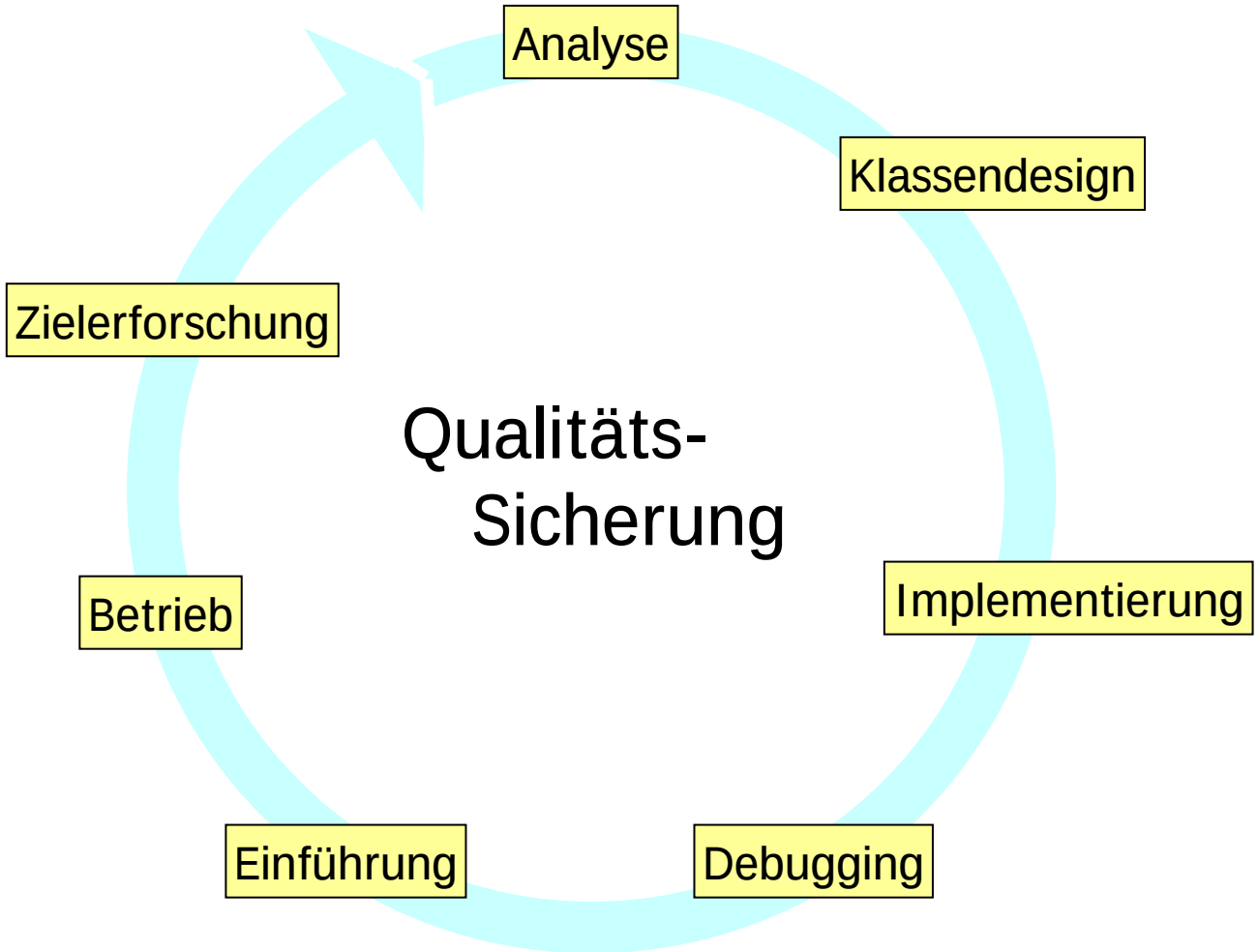
Qualitätssicherung von Software

Prof. Dr. Holger Schlingloff

Humboldt-Universität zu Berlin
und
Fraunhofer FIRST

Kapitel 2. Testverfahren

OO-Entwicklungszyklus



Besonderheiten im Entwurf

- **Hiding:** starke Kapselung von Klassen gegenüber äußeren Zugriffen
- **Inheritance:** Übernahme von großen Teilen der Funktionalität anderer Klassen durch Vererbung, ggf. sogar Mehrfachvererbung
- **Polymorphie:** verschiedene Bedeutung der selben Operation für verschiedene Objekte, dynamische Bindungen
- **abstrakte und generische Klassen:** Schablonen, die selbst keine ausführbaren Methoden enthalten

