

Strukturierter Entwurf

Lernziele

- Sie verstehen das Prinzip der Modularisierung.
- Sie können ein Anwendungsproblem in eine Modulstruktur umwandeln und mit Structured Chart graphisch darstellen.
- Sie können die Schnittstelle eines Moduls mit Hilfe einer Modulspezifikation spezifizieren.

Phasenmodell

- In der Designphase muss aus den Anforderungen der Analysephase die Softwarestruktur entwickelt werden.
- Eine gute Softwarestruktur ist die Voraussetzung für eine erweiterbare- und wartbare Software.

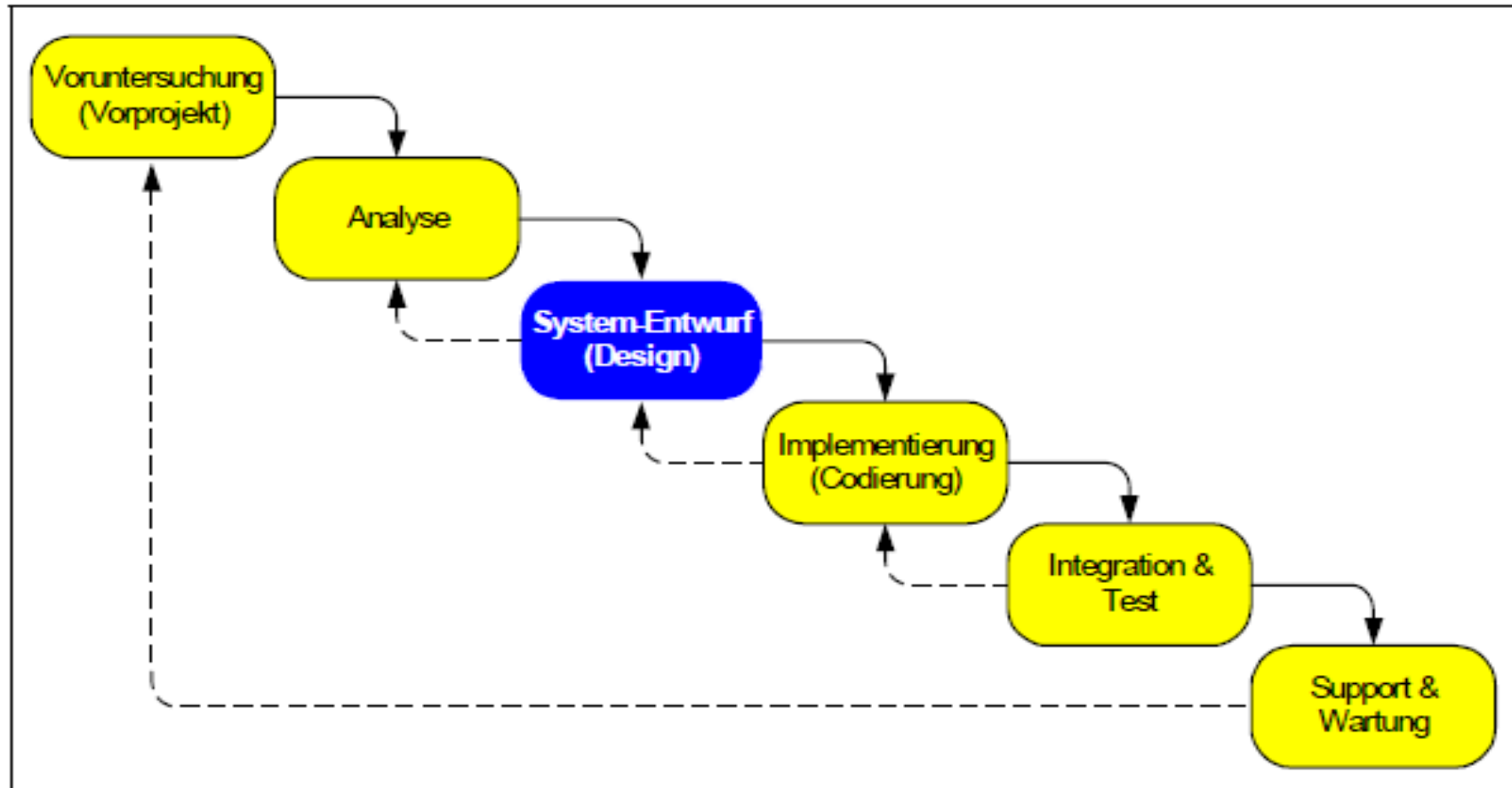
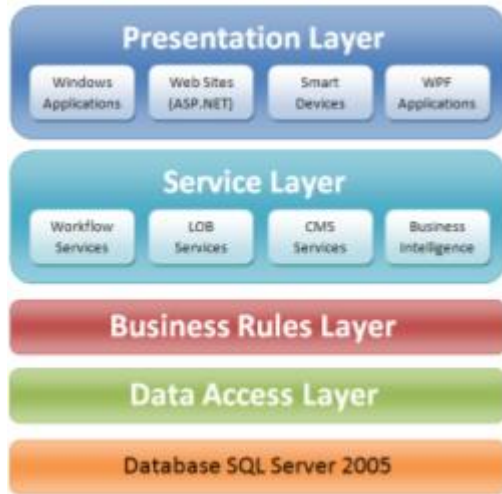


