

# Abgabe SE2 Übung 5

Furkan Aydin, M1630039

Falk-Niklas Heinrich M1630123

## SE2 Übung 5

### Aufgabe 5.1: Wasserfallmodell

#### 5.1 a)

- Falls erst beim Entwurf Fehler im Design entdeckt werden,
- sich die Wünsche des Kunden ändern,
- technische Probleme die Umsetzung sehr erschweren oder unmöglich machen.

#### 5.1 b)

Die Rückkopplungen im Wasserfallmodell sind auf die vorherige Stufe beschränkt, es ist ein rein sequenzielles Modell. Das Spiralmodell ist hingegen iterativ, hier werden Arbeitsschritte tatsächlich mehrfach ausgeführt. Die Aussage stimmt daher nicht.

### Aufgabe 5.2: Spiralmodell und Unified Process (Pflichtaufgabe)

Beide Modelle haben das Ziel der *Risikominimierung* und sind *iterativ*, sie benötigen beide einen höheren Managementaufwand aufgrund von zusätzlichen Checks und Prototypen bzw. Builds.

Das Spiralmodell gibt nicht die eigentliche Modellierung der Systemerstellung vor, UP beschreibt die Erstellung als Kernelement.

Das Unified Process Modell ist zusätzliche *inkrementell* (im Spiralmodell möglich aber nicht fakultativ) und beschreibt eine evolutionäre Herangehensweise mit frühem ersten, und aufeinander aufbauenden, Builds.

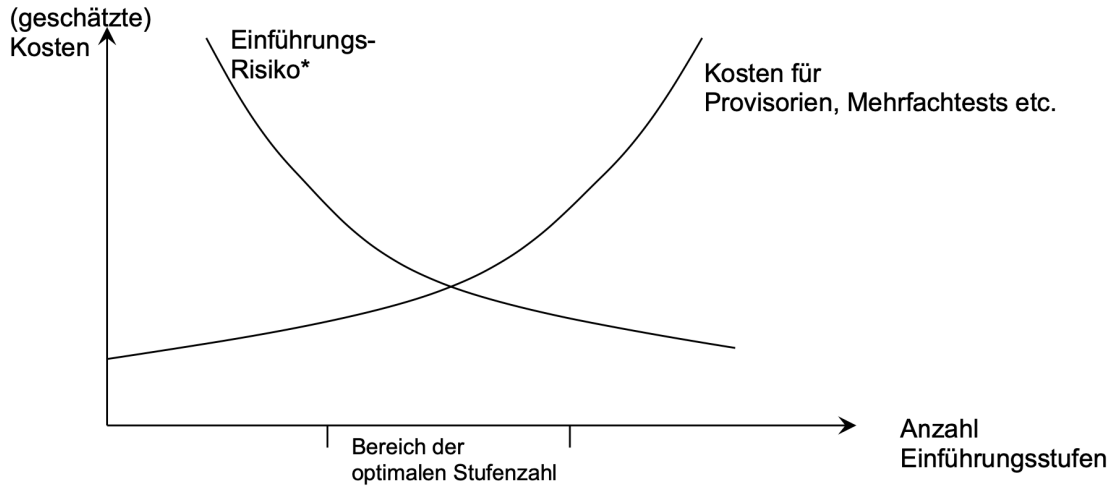
### Aufgabe 5.3: Klassische/Inkrementelle Vorgehensmodelle

#### 5.3 a)

Table 1: Vorteile Big-Bang-Integration vs. stufenweise

Vorteil Big-Bang	Nachteil
geringerer Aufwand geringere Kosten (Builds, Tests)	hohes Risiko bei Einführung Projekt schlechter steuerbar

### 5.3 b)



\* Risikokosten = Schadenserwartungswert bei Systemausfall x Eintrittswahrscheinlichkeit

Eine Herausforderung bei der stufenweisen Einführung liegt in der Bestimmung der optimalen *Anzahl von Stufen* (x-Achse).

