Software Engineering I

Prof. Dr. Martin Glinz

Kapitel 2

Zielsetzung, Messung

Universität Zürich
Institut für Informatik

Zielsetzung – warum?

- Zielgerichtetes Arbeiten ist notwendig
- Ohne Zielsetzung: Qualität ist zufällig

- Software Engineering ➪ Ziele setzen
  ➪ Ziele verfolgen

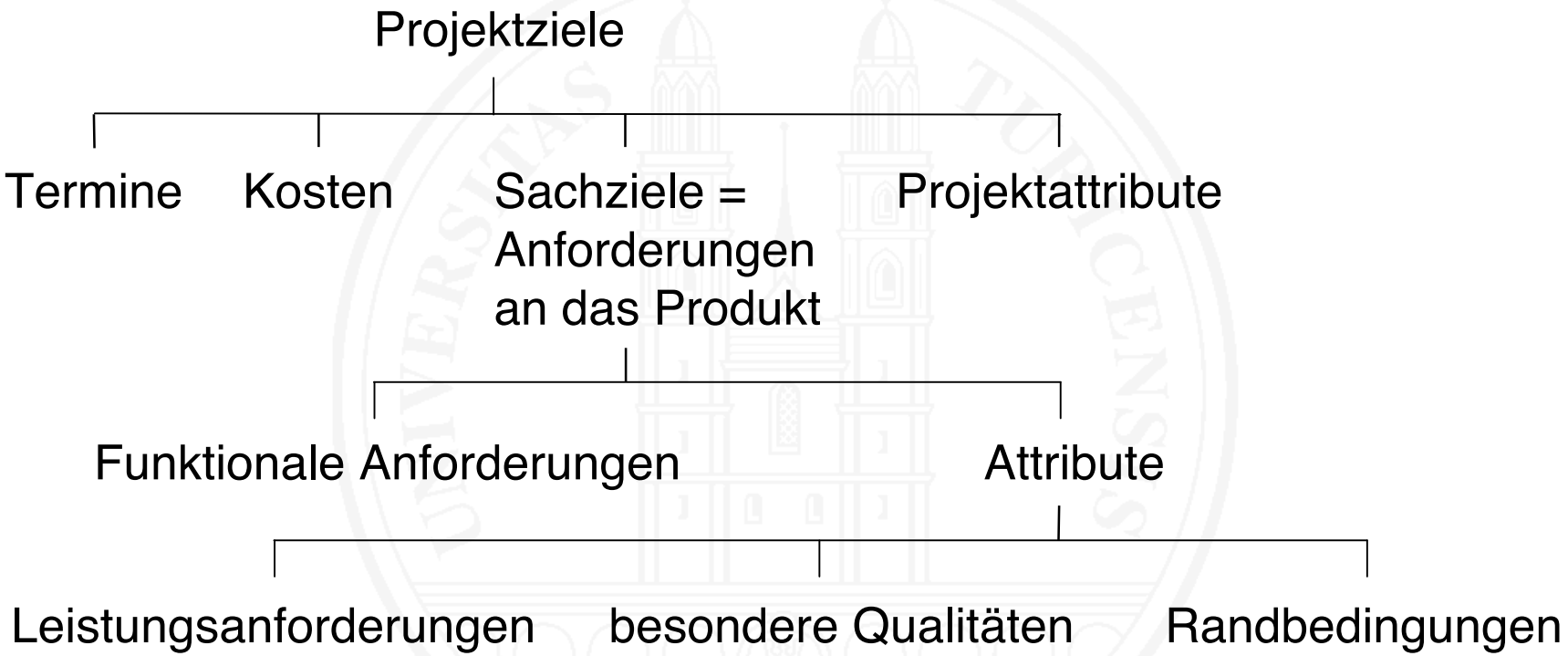
- Aber:
- Kein natürliches Ziel
- Ziele können
  - sich konkurrenzieren
  - voneinander abhängig sein
- Zielsetzung beeinflusst das Produkt erheblich
- Gegebenenfalls Prioritäten setzen

Das Experiement von Weinberg

Ziel	Qualität der Ergebnisse				
Optimiere...	Erstellungs-aufwand	Anzahl An-weisungen	Speicher-bedarf	Klarheit des Programms	Klarheit der Ausgaben
Erstellungs-aufwand	1	4	4	5	3
Anzahl An-weisungen	2 - 3	1	2	3	5
Speicher-bedarf	5	2	1	4	4
Klarheit des Programms	4	3	3	2	2
Klarheit der Ausgaben	2 - 3	5	5	1	1

1 beste Leistung 5 schlechteste Leistung

Klassifikation von Zielen



Zielverfolgung

Zielsetzung ist notwendig

- aber nicht hinreichend:
- langsame, stetige Abweichung ➪ massive Zielverfehlung

➪ Zielverfolgung ist erforderlich

➪ nur möglich, wenn Ziele messbar sind

Messung von Zielen

- Maße finden oder definieren
- Referenzwerte festlegen
  - Zielwert
  - Schwellwert

„Das System muss bei Übersäuerung schnell reagieren.“

➪ unbrauchbar, da nicht messbar

„Reaktionszeit < 0,1s von Erkennung pH-Wert < 4 bis Stellbefehl an Schließventil.“

➪ vernünftig, da messbar

Messen

Interessierende Merkmale eines Gegenstands quantitativ erfassbar machen

«Zutaten» zu einem Maß

- Menge von Gegenständen mit einem zu messenden Merkmal
- Skala
- Messverfahren (Abbildung der Gegenstände auf die Skala)
- Strukturähnlichkeit zwischen Merkmalsmenge und Skala