Abbildung 1: Wasserfallmodell

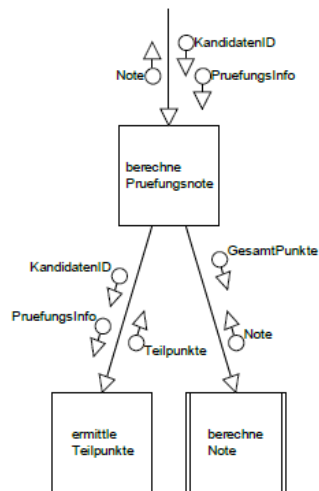
System-Entwurf

Grob-Entwurf, Architektur:



- Beschreibt Struktur und Organisation auf hohem Abstraktionsniveau (Architektur).
- Identifiziert die verschiedenen Komponenten.
- Beschreibt die Interaktion zwischen den Komponenten.

Fein-Entwurf (Detailed Design):



- Definiert die einzelnen Komponenten, so dass sie implementiert werden können.
- Beschreibt Schnittstellen.

Modularisierung

■ Prinzip der Modularisierung

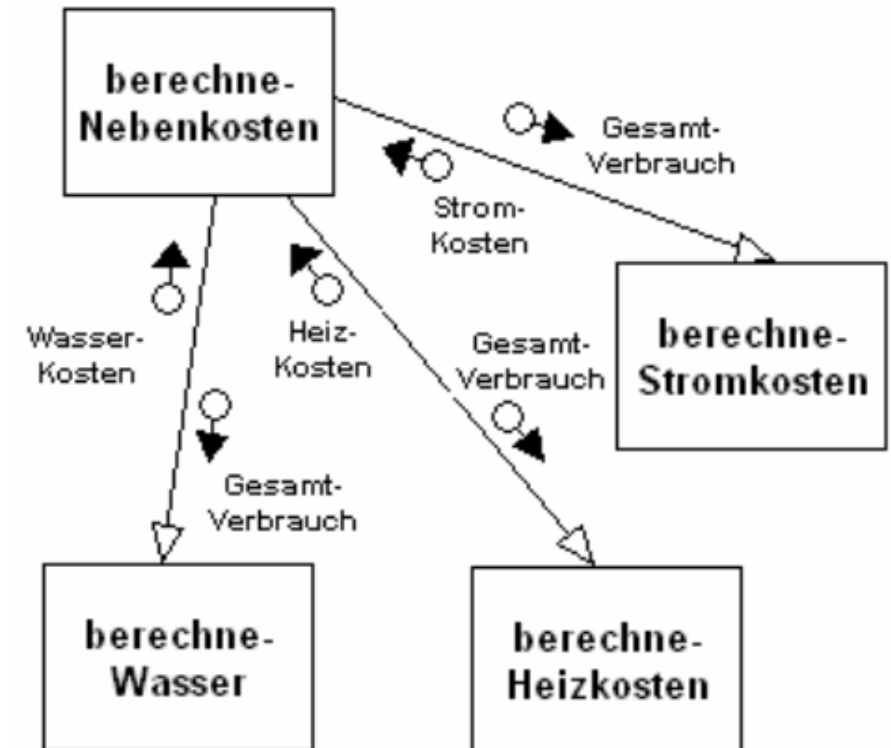
Ein Komplexsystem wird top-down in Modulen unterteilt.

