

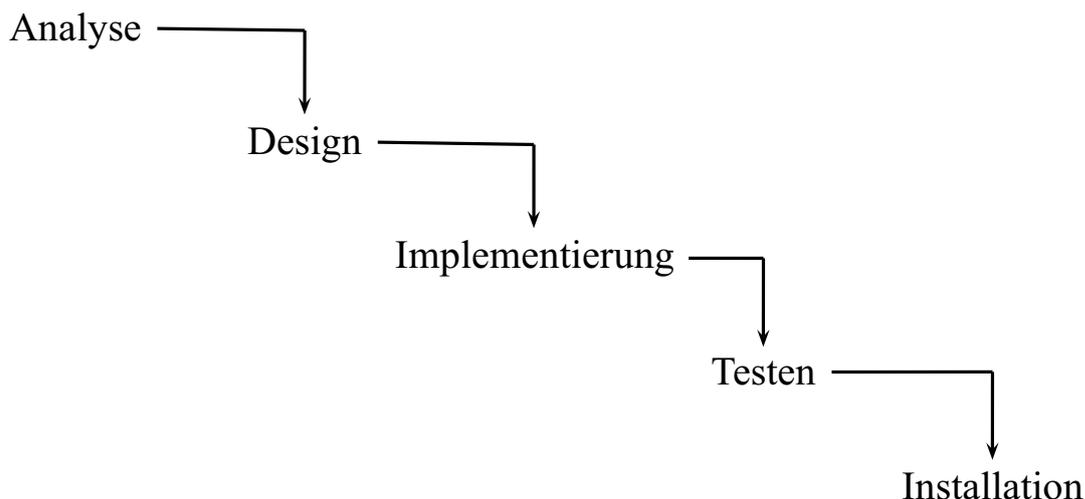
Teil III

Entwurf objektorientierter Programme

Abschnitt 19: Objektorientiertes Design

19. Objektorientiertes Design

- Zur Software-Entwicklung existiert eine Vielfalt von Vorgehensweisen und Modellen.
- Hier betrachten wir nur das sogenannte *Wasserfallmodell*, das den Prozess der Software-Entwicklung in fünf Phasen aufgliedert.



- Klare Trennung der verschiedenen Phasen.
- Schwierigkeiten in einer Phase verzögern das Gesamtprojekt.
- Streng sequentielles Vorgehen ist in der Praxis kaum möglich.

- Analyse des Problembereichs
- Festlegung der (funktionalen und nicht funktionalen) Anforderungen an das Programm:
Was soll das Programm/System leisten?
- Festlegung von organisatorischen Richtlinien (Aufwandsabschätzung, Terminplanung. . .) und Rahmenbedingungen (z.B. vorhandene Soft- und Hardware)
- Skizze der Systemarchitektur

Ergebnis der Anforderungsanalyse ist ein Anforderungskatalog (Anforderungsspezifikation).

- Beschreibt die Art und Weise, in der die gestellten Aufgaben gelöst werden sollen:
Wie lösen wir das Problem?
- Festlegung der Systemarchitektur.
- Entwurf der einzelnen Systemkomponenten (Wahl von Datenrepräsentationen und Algorithmen).
- *Objektorientierter* Entwurf: Festlegung der Klassen und Methoden.

Ergebnis der Entwurfsphase ist eine Entwurfsbeschreibung (z.B. mittels eines UML-Diagramms).

Identifizierung von Klassen, Methoden und Assoziationen:

- Kandidaten für Klassen sind:
 - Personen bzw. Rollen (z.B. Student, Angestellter, Kunde, ...)
 - Organisationen (z.B. Firma, Abteilung, Uni, ...)
 - Gegenstände (z.B. Artikel, Flugzeug, Immobilien, ...)
 - begriffliche Konzepte (z.B. Bestellung, Vertrag, Vorlesung, ...)
- Möglichkeit: Durchsuchen des Anforderungskatalogs nach Substantiven, die Mengen bezeichnen
- Heuristik zum Auffinden von Methoden: Suche nach Verben
- Kandidaten für Assoziationen sind physische oder logische Verbindungen mit einer bestimmten Dauer, wie
 - konzeptionelle Verbindungen (arbeitet für, ist Kunde von, ...)
 - Besitz (hat, ist Teil von, ...)
 - (häufige) Zusammenarbeit von Objekten zur Lösung einer Aufgabe

Beispiel: Identifizierung von Klassen, Methoden und Assoziationen

- Geg: Programm, welches Rechnungen für Kunden ausdruckt. Die Rechnungen sollen die Einzelposten sowie jeweils den Gesamtpreis enthalten.
- Klassen: Rechnung, Kunden, Einzelposten
- Methoden: ausdrucken, Gesamtpreis berechnen
- Assoziationen: Rechnungen *enthalten* Einzelposten, Rechnungen *für* Kunden

Entwurf-Optimierung:

- Je weniger Beziehungen zwischen Klassen, desto unabhängiger kann man sie implementieren (hilft bei der Verteilung der Arbeit).
- Mögliche Schritte zur Optimierung:
 - ① Reduziere Beziehungen
 - ② Fasse gemeinsame Aspekte von Klassen zu Oberklassen zusammen
 - ③ Finde isolierte Inseln im UML-Diagramm
→ Arbeit organisieren, aufteilen

Wasserfallmodell: Implementierung, Test und Installation

- Implementierung: Codierung des Entwurfs in einer Programmiersprache, ggf. unter Wiederverwendung vorhandener Komponenten.
- Test:
 - Test der einzelnen Komponenten
 - Schrittweises Zusammenfügen einzelner Komponenten mit jeweiligem Integrationstest
 - Systemtest
 - Abnahmetest (mit "echten" Daten)
- Installation (und Wartung):
 - Installation des Systems
 - Fehlerbeseitigung nach Inbetriebnahme
 - Änderung und Erweiterung des Systems