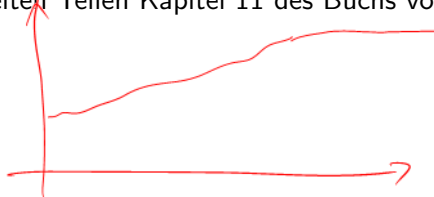


- Notwendigkeit und Grenzen des Tests verstehen
- Arten und Varianten des Software-Tests kennen
- eine geeignete Test-Strategie auswählen können
- Testpläne erstellen können
- Tests durchführen können

N.B.: Diese Darstellung folgt in weiten Teilen Kapitel 11 des Buchs von Brügg und Dutoit (2004).



Ein korrektes Programm?

```
class Kalender {
    public static class MonatUngueltig extends Exception {};
    public static class JahrUngueltig extends Exception {};
    public static boolean istSchaltJahr(int jahr)
        {return (jahr % 4) == 0;}
    public static int TageProMonat (int monat, int jahr)
        throws MonatUngueltig, JahrUngueltig {
        int anzTage;
        if (jahr < 1) { throw new JahrUngueltig(); }
        if (monat in {1, 3, 5, 7, 8, 10, 12}) { anzTage = 32; }
        else if (monat in {4, 6, 9, 11}) { anzTage = 30; }
        else if (monat == 2) {
            if (istSchaltJahr (jahr)) anzTage = 29;
            else anzTage = 28;
        } else throw new MonatUngueltig();
        return anzTage;
    }
}
```

Definition

Zuverlässigkeit: Maß für den Erfolg, inwieweit das beobachtete Verhalten mit dem spezifizierten übereinstimmt.

Software-Zuverlässigkeit: Wahrscheinlichkeit, dass ein Software-System während einer festgelegten Zeit unter festgelegten Bedingungen keinen Systemausfall verursachen wird (IEEE Std. 982-1989 1989).

Definition

Störfall (Ausfall, Failure): jegliche Abweichung des beobachteten Verhaltens vom spezifizierten.

Im Nicht-Schaltjahr 200 wird für den Monat Februar 29 ausgegeben.

Definition

Störfall (Ausfall, Failure): jegliche Abweichung des beobachteten Verhaltens vom spezifizierten.

Im Nicht-Schaltjahr 200 wird für den Monat Februar 29 ausgegeben.

Fehlerhafter Zustand (Error): Zustand, in dem ein Weiterlaufen des Systems zu einem Störfall führen würde.

Die Funktion `istSchaltjahr(200)` liefert `true`.

Definition

Störfall (Ausfall, Failure): jegliche Abweichung des beobachteten Verhaltens vom spezifizierten.

Im Nicht-Schaltjahr 200 wird für den Monat Februar 29 ausgegeben.

Fehlerhafter Zustand (Error): Zustand, in dem ein Weiterlaufen des Systems zu einem Störfall führen würde.

Die Funktion `istSchaltjahr(200)` liefert `true`.

Fehler (Defekt, Fault): mechanische oder algorithmische Ursache eines fehlerhaften Zustands.

Die Prüfung in `istSchaltjahr` ist falsch.

Definition

Test: systematischer Versuch, in der implementierten Software Defekte zu finden.

Erfolgreicher (positiver) Test: Test, der Defekt aufgedeckt hat.

Erfolgloser (negativer) Test: Test, der keinen Defekt aufgedeckt hat.

Definition

Test: systematischer Versuch, in der implementierten Software Defekte zu finden.

Erfolgreicher (positiver) Test: Test, der Defekt aufgedeckt hat.

Erfolgloser (negativer) Test: Test, der keinen Defekt aufgedeckt hat.

- Tests sind Experimente zur Falsifikation der Hypothese “System ist korrekt”.
- Aus negativem Test folgt noch lange nicht, dass kein Defekt vorhanden ist.