Dabei werden folgende Ziele verfolgt:

- Jedes Modul löst nur einen genau definierten Teil des Gesamtproblems.
- Jedes Modul hat eine definierte Schnittstelle.

■ Ergebnisse:

- Hierarchische, modular aufgebaute Modulstruktur
- Modulspezifikation



Beispiel: Brüche addieren

■ Problemstellung:

Eine Programmkomponente soll zwei Brüche addieren und das Ergebnis als Bruch gekürzt zurückgeben. Das Problem soll mit einem Structured-Chart dargestellt werden.

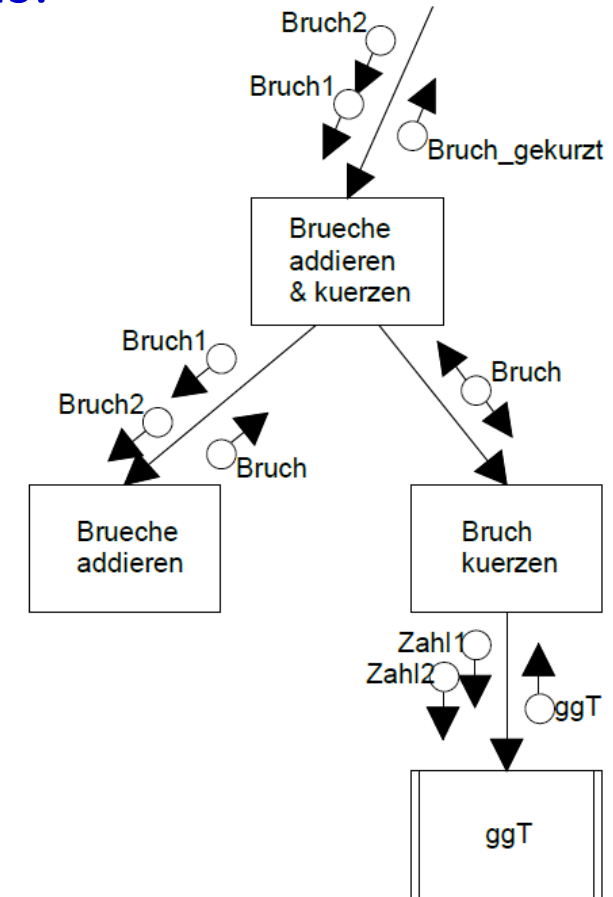
Eine erste grobe Modulaufteilung sieht folgende Module vor:

- „ggT“, Bibliotheksmodul für grösster gemeinsamer Teiler zweier ganzer Zahlen.

- „Brueche addieren“, addiert zwei Brüche nach dem Beispiel : $\frac{2}{3} + \frac{4}{5} = \frac{2 \cdot 5 + 3 \cdot 4}{15} = \frac{22}{15}$! ohne Kürzung!

