1. **Einleitung**

Computer ohne Programme sind unbrauchbar. Unter einem Programm versteht man eine Folge von zusammenhängenden, maschinenverständlichen Arbeitsanweisungen an einem Computer, um eine Verarbeitung von Daten zu ermöglichen. Die Gesamtheit aller Programme, die für ein Rechnersystem zur Verfügung stehen nennt man Software. Software umfasst alle Produkte und Dienstleistungen, die eine sinnvolle Nutzung der Hardware ermöglichen. Wir betrachten im unteren Teil wie überhaupt so eine Software entwickelt wird und wie wir eine Software anhand eines Fallbeispiels mit wichtigen Kriterien bewerten.

1. **Softwareentwicklungsprozess**

Bei der Entwicklung einer Software ist es wichtig systematisch vorzugehen, wobei der Prozess durch Vorgehensmodelle strukturiert wird. Die Vorgehensmodelle beinhalten die Organisation für die Softwareentwicklung sowie eine gut durchdachte Strategie für die Durchführung eines Projekts.

Dazu gehört:

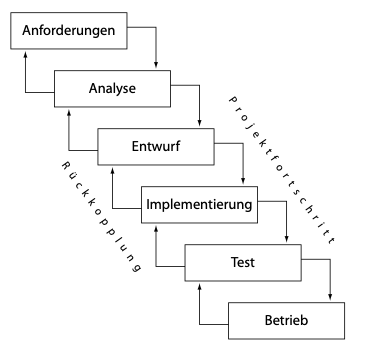
* Das Erstellen einer Aktivitäten Liste, wo die Reihenfolge des Vorgehens enthalten ist
* Die Festlegung des Inhalts und der Form von Teilprodukten
* Die Kriterien für die Fertigstellung und Abnahme der Software
* Die Verteilung von Rollen, Verantwortlichkeiten sowie Kompetenzen
* Standards, Richtlinien, Methoden & Werkzeuge

Die Aufgabe dieser Vorgehensmodelle ist es den komplexen Prozess der Entwicklung und der anschließenden Wartung in überschaubare Teilaktivitäten zu zerlegen und deren Ergebnisse sowie logische und zeitlichen Zusammenhänge zu definieren.

Diese Gliederung wird anhand des Software Lebenszyklus dargestellt

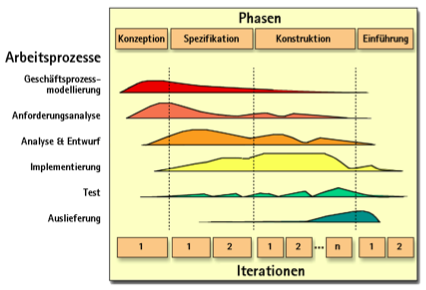
* 1. **Software Lebenszyklus**

1. **Anforderungsanalyse:** In der Anforderungsanalyse werden die Ziele der Softwareent wicklung beschrieben. Sie formuliert die funktionalen und nicht-funktionalen Anforderungen
2. **Systemspezifikation:** Hier werden mit Hilfe von Analysetechniken der Ist – Zustand, Schwachstellen, fachliche Beschreibung von Funktionalität und Leistungsumfang festgehalten. Als Ergebnis wird ein Pflichtenheft formuliert.
3. **Modellierung:** Das zu konstruierende System wird als Entwurf modelliert. Der Entwurf enthält die Struktur des Systems, wie es die Funktionen erfüllt und wie es sich zur Laufzeit dynamisch verhält.
4. **Implementierung:** Bei der Implementierung wir der Systementwurf in eine Software umgesetzt. Das Ergebnis sind Programme und Dokumentation
5. **Integration & Test:** Hier werden alle Komponenten zu einem Gesamtsystem zusammengeführt und getestet. Daraus entsteht das auslieferfähige Endprodukt.
6. **Betrieb & Wartung:** Bei Betrieb und Wartung werden Fehler korrigiert und es entstehen neue Anpassungen weshalb der Kreislauf von vorne beginnt.
   1. **Vorgehensmodelle**
      1. **Wasserfall-Modell**



Beim Wasserfallmodell wird sequenziell vorgegangen. Wenn eine Phase vollständig abgeschlossen wurde ist das Ergebnis dieser Phase die Voraussetzung für die nächstbeginnende Phase. Sollte eine Phase nicht den Anforderungen entsprechen so muss diese Phase erneut durchlaufen werden.

