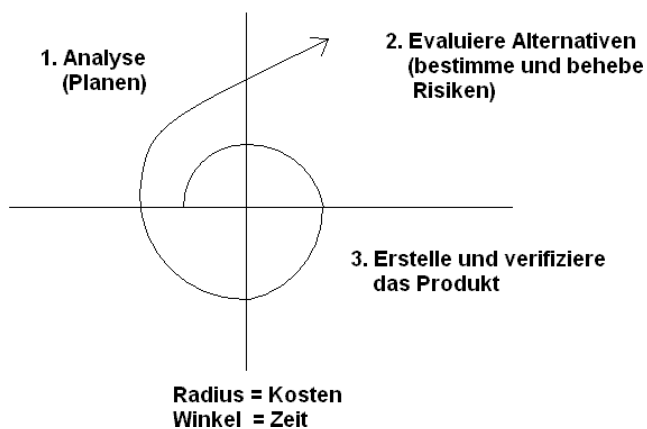


Prozeßmodelle : (organisatorischer Ablauf der Software-Entwicklung)

- **Programmieren durch Probieren (code&fix - bzw. trial&error – model)**
- **Wasserfallmodell (Phasenmodell mit Rückkopplungen)**
- stufenweises Abarbeiten , Ergebnisse einer Phase fallen in die nächste
- **V-Modell (Variante des Wasserfallmodells mit Qualitätsanforderungen)**
- Verifikation und Validierung (mittels ISO 9000)
- **Prototypenmodell**
- erstelle Prototypen -> wirf Prototypen weg -> Wasserfallmodell
- Arten von Prototypen :
 - Demonstrationsmodell \
 - Prototyp im engeren Sinne \ - Wegwerfprototypen
 - Labormuster /
 - Pilotsystem (Kern des Produkts)
- **Versionsmodell (inkrementelles- bzw. evolutionäres Modell)**
- 1. Version , dann verfeinerte Versionen
- **Transformationsmodell**
- formale Spezifikation mit initialem Wissen
- automatische und halbautomatische Transformation in formale Spezifikation
- Benutzung von Software-Systemen
- **ObjektOrientiertes Modell**
- zeit- und kostenmindernd durch Wiederverwendung von Code
- zusätzlicher Aufwand im Management
- **Nebenläufiges Modell**
- Zusammenarbeit von Entwicklungsgruppen
- hoher Management-Aufwand
- **Spiral-Modell (Risiko-Modell)**
- Meta-Modell (alle anderen Modelle sind Spezialfälle)



1. Wasserfallmodell (Erweiterung um Rückkopplungen des stagewise-model)

Phase	Dokumente	Personengruppen
Anforderungsanalyse	- (Lastenheft) - Pflichtenheft (ist genauer als Lastenheft) - Projektplan - vorläufiges Benutzerhandbuch - Verifikationsplan	- Kunde - Systemanalytiker
Entwurf	- Systemarchitektur - Benutzerhandbuch - Verifikationsplan	- Entwickler - Designer - Software-Architekten
Implementierung	- Code - Dokumentation - Testdaten	- Programmierer
Test	„System“	- Programmierer
Betrieb und Wartung	„Produkt“	- Kunde - Kundenberater

2.a) Vorteile :

einfach , verständlich , sequentiell (vom Allgemeinen zum Speziellen) , fester Lösungsweg

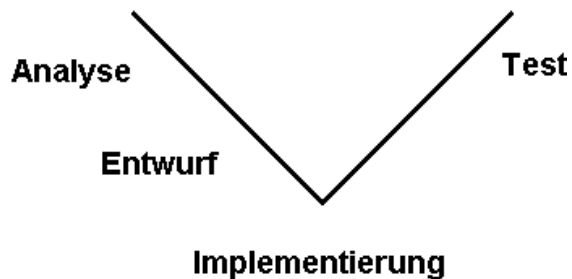
2.b) Nachteile :

Dokumentationsintensiv (viel Bürokratie) , fester Lösungsweg

(Wasserfallsystem eignet sich somit nicht für große Projekte , da alle Spezifikationen bekannt sein müssen)

Kunde wird erst mit dem fertigen Produkt konfrontiert

2.c) V-Modell



3.)

Lastenheft :

- grobes Festlegen von Zielen und Anforderungen
- Aufbau („festgelegt“):
 1. Ziel des Produkts
 - Programm zur Verwaltung von Klausurergebnissen (max 10 Jahre)
 2. Produkteinsatz
 - a) Anwendungsbereich : Eingabe der Klausurergebnisse ,Berechnung der Noten unter Berücksichtigung von Notenskalierungen , Ausgabe von Notenlisten
 - b) Zielgruppe : Sekretärinnen und Assistenten am Lehrstuhl
 3. Produktfunktionen (Lastenheftfunktionen)
 - /LF 10/ Eintragen der Punktzahlen
 - /LF 20/ Ausgabe von Notenlisten
 - /LF 30/ Berechnen der Noten (Skalierung)
 - /LF 40/ Austausch (Anmeldung , Ergebnisse)
 4. Produktdaten (Lastenheftdaten)
 - /LD 10/ Notendaten (ca. 30000 – bei 100 Studenten pro Jahr)
 - /LD 20/ Studentendaten (ca. 1000)
 - /LD 30/ Skalierungsdaten
 5. Produktleistungen (Lastenheftleistungen)
 - /LL 10/ -/LF 20/ - Ausgabe in maximal 2 Sekunden
 6. Qualitätsanforderungen
 - siehe Tabelle Vorlesung
 7. Ergänzungen
 - sehr spezielle Anforderungen

Glossar : (Begriffslexikon – und Beschreibung wer was darf)

- Sekretärin
- Assistent
- Notenskalierung
- Leistungs- und Fachpunkte