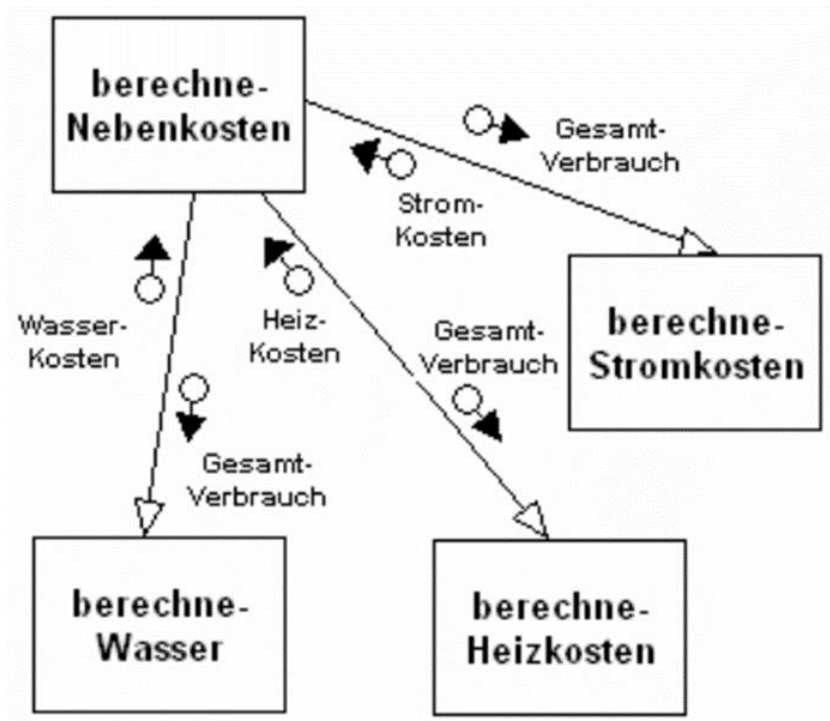
- „Bruch kuerzen“, kürzt einen Bruch z.B. : $\frac{9}{15} = \frac{3}{5}$

■ Ergebnis:



Modulen

- **Modul: eine klar abgegrenzte Komponente eines Systems**
 - Es hat einen Namen welcher ausdrückt, was das Modul macht.
 - Es ist über die definierte Schnittstelle aufrufbar (Daten von/zum aufrufenden Modul).
 - Das „innere“ wird verborgen (Geheimnisprinzip, Information-Hiding).

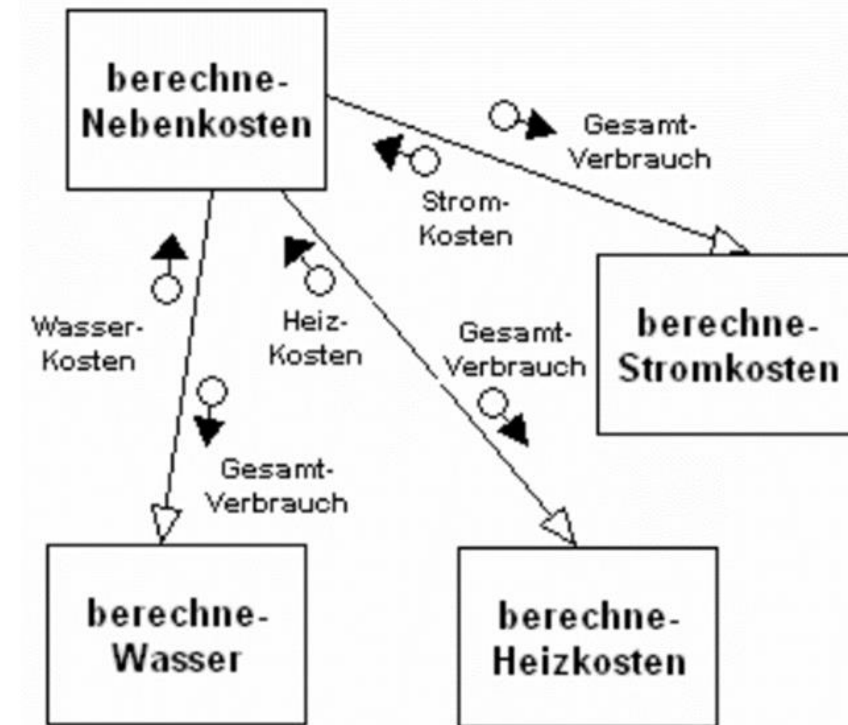


Structure Chart

Die Grafische Darstellung der Aufrufstruktur und des Datenflusses zwischen den Modulen

Ein Structure Chart zeigt also folgende Aspekte :