* + 1. **RUP (Rational Unified Process)**

****

Bei RUP werden die Arbeitsprozesse im Laufe des Entwicklungsprozesses in den Projektphasen Konzeption, Spezifikation, Konstruktion und Einführung iterativ mit unterschiedlicher Intensität und Dauer mehrmals durchlaufen.

Quellen:

Wasserfallmodell Bild: <http://spolwig.de/is/softwareprojekte/slc_3.htm>

RUP modell Bild: <http://www.infforum.de/themen/anwendungsentwicklung/se-rup.htm>

Buch Grundkurs Wirtschaftsinformatik dietmar abs

1. **Zu beurteilende Softwarekriterien**
   * 1. **Softwarekosten**

Um eine Software zu erstellen entstehen viele Kosten wie:

* Rechercheaufwand
* Installation der Software
* Anpassungen
* Schulungen für das Personal
* Pflege und Wartung
* Updates
* Wartungskosten
* Mitarbeiterkosten
* Hardware, Strom und Internet
  + 1. **Technische Anforderungen der Software**

Die technischen Anforderungen spielen eher eine geringe Rolle in der Bewertung, da eine Hardware sehr preiswert ist und leicht ausgetauscht oder erweitert werden kann.

Neben Hardwarekosten entstehen aber auch noch Nebenkosten wie Dokumentationsänderungen, Inkompatibilität der bereits laufenden Hardware sowie Schulungskosten.

* + 1. **Unternehmensspezifische Anforderungen der Software**

Unternehmensspezifische Anforderungen sind nicht technische Anforderungen wie:

* Datenschutzrichtlinien
* Steuerregelungen
* Normen
* Risikomanagement
  + 1. **Normen**
    2. **Benutzeroberfläche**

Zur Bewertung der graphischen Benutzeroberfläche sind die notwendigen Anforderungen:

* **Aufgabenangemessenheit:** Dem Benutzer sollen nur die wichtigsten Informationen, im Zusammenhang mit der Aufgabe, angezeigt werden
* **Selbstbeschreibungsfähigkeit:** auf den ersten Blick einschätzen welches grafische Element welche Funktion hat
* **Steuerbarkeit:** Sind Steuerelemente erkennbar die einem die Navigation innerhalb eines Arbeitsablaufs ermöglicht. Gibt es Vor-/Zurück Buttons. Der aktuelle Arbeitsschritt soll innerhalb der Prozessdarstellung hervorgehoben werden
* **Erwartungskonformität:** Bedienelemente sollen dort vorhanden sein wo sie zu erwarten sind. Funktionen die man häufiger benutzt leicht wiederzufinden
  + 1. **Softwaredokumentation**

Die Softwaredokumentation dient als Hilfe für die Benutzer. Es sollte prozessorientiert strukturiert und leichten und schnellen Zugang zur gesuchten Information darstellen.

Sie besteht aus:

* Programmierdokumentation (Quellcodebeschreibung)
* Methodendokumentation
* Installationsdokumentation
* Benutzerdokumentation (Handbuch, Online-Hilfe etc.)
* Datendokumentation
* Testdokumentation
* Entwicklungsdokumentation
  + 1. **Datensicherung**

Datensicherung heißt, dass man die Daten vor Verlust, Manipulation oder vor unberechtigten Zugang durch Dritte schützt. Bei der Datensicherung werden die Daten kopiert. (Backup)

* + 1. **Datenschutz**

Datenschutz ist die Vertraulichkeit, Verfügbarkeit und Integrität von Daten in Softwaresystemen. Bei der Bewertung von Software ist der Datenschutz und die damit verbundenen Informationssicherheit ein wichtiger Aspekt.

* + 1. **Schnittstellen**

Bei der Bewertung von Schnittstellen stellt sich die Frage welche Schnittstellen man für die Software hat.

Allgemein wird zwischen Daten-, Maschinen-, Hardware-, Netzwerk-, Software- und Benutzerschnittstellen unterschieden.

* 1. **Performance**

Für die Performance ist die Geschwindigkeit, Reaktionszeit, Bearbeitungszeit und Berechnungszeit entscheidender Faktoren bei der Bewertung. Eine langsame Software kommt bei den Benutzern seht schlecht an