Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

Bachelor of Science (BSc) in Informatik

Modul Advanced Software Engineering 2 (ASE2)

LE 08 – Software Testing

4 Statischer Test

Institut für Angewandte Informationstechnologie (InIT)
Walter Eich (eicw) / Matthias Bachmann (bacn)
https://www.zhaw.ch/de/engineering/institute-zentren/init/

Agenda

4 Statischer Test

4.1.- 4.6 Strukturierte Gruppenprüfungen (Reviews)

4.7 Statische Analyse (Exkurs)

4.8 Wrap-up

Lernziele nach Syllabus ISTQB CTFL

3.1 Grundlagen des statischen Tests

FL-3.1.1 (K1) Arten von Softwarearbeitsergebnissen erkennen können, die durch die verschiedenen statischen Testverfahren geprüft werden können

FL-3.1.2 (K2) Beispiele nennen können, um den Wert des statischen Tests zu beschreiben

FL-3.1.3 (K2) Den Unterschied zwischen statischen und dynamischen Verfahren unter Berücksichtigung der Ziele, der zu identifizierenden Fehlerzustände und der Rolle dieser Verfahren innerhalb des Softwarelebenszyklus erklären können

3.2 Reviewprozess

FL-3.2.1 (K2) Die Aktivitäten des Reviewprozesses für Arbeitsergebnisse zusammenfassen können

FL-3.2.2 (K1) Die unterschiedlichen Rollen und Verantwortlichkeiten in einem formalen Review erkennen können

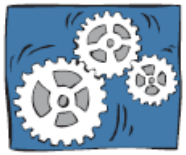
FL-3.2.3 (K2) Die Unterschiede zwischen den unterschiedlichen Reviewarten erklären können: informelles Review, Walkthrough, technisches Review und Inspektion

FL-3.2.4 (K3) Ein Reviewverfahren auf ein Arbeitsergebnis anwenden können, um Fehlerzustände zu finden

FL-3.2.5 (K2) Die Faktoren erklären können, die zu einem erfolgreichen Review beitragen

Statischer Test

- Testen von Software-Entwicklungsartefakten, ohne diese auszuführen, z.B. durch Reviews oder statische Analyse
- **Reviews**
  - Manuelle Prüfungen durch eine oder mehrere Personen
  - Menschliche Analyse- und Denkfähigkeit wird genutzt, um komplexe Sachverhalte zu prüfen und zu bewerten
  - Kann bei allen Dokumenten durchgeführt werden, die während des Softwareentwicklungsprozesses erstellt oder verwendet werden (z. B. Vertrag, Anforderungsspezifikation, Designspezifikation, Quelltext, Testkonzepte, Testspezifikationen, Testfälle, Testskripte oder Anwenderhandbücher)
- **Statische (Code-)Analyse**
  - Automatisierte Prüfungen durch entsprechende Werkzeuge
  - Nur bei Dokumenten mit formaler Struktur (z.B. Programmtext, UML-Diagramme)



4.1 Was kann analysiert und geprüft werden?

- Analyse des Testobjekts (Dokumente, Code etc.) durch intensive Betrachtung
- **Ziel:** Ermittlung von Fehlerzuständen (Defekten) im Dokument bzw. Code
  - Verstösse gegen Spezifikationen oder einzuhaltende Standards, Fehler in Anforderungen, Fehler im Design, unzureichende Wartbarkeit und falsche Schnittstellenspezifikationen
  - Nachweis der Verletzung von Projektplänen
  - Ergebnisse der Untersuchungen werden darüber hinaus dazu benutzt, den Entwicklungsprozess zu optimieren
- **Grundidee:** Prävention!
  - Fehler(zustände) und Abweichungen sollen so früh wie möglich erkannt werden, noch bevor sie im weiteren Verlauf der Softwareentwicklung zum Tragen kommen und aufwändige Nach- oder Verbesserungen nach sich ziehen.

4.2 Vorgehen beim statischen Test (1/2)

- Bei den nicht-werkzeuggestützten Verfahren wird die menschliche Analyse- und Denkfähigkeit genutzt, um komplexe Sachverhalte zu prüfen und zu bewerten.
- Dies erfolgt durch intensives Lesen und Nachvollziehen der untersuchten Dokumente.
- Es gibt verschiedene Vorgehensweisen zur Überprüfung der Dokumente, die sich durch die Intensität und die benötigten Ressourcen (Personen und Zeit) und die verfolgten Ziele unterscheiden.
- Das bekannteste und wichtigste Verfahren ist das sogenannte «Review».
- Reviews können von informell bis hin zu sehr formal durchgeführt werden.