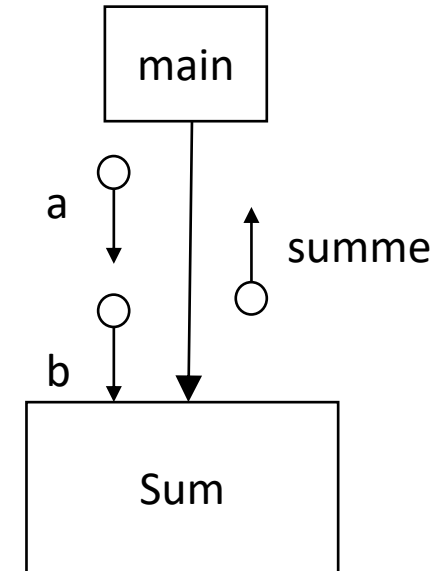
- **Hierarchie:** zeigt, welches Modul von anderen Modulen aufgerufen (verwendet) wird.
- **Modularität:** Zusammengehörigkeit von Modulen.
- **Kommunikation:** Übergabe von Daten- und Kontrollinformationen zwischen den Modulen
- **Abstraktion:** Nur die Schnittstellen und die Aufgabe der Module werden beschrieben.
- **Kopplung:** Zusammenhalt der Module (Datenfluss).



Structure Chart: Beispiele

Beispiel 1: Summe berechnen

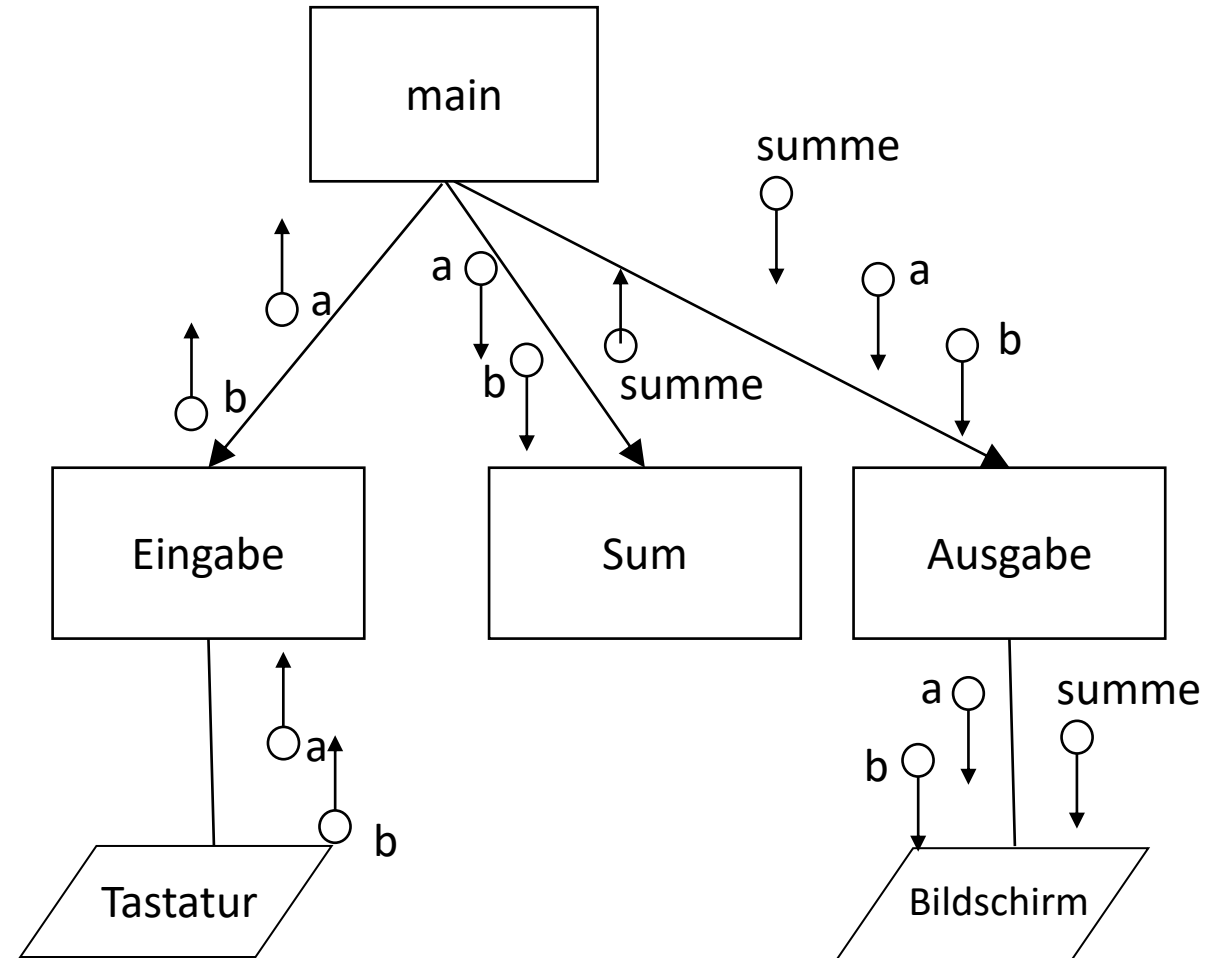
```
int Sum(int a, int b) {  
    return a + b;  
}  
  
int main()  
{  
    int a, b, summe;  
  
    printf("Geben Sie zwei ganze Zahlen ein (getrennt mit Leerstelle)>");  
    scanf_s("%d %d", &a, &b);  
    summe = Sum(a, b);  
    printf("Die Summe von %d und %d ist %d\n", a, b, summe);  
  
    return 0;  
}
```