Tests sind nur *ein* Mittel, die Zuverlässigkeit zu steigern:

- Fehlervermeidung (z.B. Entwicklungsmethoden, Verifikation, statische Analyse)
- Fehlerentdeckung (z.B. Tests, assert, "Quality-Feedback-Agent", Flugschreiber)
- Fehlertoleranz: Behandlung von Fehlern zur Laufzeit, um Programm fortzusetzen

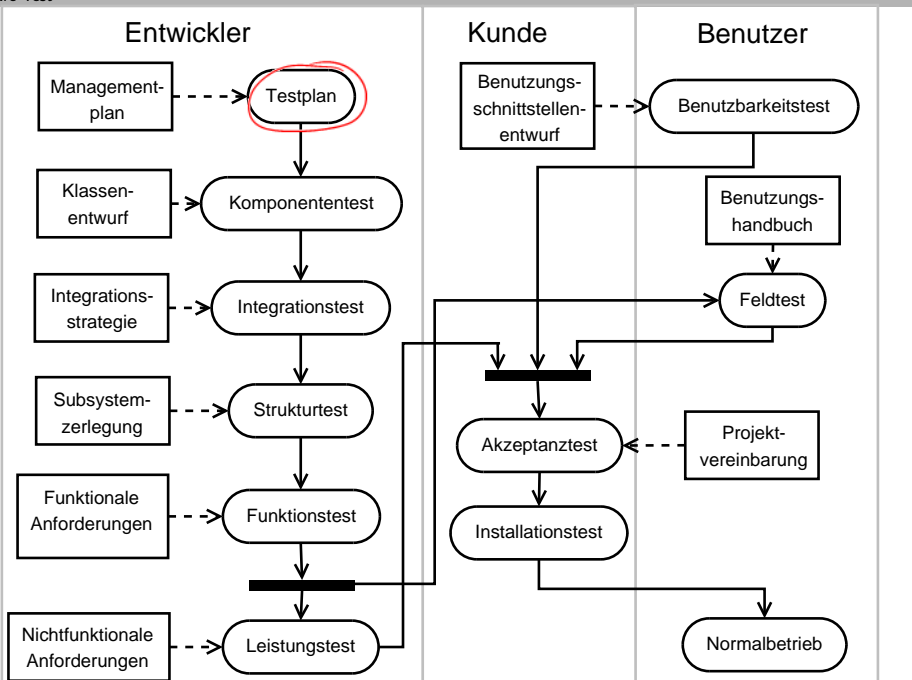
h-version - Programmierung

- Testen wird häufig (aber zu unrecht) als niedrigere Arbeit angesehen
- Frischlinge werden zu Testern
- Tester brauchen jedoch ein umfassendes Systemverständnis (Anforderungen, Entwurf, Implementierung)
 - Tester brauchen darüber hinaus, Wissen über Prüftechniken

- Testen wird häufig (aber zu unrecht) als niedrigere Arbeit angesehen
- Frischlinge werden zu Testern
- Tester brauchen jedoch ein umfassendes Systemverständnis (Anforderungen, Entwurf, Implementierung)
- Tester brauchen darüber hinaus, Wissen über Prüftechniken
- Autoren haben eine Lesart der Spezifikation verinnerlicht
- Denkfehler in der Implementierung werden sich bei Erstellung von Testfällen wiederholen
- Tester sollten nicht gleichzeitig Autoren sein

Soziologie des Testens

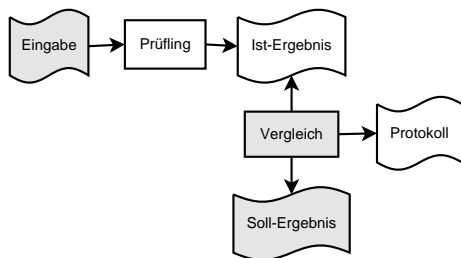
- Testen wird häufig (aber zu unrecht) als niedrigere Arbeit angesehen
- Frischlinge werden zu Testern
- Tester brauchen jedoch ein umfassendes Systemverständnis (Anforderungen, Entwurf, Implementierung)
- Tester brauchen darüber hinaus, Wissen über Prüftechniken
- Autoren haben eine Lesart der Spezifikation verinnerlicht
- Denkfehler in der Implementierung werden sich bei Erstellung von Testfällen wiederholen
- Tester sollten nicht gleichzeitig Autoren sein
- Tester versuchen, Fehler zu finden
- das Produkt wird kritisiert, dann fühlen sich Autoren selbst kritisiert
- Egoless-Programming (eine schöne Illusion)



Testfall

Ein Testfall besteht aus:

- Name
- Ort: vollständiger Pfadname des lauffähigen Testprogramms
- Eingabe: Eingabedaten oder Befehle
- Orakel: erwartete Testergebnisse, die mit den aktuellen Testergebnissen des Testlaufs verglichen werden
- Protokoll: gesamte Ausgabe, die durch Test erzeugt wird



Komponententest: Teststümpfe und -treiber

vom Tester
geschrieben