Schwierigkeiten

- starke Zustandsabhängigkeit: Operationen sind über den Zustand der jeweiligen Objektattribute verknüpft, sehr viele mögliche Zustände
- Verlagerung der Komplexität von strukturellen Einheiten in deren Beziehungsgeflecht (Methodenaufrufe als Botschaften)
- viele Entscheidungen werden erst zur Laufzeit getroffen (Objekterzeugung, Methodenbindung usw.)
- weitere Freiheitsgrade durch andere OO-Konstrukte

→ komplexere Beziehung zwischen Programmtext und Programmverhalten

→ Ableitung von Testfällen aus dem Programmtext schwierig

Stimmen zum Test objektorientierter SW

- Grady Booch

  "... the use of object-oriented design doesn't change any basic testing principles; what does change is the granularity of the units tested."

  Booch, G. Object Oriented Design With Applications. Benjamin/Cummings, 1991, S. 212

- Boris Beizer

  "... it costs a lot more to test OO-software then to test ordinary software - perhaps four or five times as much ... Inheritance, dynamic binding, and polymorphism creates testing problems that might exact a testing cost so high that it obviates the advantages."

  Beizer, B. Testing Technology - The growing gap. American Programmer, Vol. 7, No. 4, 1994, S. 3-12

Objektorientierter Testprozess

- Klassentest *(intra-class – testing)*
  - (normale) Klasse als kleinste sinnvoll testbare Einheit, Test aller einzelnen Methoden
  - Kombination von zustands- und funktionsorientiertem Test
- Erweiterungen
  - Vererbung, Polymorphie, generische Klassen
- Integrationstest *(inter-class – testing)*
  - Test der Kommunikation zwischen einzelnen Klassen
- Systemtest, Abnahmetest
  - Black-Box-Sicht, Benutzungsschnittstelle

siehe auch Spillner, http://www.fbe.hs-bremen.de/spillner/HHOOTest/HHOOTest.pdf

Testen von Klassen

- Erzeugung eines (instrumentierten) Objektes der zu testenden Klasse
  - Aufruf der Konstruktoren, Besetzung der Attribute
- Überprüfung der einzelnen Methoden
  - zunächst die nicht zustandsverändernden, dann die Datenfeld-manipulierenden
  - Äquivalenzklassenbildung bzw. Grenzwertanalyse bezüglich der Objekt- und Methodenparameter
  - Vergleich des erreichten Objektzustands mit dem Sollzustand nach jeder Methodenausführung
- Test von Folgen von Methodenaufrufen
  - Beachtung der Modalität der Methoden (nächste Folie)

Modalität der Methoden

- Nichtmodale Klassen
  - Methoden sind unabhängig vom Zustand aufrufbar
  - z.B. getAccountValue(...), setAccountValue(...)
- Unimodale Klassen
  - zulässige und unzulässige Reihenfolgen der Methodenaufrufe
  - z.B. getSensorReading(...), setActuatorOutput(...)
- Quasimodale Klassen
  - Zulässigkeit eines Methodenaufrufs abhängig vom Zustand des Objektes
  - z.B. pushStack(...), popStack(...)
- Modale Klassen
  - Reihenfolge und Zustand

Checkliste für den Klassentest

- Wird jede Methode / Operation einmal ausgeführt?
- Werden alle Nachrichtenparameter und exportierten Attribute berücksichtigt mittels Äquivalenzklassen und Grenzwertanalyse?
- Wird jede geworfene Ausnahme erzeugt und jede abgefangene behandelt?
- Wird jeder Zustand erreicht? Wird jeder Zustandsübergang durchgeführt?
- Wird jede Operation in jeder äquivalenten Zustandsklasse ausgeführt (korrekt bearbeitet wo zulässig, zurückgewiesen wo unzulässig)?
- Werden adäquate Stress- und Lasttests für jede Klasse durchgeführt?

nach Binder, R.V.: Testing Object-Oriented Software: a Survey. Journal of Software Testing, Verification and Reliability, John Wiley & Sons, 1996, Vol. 6, No. 3/4