



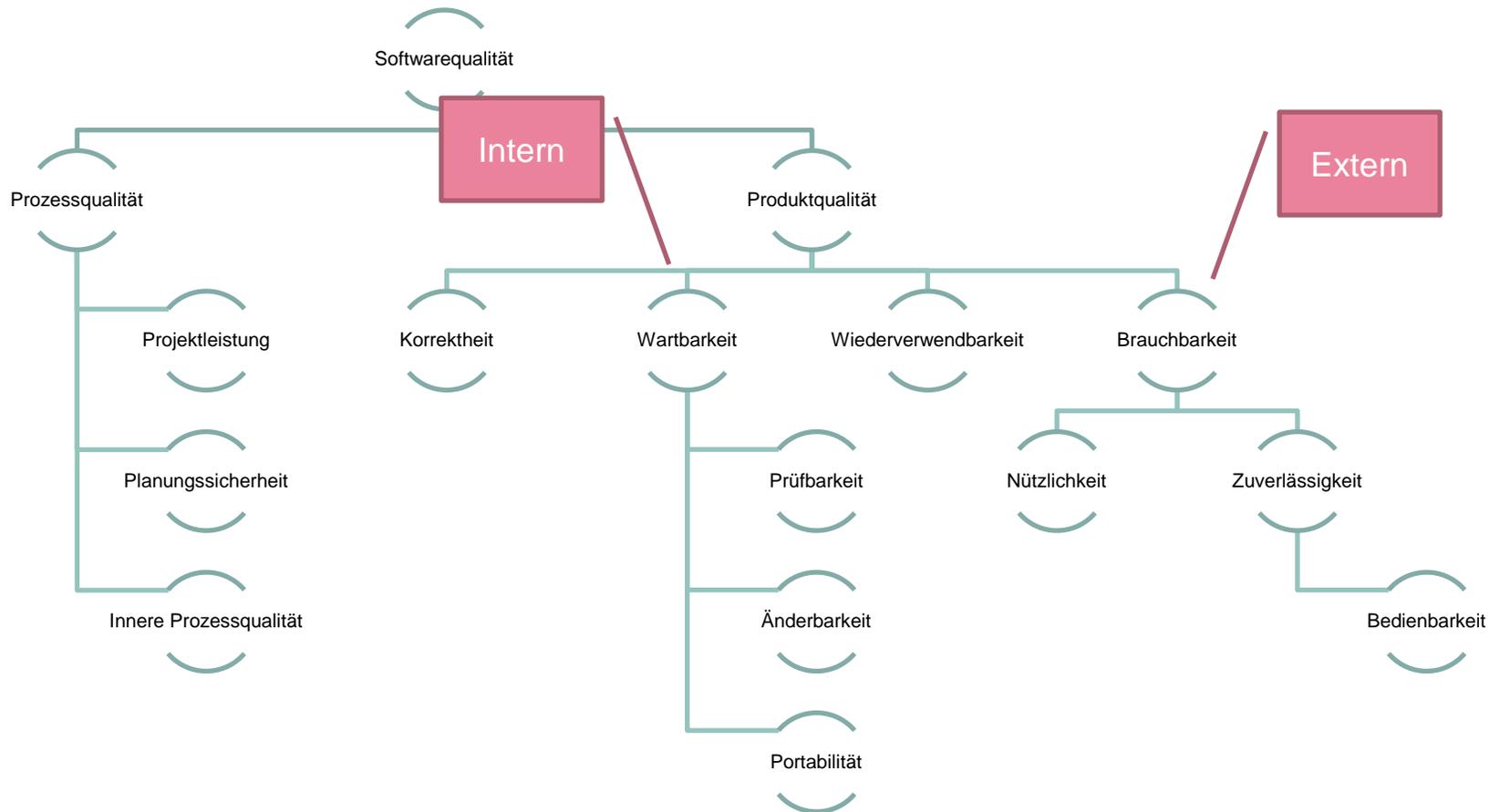
University
of Basel

Software Engineering

Marcel Lüthi, Universität Basel

Qualitätsmerkmale

Taxonomie



Interne vs. Externe Qualitätsmerkmale

- **Intern:** Sieht das Entwicklungsteam
- **Extern:** Sieht Benutzer*in

Beispiel 1: Korrektheit

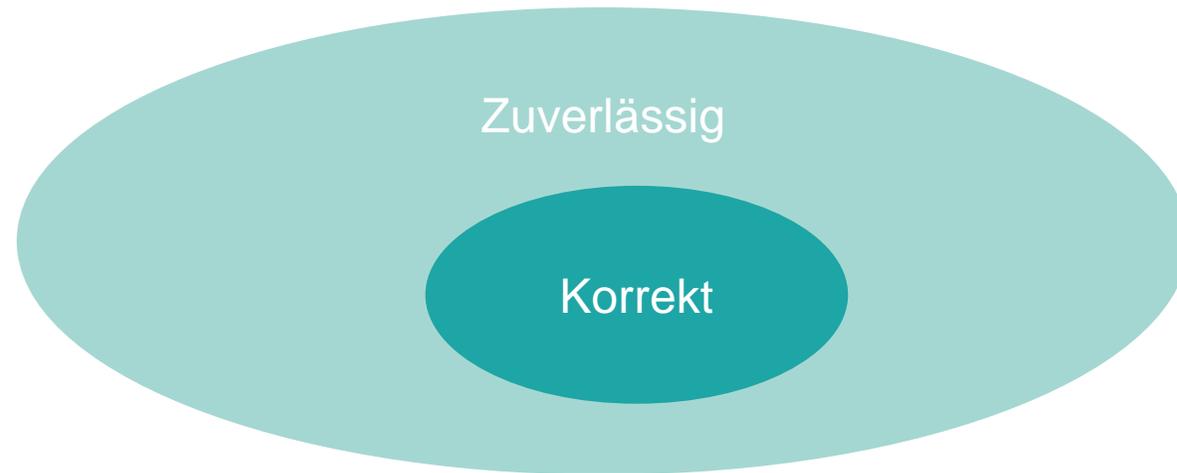
Software ist korrekt, wenn sie die spezifizierten funktionalen Anforderungen erfüllt

- Erfordert spezifizierte Anforderungen
 - Korrektheit ist absolut
 - Programm ist nicht "ein bisschen falsch"
 - Formale Spezifikationen erlauben Korrektheitsbeweise
-

Beispiel 2: Zuverlässigkeit

Wahrscheinlichkeit, dass in einem bestimmten Zeitintervall kein Fehler auftritt

Intuitiv: Der Benutzer kann sich auf die Software verlassen



Beispiel: Zuverlässig / Korrekt

Ist dieses Programm korrekt?

Ist es zuverlässig?

```
double plus(double a, double b) {  
    double sum = a + b;  
    Random r = new Random();  
    if (Math.abs(r.nextGaussian()) < 1000) {  
        return sum;  
    } else {  
        return sum + 1;  
    }  
}
```

Robustheit

Software verhält sich auch unter unvorhergesehenen Umständen sinnvoll

Beispiele:

- Hardwareausfall
- Netzwerkausfall
- Falscher Input

Schwierig formal zu spezifizieren

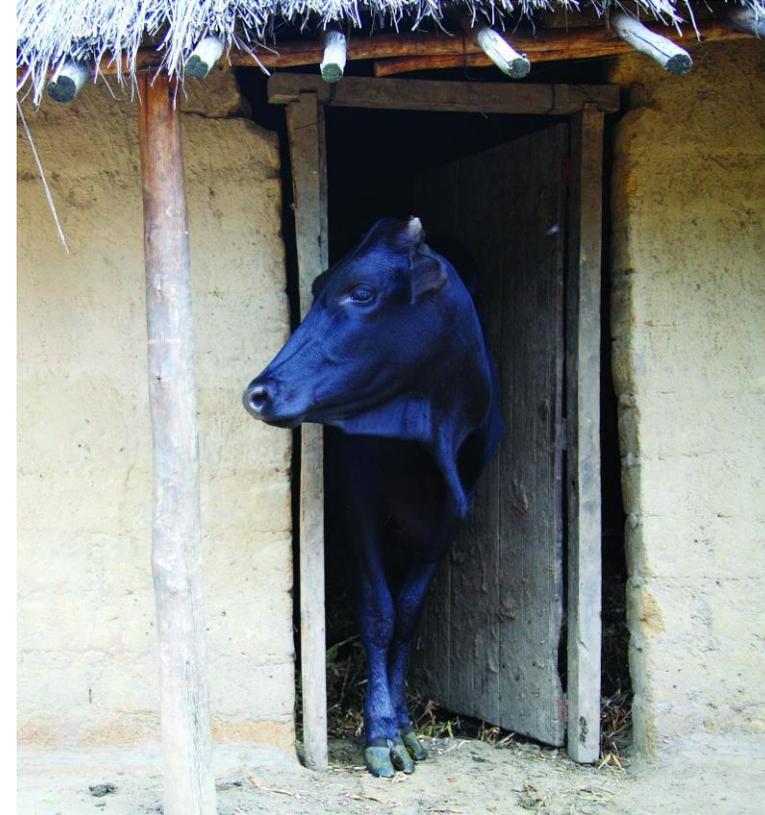


Korrektheit versus Robustheit
Anforderungen spezifiziert → Korrektheit
Anforderung nicht spezifiziert → Robustheit

