Inhalt der Vorlesung

- Einführung
- Kommunikation
- Software Qualität
- Vorgehensmodelle
- Requirements Engineering
- Software Architektur und – Design
- Konfiguration Management

Software Qualität

Wesentliche Quelle zu diesem Kapitel:
Dirk W. Hoffmann: Software-Qualität, 2 Auflage, Springer Vieweg

Software Qualität

- Einführung
- Software Fehler
- Konstruktive Qualitätssicherung
- Software Test
- Statische Analyse

Software Qualität

- Immer komplexere Software
- Immer größere Durchdringung aller Lebensbereiche mit Software
- Sehr kurze Produktzyklen
- Immer höhere Erwartung der Nutzer

➔ Software Qualität kommt immer größere Bedeutung zu.

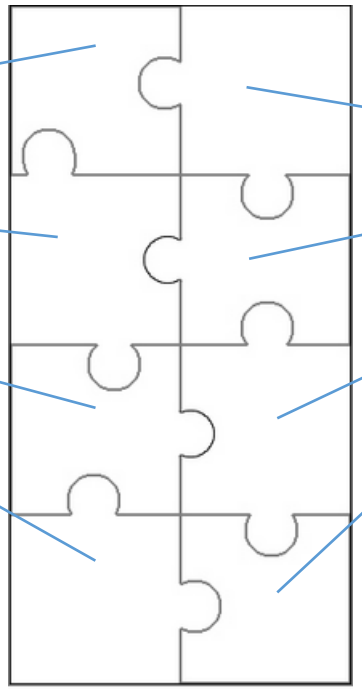
Definition nach DIN-ISO 9126:

Software Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software Produkts, die sich auf dessen Eignung beziehen, festgelegte Erfordernisse zu erfüllen.

