

Inhalt der Vorlesung

- Einführung
- Kommunikation
- Konfiguration Management

- Software Qualität



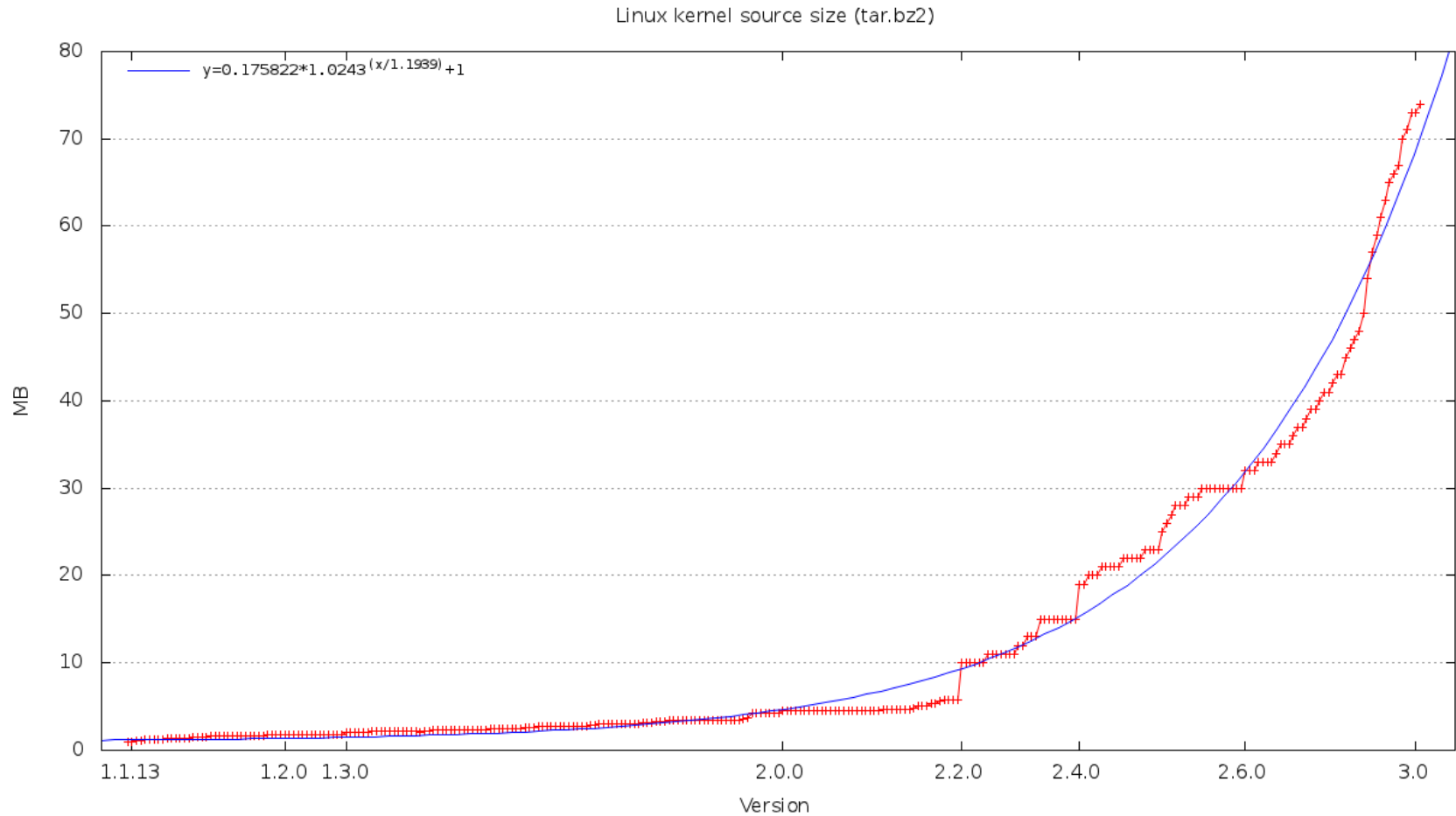
- Einführung
 - Software Fehler
 - Konstruktive Qualitätssicherung
 - Software Tests
 - Statische Analyse
- Vorgehensmodelle

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

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

➔ Software Qualität kommt immer größere Bedeutung zu.

Linux Kernel Source size



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

Was ist Software Qualität eigentlich?

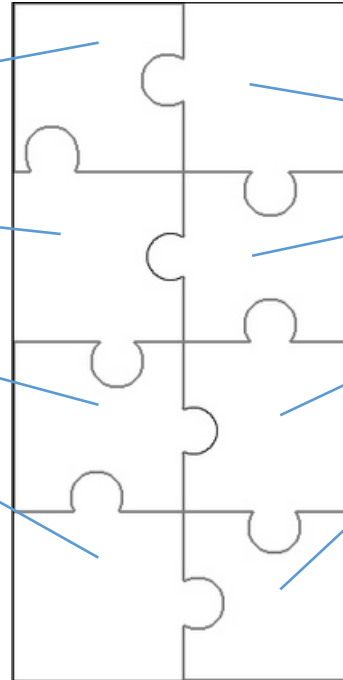
→ Definition nach DIN-ISO 9126:

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.

Korrelation von Qualitätskriterien:

	Funktionalität	Zuverlässigkeit	Effizienz	Benutzbarkeit	Übertragbarkeit	Wartbarkeit
Funktionalität	+	+	-		+	+
Zuverlässigkeit		+	-			+
Effizienz			+		-	-
Benutzbarkeit				+		
Übertragbarkeit					+	
Wartbarkeit						+

Effizienz (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>

Frage: Wie testen Sie die folgende Funktion vollständig?

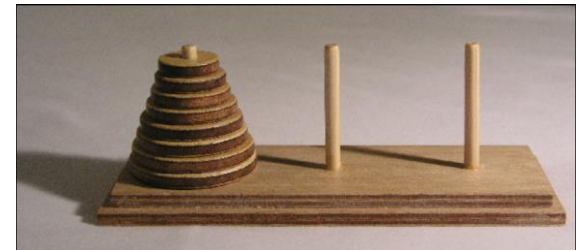
```
int machWas(int anzahl)
{
    int i = 0;
    for(i = 0; i < anzahl; i++)
    {
        printf("ich mach zum %iten Mal was\n", i+1);
    }
    return 0;
}
```

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 Testbarkeit

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

Thema jetzt

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