Wartbarkeit



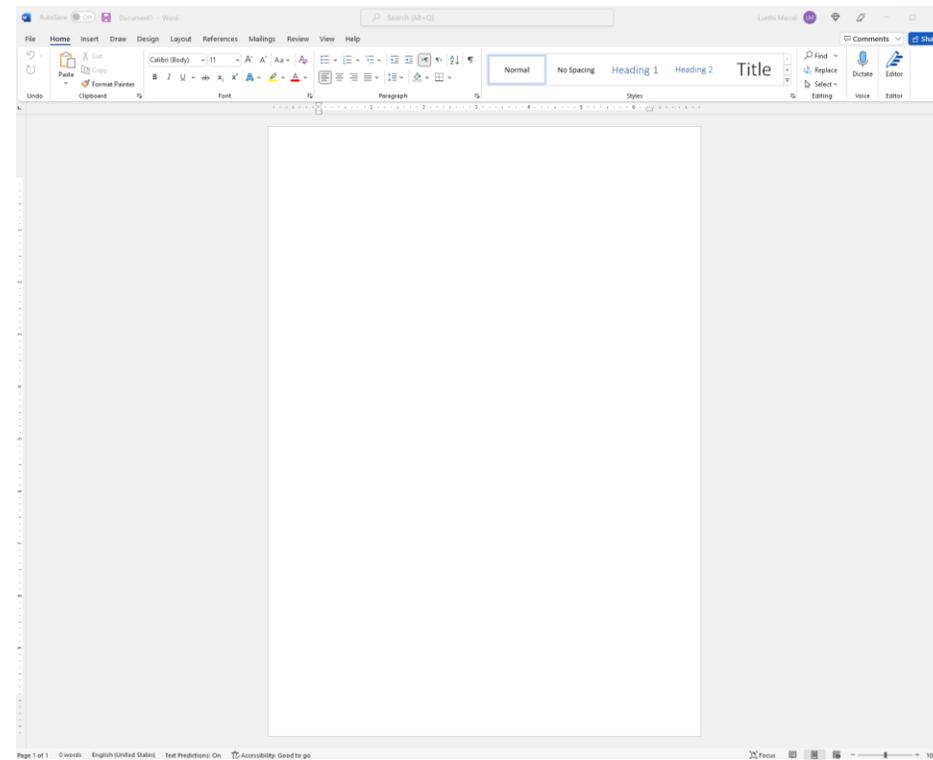
Software ist nicht das



sondern das!

Wartbarkeit

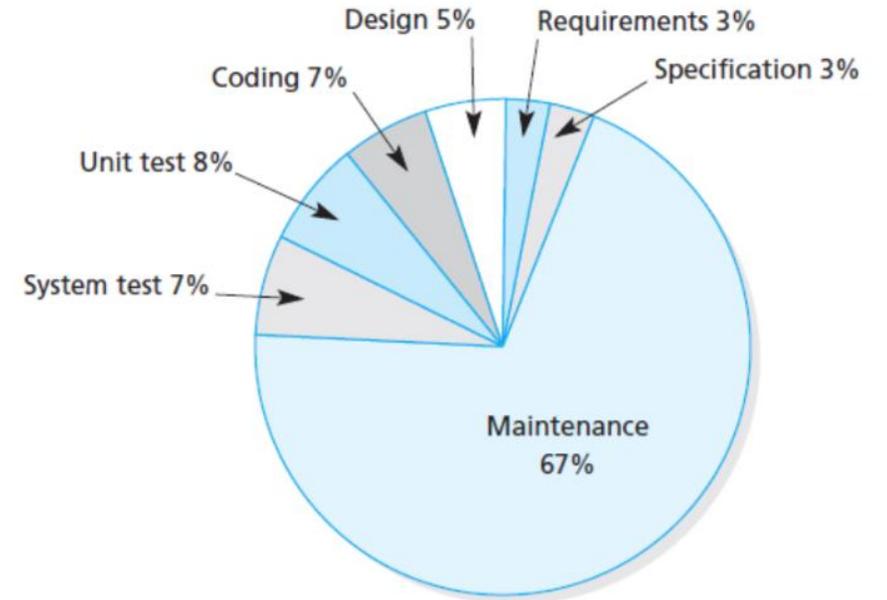
Wie einfach ist es die Software weiterzuentwickeln und an neue Gegebenheiten anzupassen?



Wartbarkeit

Gründe von Software-Änderung:

- Neue Hardware
- Neue Betriebssysteme
- Neue Anforderungen an Software
 - Gesetze
 - Marktdruck
- Fehlerbehebung
- Verbesserung (Performance / Wartbarkeit, ...)



Quelle: Software Engineering for Students, D. Bell

Grossteil der Kosten fällt nach der Entwicklung an!

Verständlichkeit

Wie einfach ist es den Code zu verstehen?

```
#include
int l;int main(int o,char **O,
int I){char c,*D=O[1];if(o>0){
for(l=0;D[l
];D[l
++]-=10){D[l++]-=120;D[l]-=
110;while(!main(0,O,l))D[l]
+=20;putchar((D[l]+1032)
/20);}putchar(10);}else{
c=o+(D[I]+82)%10-(I>1/2)*
(D[I-1+I]+72)/10-9;D[I]+=I<0?0
:!(o=main(c/10,O,I-1))*((c+999)
)%10-(D[I]+92)%10);}return o;}
```

Winner 2001, Best Short Program, International Obfuscated C code Contest

Wiederverwendbarkeit

Wie einfach ist es die Software in anderem Kontext wiederzuverwenden?

- Komponentenbasierte Softwareentwicklung
 - Kosten für Verifikation, Wartung, etc. pro Komponente und nicht pro Produkt
-

Portabilität

Grad der Plattformunabhängigkeit einer Software

Plattformen:

- Verschiedene Betriebssysteme
 - Verschiedene Browser
 - Verschiedene Hardware
-

Taxonomie