4.2 Vorgehen beim statischen Test (2/2)

- Welche Ausprägung des Prozesses zur Durchführung eines Reviews genutzt werden soll bzw. kann, hängt von folgenden Faktoren ab:
  - Softwareentwicklungslebenszyklus-Modell – Welches Modell wird genutzt und welche «Stellen» und Arbeitsergebnisse im Modell eignen sich für die Durchführung von Reviews?
  - Reife des Entwicklungsprozesses – Wie hoch ist der Reifegrad des Entwicklungsprozesses und wie hoch ist damit die Qualität der Dokumente, die geprüft werden sollen?
  - Komplexität des zu überprüfenden Dokuments – Welche Arbeitsergebnisse weisen eine hohe Komplexität auf und sind einem – eher formalen – Review zu unterziehen?
  - Gesetzliche oder regulatorische Anforderungen und/oder die Notwendigkeit eines Prüfnachweises («Audit-Trail») – Welche Vorschriften sind einzuhalten und welche Massnahmen zur Qualitätssicherung – somit auch Reviews – sind nachzuweisen?

4.3 Der Reviewprozess

- Bei einem Review sind folgende Hauptaktivitäten durchzuführen:
  - Planung
  - Initiierung, Kick-Off
  - Individuelles Review
  - Diskussion der Befunde
  - Bericht und Fehlerbehebung

4.3.1 Planung

- Management entscheidet, welche Dokumente mit welcher Art von Review unterzogen werden sollen.
- Der zu veranschlagende Aufwand für die einzelnen Reviews ist bei der Projektplanung vorzusehen.
- Eingangs- und Austrittskriterien für die Durchführung des Reviews sind festzulegen.
- Manager wählt die fachlich geeigneten Personen aus und stellt ein Review Team zusammen.
- Er vergewissert sich in Kooperation mit dem Autor des Prüfobjektes, dass dieses einen «review-fähigen» Status hat, d.h., die Arbeiten am Dokument einen gewissen Abschluss gefunden und eine ausreichende Vollständigkeit erreicht haben.

4.3 Initiierung (Kick-off) (1/2)

- Versorgung der am Review beteiligten Personen mit allen benötigten Informationen.
- Überprüfung, ob Eingangsbedingungen für die Durchführung des Reviews erfüllt sind.
- Schriftliche Einladung oder sofortiges erstes Treffen des Reviewteams, um über Bedeutung, Sinn und Zweck der durchzuführenden Reviewsitzung zu informieren.
  - Sind die beteiligten Personen mit dem Umfeld des zu prüfenden Dokumentes wenig vertraut, kann auf dem Treffen neben der kurzen Vorstellung des Prüfdokuments auch seine Einbettung in das Anwendungsgebiet dargestellt werden.

4.3 Initiierung (Kick-off) (2/2)

- Neben dem Arbeitsergebnis, das einem Review unterzogen werden soll, müssen den beteiligten Personen weitere Unterlagen zur Verfügung stehen:
  - Dokumente, die herangezogen werden müssen, um zu entscheiden, ob eine Abweichung, ein Fehler oder eine korrekte Beschreibung des Sachverhalts vorliegt, also die Dokumente (z.B. Use Cases, Designdokumente, Richtlinien oder Standards), gegen die geprüft wird.
  - Diese Dokumente werden auch als Basisdokumente («Baseline») bezeichnet.
  - Darüber hinaus sind Prüfkriterien (z.B. in Form von Checklisten) sehr sinnvoll, die ein strukturiertes Vorgehen unterstützen.
  - Sind Vorlagen zur Protokollierung der Befunde einzusetzen, dann sind diese zu verteilen.

4.3 Individuelles Reviews (Individuelle Vorbereitung)

- Die beteiligten Personen des Reviewteams haben sich individuell auf die Sitzung vorzubereiten.
- Reviewer oder Gutachter setzen sich intensiv mit dem zu prüfenden Dokument auseinander und verwenden dabei die zur Verfügung gestellten Dokumente als Prüfgrundlage.
  - Meist ist es ratsam, eine Beschränkung auf bestimmte Aspekte vorzunehmen, gegeben durch die gezielte Auswahl der verwendeten Checklisten oder durch weitere Dokumente.
- Erkannte potenzielle Fehler(zustände), Empfehlungen, Fragen oder Kommentare werden notiert.

4.3 Diskussion der Befunde (Reviewsitzung)

- Nach der individuellen Vorbereitung werden die Ergebnisse zusammengetragen und diskutiert.
- Dies kann bei einer Reviewsitzung erfolgen oder auch auf andere Weise, z.B. in einem firmeninternen Onlineforum.
- Die von den einzelnen Reviewern identifizierten potenziellen Abweichungen und Fehler(zustände) werden besprochen und näher analysiert.
- In der Befundanalyse wird nun jedes Merkmal bewertet und das Ergebnis dokumentiert.
- Vom Reviewteam ist eine Empfehlung über die Annahme des Arbeitsergebnisses abzugeben:
  - *akzeptieren* (ohne Änderungen) bzw. akzeptieren (mit geringfügigen Änderungen),
  - *Überarbeitung erforderlich*, da umfangreiche Änderungen notwendig,
  - *nicht akzeptieren*.

4.3 Diskussion der Befunde (Reviewsitzung) (Exkurs)

- Findet zur Diskussion eine Reviewsitzung statt, können folgende Empfehlungen gegeben werden, die sich in der Praxis bewährt haben:
  - