kunden ein attraktives Teilsystem erstellt. Es durchläuft die Phasen Planung, Entwicklung, Integration und Qualitätssicherung. Dann wird das **System schrittweise durch neue Funktionalitäten erweitert**. Dabei wird in der Spirale jeweils eine neue Erweiterung bearbeitet, und zwar mit allen Phasen, die dazu gehören. Der Kunde wird bei jedem Inkrement beteiligt. Dadurch wird gewährleistet, dass Unzufriedenheiten oder Fehler den Entwicklern früh gemeldet und beseitigt werden können.

/Peter Forbrig, Immo O. Kerner, Lehr- und Übungsbuch Softwareentwicklung, Fachbuchverlag Leipzig, 2004/

Inkrementelle Softwareentwicklung



/Peter Forbrig, Immo O. Kerner, Lehr- und Übungsbuch Softwareentwicklung, Fachbuchverlag Leipzig, 2004/

Projektbearbeitung

Projekte werden heute in der Regel nicht von Einzelpersonen, sondern von Entwicklungsteams realisiert.

→ Wie können Ideen, Gedanken, Konzepte, Lösungen ... so dargestellt werden, dass sie für alle Teammitglieder verständlich und nutzbar sind?

→ Während der Projektbearbeitung entwickelt sich die umgebende, reale Welt weiter.

Literatur

- /FO04/ Peter Forbrig, Immo O. Kerner
Lehr- und Übungsbuch Softwareentwicklung
Fachbuchverlag, Leipzig, 2004
- /LE03/ Paul Levi, Ulrich Rembold
Einführung in die Informatik;
Carl Hanser Verlag, München, Wien, 2003
- /OR02/ Orlamünder; Liskowski; Hußmann
Softwareentwicklung mit Delphi
Fachbuchverlag, Leipzig, 2002

Kontrollfragen

1. Erklären Sie die Begriffe Software und Softwaresystem.
2. Was versteht man unter einer Softwareentwicklungsumgebung? Nennen Sie 3 Komponenten der Softwareentwicklungsumgebung.
3. Erläutern Sie die Begriffe „Softwarelebenszyklus“ und „Lebenszyklusmodell“. Vergleichen Sie das Wasserfallmodell und das V-Modell. Nennen Sie die Phasen des Softwarelebenszyklus.
4. Was versteht man unter inkrementeller Softwareentwicklung? Welches Vorgehensmodell wird bei diesem Ansatz verwendet?
5. Erläutern Sie die Begriffe „Software“ und „Computerprogramm“. Nennen Sie die Grundoperationen, die auf Daten ausgeführt werden können. Beschreiben Sie grundlegende Eigenschaften von Software.
6. Erläutern Sie den Qualitätsbegriff für Softwareprodukte. Nennen Sie fünf Bewertungskriterien für Softwareprodukte und erläutern Sie diese kurz.
7. Erläutern Sie die Begriffe „Softwarekrise“ und „Softwareengineering“. Beschreiben Sie den Softwarelebenszyklus am Beispiel des Wasserfallmodells. Gehen Sie auf die unterschiedlichen Ansätze der klassischen und objektorientierten Betrachtungsweise ein.