Qualitätsmerkmale

- Funktionalität
- Zuverlässigkeit
- Effizienz
- Benutzbarkeit

kundenorientiert



- Portabilität
- Wartbarkeit
- Transparenz
- Testbarkeit

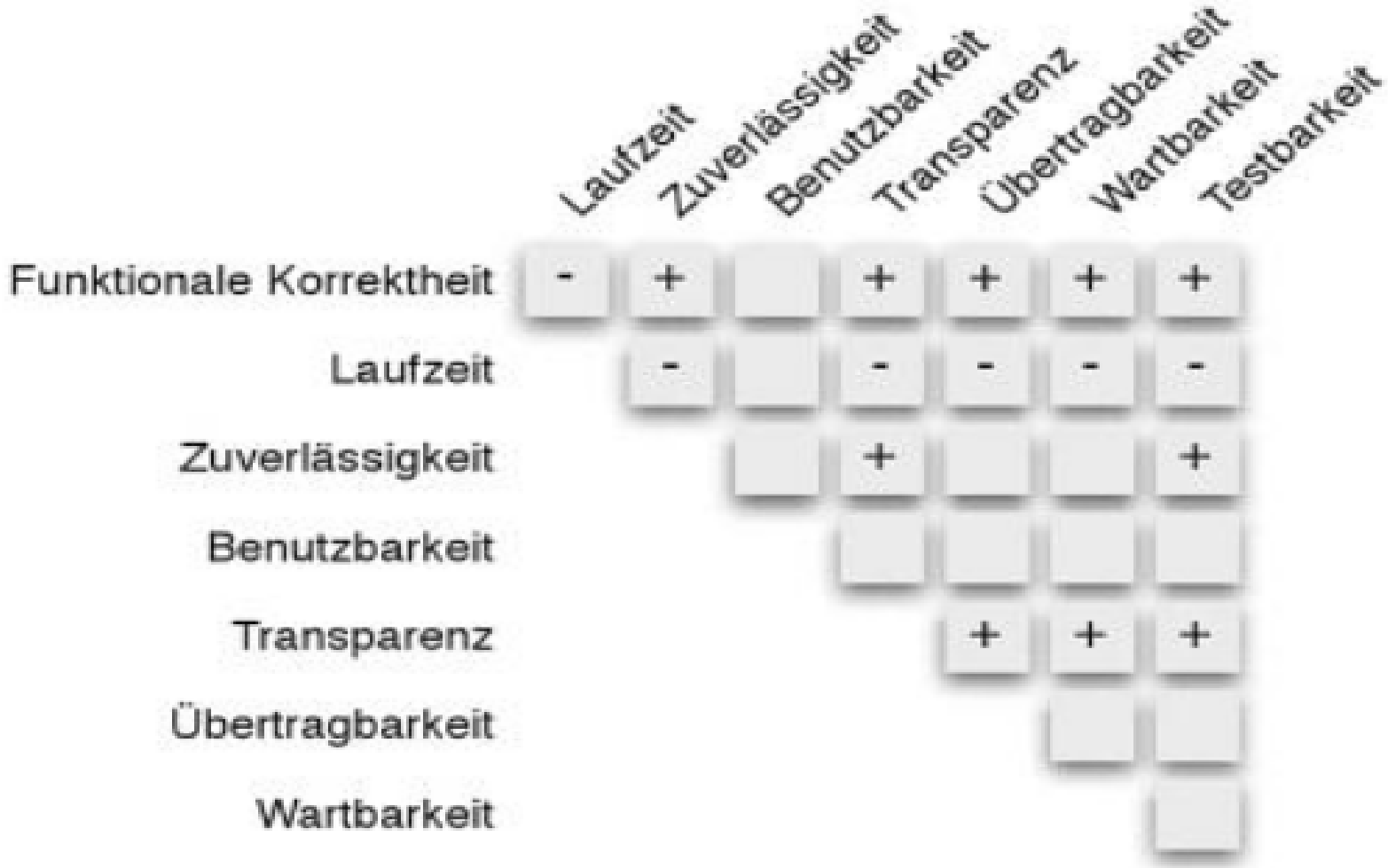
herstellerorientiert

Schwarz: Qualitätsmerkmale nach ISO/IEC 9126-1
Grau: Ergänzung nach D. Hoffmann.

Aufgabe: Korrelation von Qualitätskriterien:

	Funktionalität	Zuverlässigkeit	Effizienz	Benutzbarkeit	Übertragbarkeit	Wartbarkeit
Funktionalität	+	??	??	??	??	??
Zuverlässigkeit		+	??	??	??	??
Effizienz			+	??	??	??
Benutzbarkeit				+	??	??
Übertragbarkeit					+	??
Wartbarkeit						+

Korrelation von Qualitätskriterien



Besonderheiten der Korrelationen

Effizient (Laufzeit):

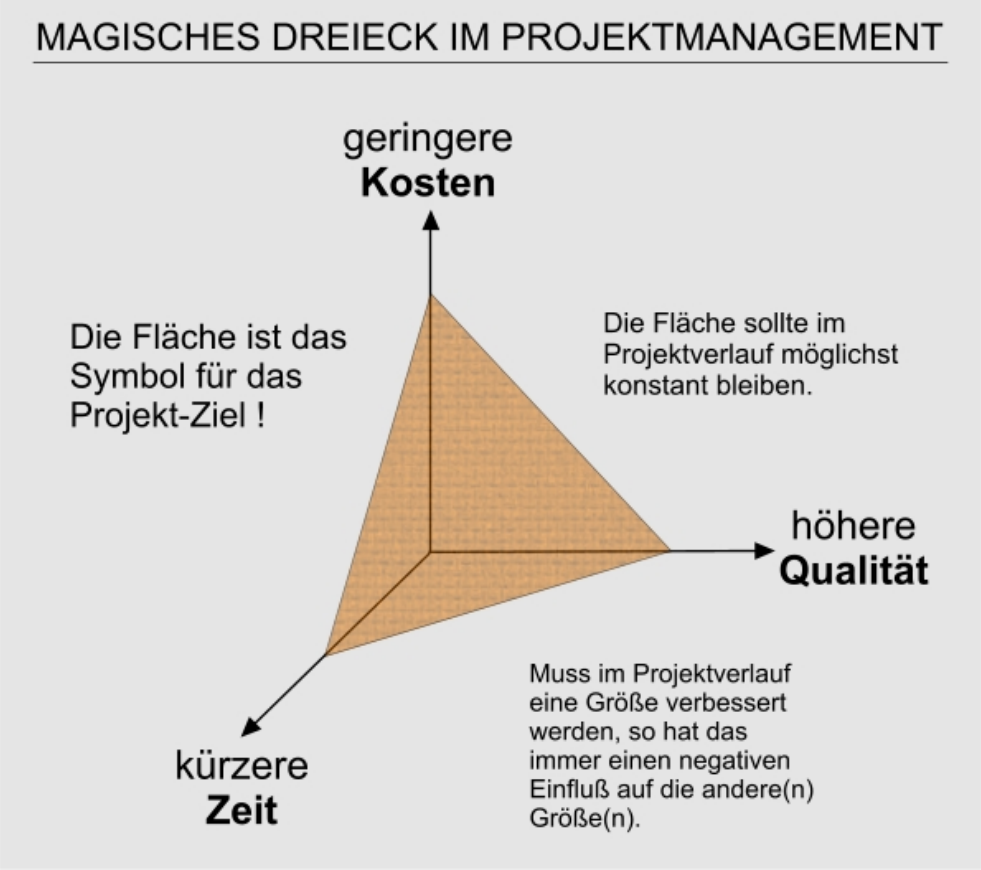
- Negative Korrelation mit fast allen anderen Qualitätsmerkmalen ➔mit Bedacht optimieren

Benutzbarkeit:

- Keine Korrelation mit anderen Merkmalen ➔ Benutzerfreundliche Programme sind möglich, ohne die anderen Merkmale zu beeinträchtigen.