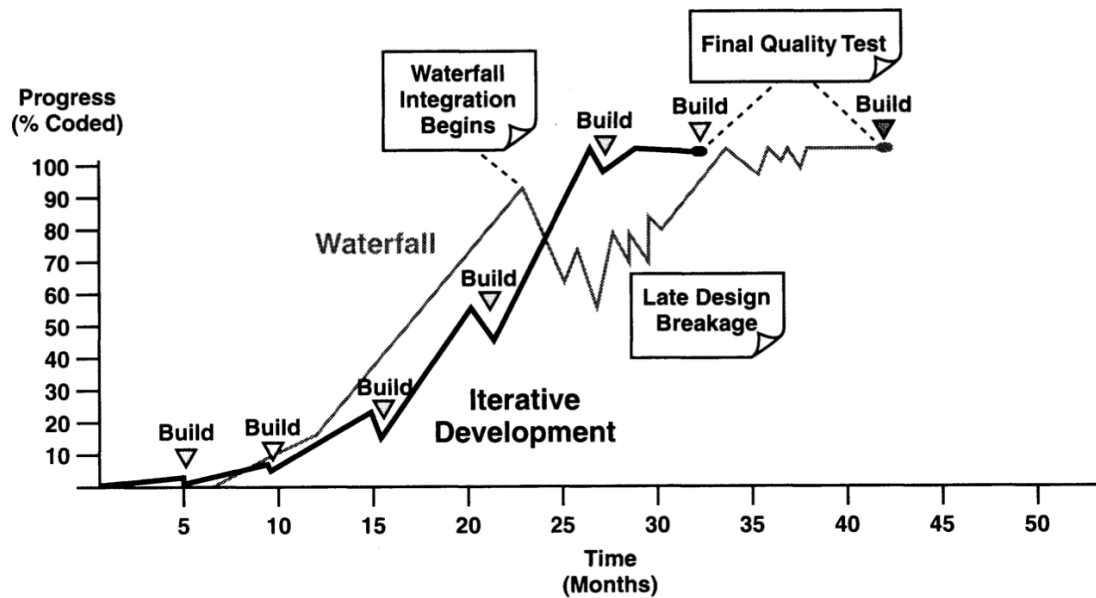
Je feiner die Abstufung (mehr Integrationen), desto geringer ist das Risiko für einen Systemausfall beim Roll-out; allerdings steigen mit jeder Stufe auch die *Kosten* (y-Achse).

Werden andererseits zu wenige Stufen gewählt (Extremfall Big-Bang: 1 Stufe) steigt das Einführungsrisiko (und damit die Risikokosten), da potentiell ein großer Bereich der Anwendung im Fehlerfall nicht verfügbar ist.

Um den Punkt, in dem sich die Kurven für die Kosten des Einführungsrisikos und Build-Kosten schneiden, liegt der Bereich der optimalen Anzahl für die Inkremente.

Hier halten sich Schadenfallkosten und Stufenkosten die Waage.

### 5.3 c)



Die Abbildung stellt den Programmierfortschritt (y-Achse) über die Zeit (x-Achse) dar.

In der Grafik lassen sich gut die einzelnen Stufen der iterativen Entwicklung erkennen.

Beim iterativen Vorgehen zeigen sich immer wieder nach den Builds kleinere Rückschritte um Fehler (auch im Design) zu korrigieren.

Das Wasserfallmodell zeigt einen steileren Anstieg im Fortschritt, wird dann aber (~23. Monat) durch einen erkannten Entwurfsfehler harsch zurückgeworfen.

Nach dem Fortschrittsverlust zeigt sich durch die Zickzacklinie der hektische Versuch der Korrektur. Das Projektende wird in der Grafik vom iterativen Modell deutlich früher erreicht; veranschaulicht wird hier also das große Risiko für eine Zeit- und Kostensteigerung bei der Nutzung des Wasserfallmodells.

Die Programmierung beginnt bei der iterativen Entwicklung sehr früh, im Wasserfallmodell erst später (in der Grafik nach 6-7 Monaten).

### Aufgabe 5.4: Prototypische Entwicklung / Vorgehensmodelle (Pflichtaufgabe)

#### a) Evolutionäres Prototyping / Evolutionäre Softwareentwicklung

In der Evolutionären Softwareentwicklung wird der Prototyp als Grundlage verwendet. Während des Prozesses erweitert man den Prototypen immer wieder (Inkremente) und nähert sich so dem vollständigen Produkt an.

Durch die frühe Verfügbarkeit eines vorzeigbaren Produkts ist es möglich schnell Erfahrungen zu sammeln und Rückmeldungen vom Kunden zu erhalten. Der Prototyp wird dabei stets funktionsfähig und stabil gehalten.

#### b) Wasserfallmodell und Prototyping

Wie könnte man das Wasserfallmodell sinnvoll durch Prototyping erweitern und verbessern?

**Variante A** Das Wasserfallmodell könnte in ein Spiralmodell eingebettet werden. Dadurch würde das Wasserfallmodell sich nur auf einzelne Prototypen/Inkmente beziehen.

**Variante B** Durch das Aufbrechen des Wasserfallmodells, indem man Rücksprünge, von z.B. der Test-Stufe zurück zur Analyse, erlaubt.

**Variante C** Mit 2 Iterationen des Wasserfalls arbeiten: In der 1. Iteration wird ein Prototyp für die Grundfunktionalitäten umgesetzt. Während der 2. Iteration ganz normal Wasserfallmodell zur Vervollständigung der Software; dabei auf das Feedback des Kunden eingehen und berücksichtigen.