Structure Chart: Beispiele

Beispiel 2: Vollständiges Structure Chart für die Berechnung von Summe mit Modulen:

- Eingabe aus der Tastatur
- Summe berechnen
- Ausgabe auf die Bildschirm



Übung zu Structure Chart: Schiefer Wurf

Mit Modularisierung und Structure Chart kann ein Programm hierarchisch in Modulen untergeteilt und graphisch dargestellt werden.

- 1) Studieren Sie zuerst die Theorie über die Modularisierung und die Notationen des Structure Charts im Skript (Kapitel 1 bis 5).
- 2) Studieren Sie die Aufgabenstellung [Schiefer Wurf](#).
- 3) Erstellen Sie die Modulstruktur als Structured-Chart unter der Annahmen:
 - Die Bibliotheksfunktionen Sinus bzw. Cosinus stehen Ihnen zur Verfügung.
 - Die Ein-/Ausgabe müssen nicht berücksichtigt werden.
- 4) Schreiben Sie die Modulspezifikation zu einem Modul (Skript Kapitel 6).

Modulspezifikation

- Structure Chart:
Modulstruktur
- Implementierung der Modulen:
genauere Spezifikationen der Module notwendig
- Modulspezifikation:
 - ✓ Schnittstelle (Input/Output)
 - ✓ die Aufgabenbeschreibung (Was, nicht wie?)

Modulspezifikation

Eine Modulspezifikation enthält folgende Abschnitte:

- 1) Modulname
- 2) Aufrufsyntax
- 3) Precondition
(was muss erfüllt sein, damit die Funktion korrekt arbeitet)
- 4) Postcondition
(welche Ergebnisse die Funktion unter bestimmten Bedingungen liefert)
- 5) Kurzbeschreibung der Funktion.
- 6) Beschreibung der Parameter
(Name, Datenflussrichtung, Format, Bedeutung)

Modulspezifikation

Modulspez „ggT“

Aufruf: ggT(Zahl1, Zahl2)

Precondition: --

Postcondition: WENN einer der beiden Zahlen = 0 ist

DANN

WENN beide Zahlen = 0 sind

DANN ggT = 1

SONST ggT = | Zahl1 + Zahl2 |

SONST

ggT = grösster gemeinsamer Teiler der Absolutwerte von Zahl1 und Zahl2

Kurzbeschreibung: Der ggT wird aus den Beträgen von Zahl1 & Zahl2 ermittelt

Parameter:

Name	i/o/u/r	Format	Beschreibung
Zahl1	i	Ganzzahl	die erste Zahl für die ggT-Bestimmung
Zahl2	i	Ganzzahl	die zweite Zahl für die ggT-Bestimmung
ggT	r	Ganzzahl	Ergebnis > 0