Qualität im Spannungsfeld von Kosten und Zeit

Korrelierende Merkmale auch im Projektmanagement



Quelle:
http://www.seibit.de/www/softwareentwicklung/projektmanagement/index.php

Warum ist SW Qualität oft schlecht?

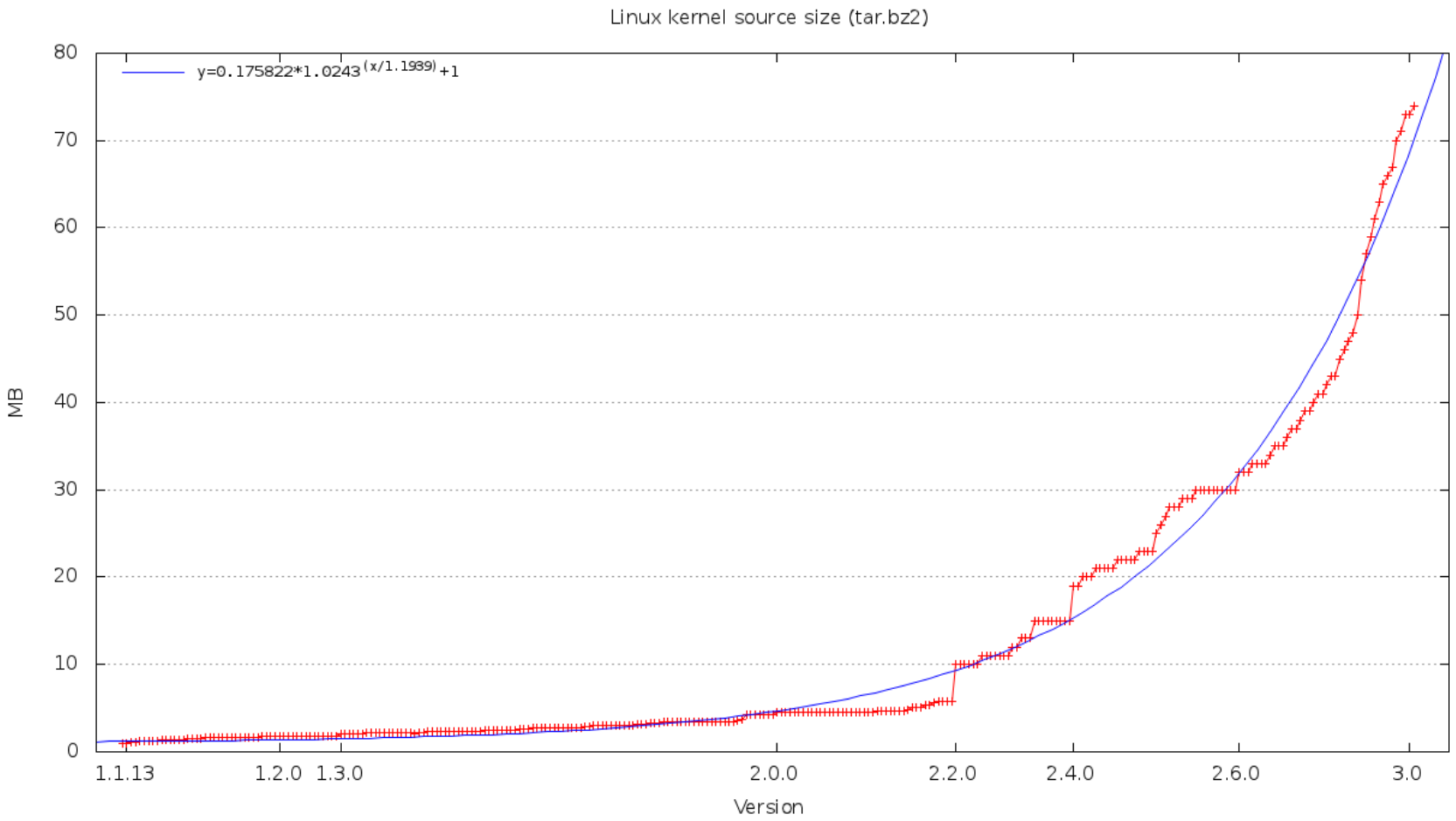
- Wachsende Komplexität
- Neue Anwendungsgebiete
- Vollständige Tests praktisch unmöglich
- Produktlebensdauer >> Projektdauer
- Erwartungshaltung/Fehlerakzeptanz des Kunden