### c) Prototyping im V-Modell XT

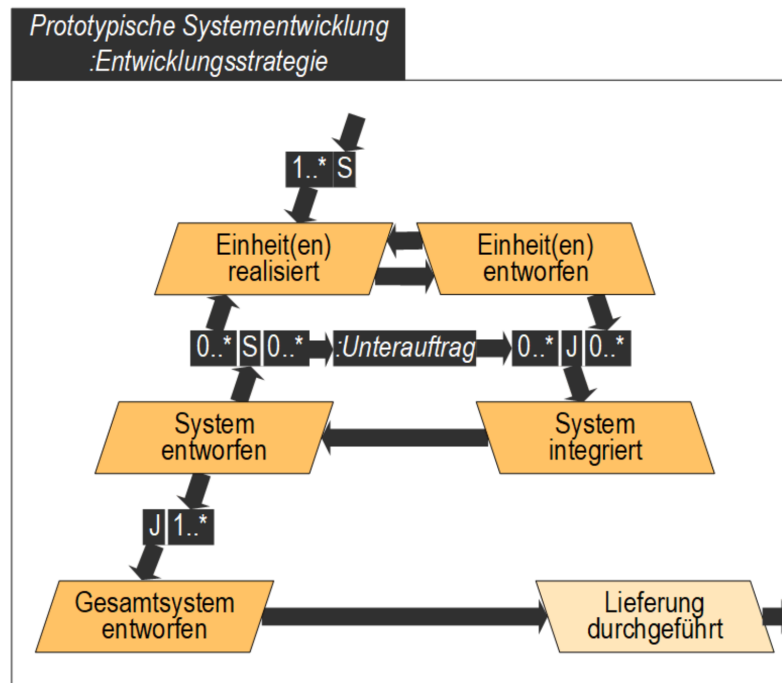


Figure 1: Entwicklungsstrategie bei prototypischer Systementwicklung (aus V-Modell XT Bund 2.3, S. 240)

Dabei erlaubt das V-Modell unterschiedliche Reihenfolgen für den „V-Durchlauf“: Während die Inkrementelle Systementwicklung dem Weg des „V“s unmittelbar folgt (I1-I8), durchläuft die Prototypische Entwicklungsstrategie die Entscheidungspunkte in anderer Reihenfolge (P1-P8). Darüber hinaus sind in beiden Strategien Iterationen auf feineren Abstraktionsebenen möglich.

V-MODELL XT BUND 2.3, S. 38

1. Anforderungen festgelegt
2. Einheit realisiert
3. Einheit entworfen
4. System integriert
5. System entworfen
6. Gesamtsystem entworfen
7. Lieferung durchgeführt
8. Abnahme erklärt

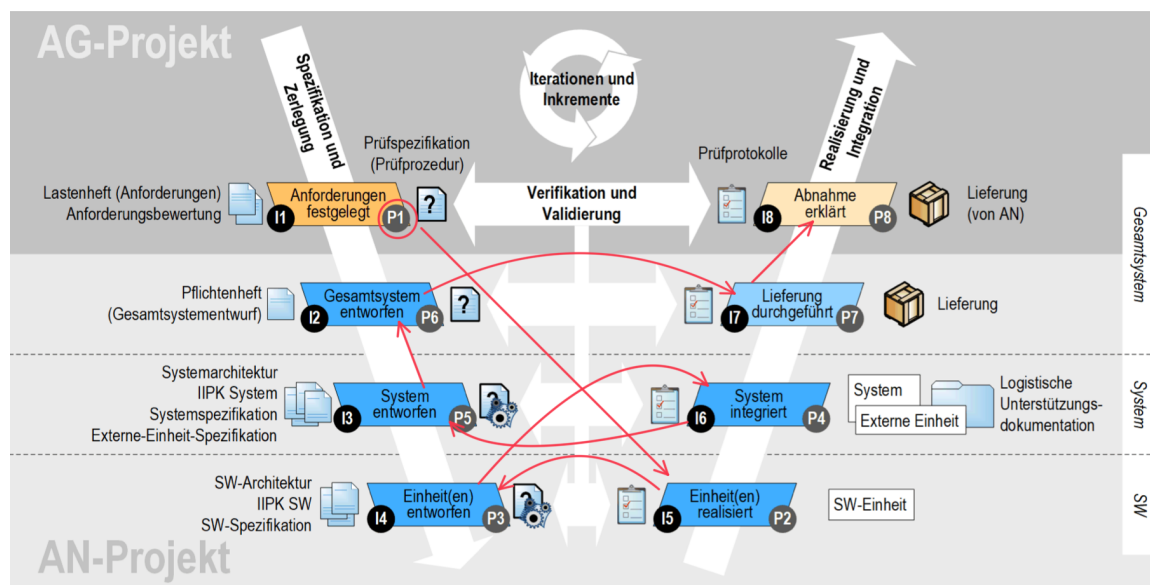


Figure 2: V-Modell-Entwicklungsprozess im Überblick - Anmerkungen Ablauf prototypische Entwicklung (aus V-Modell XT Bund 2.3, S. 38)

Der Auftragnehmer entwirft, realisiert und liefert das System dann ähnlich wie bei der Entwicklungsstrategie inkrementelle Entwicklung in einzelnen Stufen. Diese Stufen werden jede für sich vom Auftraggeber abgenommen.

V-MODELL XT BUND 2.3, S. 243