Beispiel: Menge von Stäben

- Zu messendes Merkmal: Länge
- Skala: Nicht negative reelle Zahlen
- Messverfahren: Vergleich mit Urmeter, Dreisatzrechnung
- Strukturähnlichkeit: Vergleichbarkeit, Additivität von Längen ↔ Zahlen

Mathematische Fundierung von Maßen

D Menge gleichartiger Gegenstände

M zu messendes Merkmal der Gegenstände aus D

M(d) Ausprägung des Merkmals M für den Gegenstand d aus D

$\mathcal{M}$ Menge aller Ausprägungen des Merkmals M

S Skala

Maß: Abbildung μ: D → S so dass $\mathcal{M}$ und S strukturähnlich sind

Strukturähnlich: Zu jeder Relation R auf $\mathcal{M}$ gibt es Relation R* auf S mit

(1) $R(M(d1), M(d2)) \Rightarrow R^*(\mu(d1), \mu(d2))$

(2) $R^*(\mu(d1), \mu(d2)) \Rightarrow$ R(M(d1), M(d2)) und die Aussage R(M(d1), M(d2)) ist sinnvoll interpretierbar.

(1) und (2) heißen auch *Repräsentations-* oder *Homomorphiebedingung*

Software-Maße

- Software-Maße werden auch Metriken genannt
- Software-Maße werden häufig definitorisch eingesetzt

- Beispiel für ein definitorisches Software-Maß:
  - Größe von Programmen
  - gemessen über die Anzahl der Codezeilen
  - auf der Skala der nicht negativen ganzen Zahlen

Skalentypen

Typ	erlaubt	Eigenschaften
Nominalskala	= ≠	Reine *Kategorisierung* von Werten Nur nicht-parametrische Statistik
Ordinalskala	= ≠ < >	Skalenwerte *geordnet* und vergleichbar Medianwert, sonst nur nicht-parametrische Statistik
Intervallskala	= ≠ < > Distanz	Werte geordnet, *Distanzen* bestimmbar Mittelwert, Standardabweichung
Verhältnisskala (auch Rational-skala genannt)	= ≠ < > Distanz, (+, –) Vielfaches, %	Werte geordnet und in der Regel *additiv** Skala hat *absoluten Nullpunkt* Übliche parametrische Statistik
Absolutskala	= ≠ < > Distanz, (+, –) Vielfaches, %	Skalenwerte sind *absolute Größen* Sonst wie Verhältnisskala

* Verhältnisskalen sind meistens additiv, müssen es aber nicht zwingend sein

Mini-Übung: Skalentypen

Mini-Übung 2.1 (Aufgabe 2.2 im Skript)

Geben Sie für die Skalentypen Nominalskala, Ordinalskala, Intervallskala, Verhältnisskala und Absolutskala je ein Beispiel aus dem täglichen Leben an.

Mini-Übung 2.2

a) Von welchem Skalentyp ist die in der Schweiz übliche Notenskala mit den Noten von 1 (schlechteste Note) bis 6 (beste Note)?

b) Zur Bewertung einer Klausur wird eine Punkteskala von 0 bis 100 Punkten verwendet. 0 bedeutet „nichts gelöst“ bzw. „alles falsch“, 100 bedeutet „alles richtig gelöst“.

   Von welchem Typ ist diese Skala?

Skalen für Maße im Software Engineering

Beispiele

- Nominalskala: Testergebnisskala mit den Werten {erfüllt, nicht erfüllt, nicht getestet}
- Ordinalskala: Eignungsskala mit den Werten { – –, –, o, +, ++ }
- Intervallskala: Datumskala für Zeit
- Verhältnisskala: Anzahl-Codezeilen-Skala für Programmgröße
- Absolutskala: Zählskala für die Anzahl gefundener Fehler in Programmen

Direkte und indirekte Maße

Direkte Maße

- Interessierende Merkmale in einfacher Weise direkt messbar
- Beispiele: Kosten, Durchlaufzeit

Indirekte Maße

- kein direktes Maß vorhanden oder Messung zu teuer
- messbare Indikatoren bestimmen
- Indikatoren müssen mit dem zu messenden Merkmal korreliert sein
- Indikatormaße bilden zusammen indirektes Maß für interessierendes Merkmal
- Beispiele: Portabilität, Benutzerfreundlichkeit

Beispiel: Messung von Portabilität

- Direktes Maß
  - Verhältnis Portierungsaufwand / Neuentwicklungsaufwand
  - Messung zu teuer

- Indirektes Maß mit drei Indikatoren

Indikator	**Skala**	**Messverfahren**	**Planwert**	**Schwellwert**
Anzahl Betriebssystem-Aufrufe / Anzahl Prozeduraufrufe	0-100%	Zählen im Code	5%	10%
Anteil hardware- oder betriebssystemabhängiger Module	0-100%	Zählen im Code	10%	15%
Anteil Nichtstandard Codezeilen	0-100%	Zählen, vgl. mit ISO-Standard	2%	5%

Maße für Ziele im Software Engineering

- Produktziele
  - Funktionserfüllung, Größe, Zuverlässigkeit, Benutzerfreundlichkeit, Wartbarkeit, ...
  - Meist nur indirekte Maße
  - Messung oft schwierig
- Projektziele
  - Aufwand, Durchlaufzeit (Dauer), Arbeitsfortschritt, Entwicklungskosten, Fehlerkosten, ...
  - Meist direkte Maße
  - Messung nicht schwierig, wenn richtige Basisdaten erhoben werden