Komplexitätszuwachs bei SW

Größenentwicklung des Linuxkernels (Quelle: http://www.pcwelt.de/ratgeber/Die-Kernel-Entwicklung-von-Linux-in-Zahlen-Entwicklung-in-Zahlen-9715212.html) :

Version	Jahr	Dateien	Patches	Codezeilen
0.01	1991	88	–	10.000
1.0	1994	563	k A	170.000
3.11	2013	44.017	10.893	17.407.037
3.12	2013	44.601	10.927	17.730.630
3.13	2014	44.985	12.127	17.934.674
3.14	2014	45.950	12.311	18.275.747
3.15	2014	46.795	13.722	18.636.331
3.16	2014	47.440	12.804	18.882.881
3.17	2014	47.505	12.354	18.868.140
3.18	2014	47.986	11.379	18.997.848
3.19	2015	48.424	12.617	19.130.604
4.0	2015	48.945	ca 10.000	19.312.370

Linux Kernel Source size



Quelle: https://www.bitblokes.de/2011/11/bald-100-mbyte-die-evolution-des-linux-kernels/

Einschub zur Testbarkeit

Frage: Wie testen Sie die folgende Funktion vollständig?

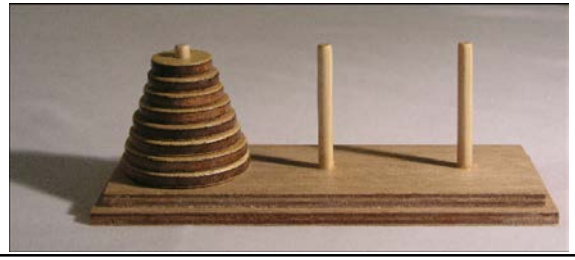
```
int machWas(int anzahl)
{
    int i=0;
    for(i=0; i<anzahl; i++)
    {
        printf("ich mach zum %iten Mal was\n", i+1);
    }
    return 0;
}
```

Weiteres Bsp zu Testbarkeit

```
int fakultaet(int x)
{
    int ret;
    if(x<0)
    {
        ret= 0;
    }
    else if( x==0)
    {
        ret=1;
    }
    else
    {
        ret = x*fakultaet(x-1);
    }
    return ret;
}
```

Weiteres Bsp zur Testbarbeit

```
void hanoi(int anzahlScheiben, char ausgangsstab, char zwischenstab, char zielstab)
{
    if(anzahlScheiben==1)
    {
        printf("Eine Scheibe von %c nach %c schieben.\n", ausgangsstab, zielstab);
    }
    else
    {
        hanoi(anzahlScheiben-1,ausgangsstab,zielstab,zwischenstab);
        hanoi(1,ausgangsstab,zwischenstab,zielstab);
        hanoi(anzahlScheiben-1,zwischenstab,ausgangsstab,zielstab);
    }
}
```



Was kann man gegen schlechte SW Qualität tun?

Software Qualität

Produktqualität

- Konstruktive Qualitätssicherung
  - Software Richtlinien
  - Typisierung
  - Vertragsbasierte Programmierung
  - Portabilität
  - Dokumentation
- Analytische Qualitätssicherung
  - Software Test
  - Statische Analyse
  - Software Verifikation

Prozessqualität

- Software Infrastruktur
  - Konfigurationsmanagement
  - Build Automatisierung
  - Test-Automatisierung
  - Defekt Management
- Management Prozesse
  - Vorgehensmodelle
  - Reifegradmodelle

Was kann man dagegen tun?

Software Qualität

Produktqualität

Thema jetzt

- Konstruktive Qualitätssicherung
  - Software Richtlinien
  - Typisierung
  - Vertragsbasierte Programmierung
  - Portabilität
  - Dokumentation
- Analytische Qualitätssicherung
  - Software Test
  - Statische Analyse
  - Software Verifikation

Prozessqualität

später

- Software Infrastruktur
  - Konfigurationsmanagement
  - Build Automatisierung
  - Test-Automatisierung
  - Defekt Management
- Management Prozesse
  - Vorgehensmodelle
  - Reifegradmodelle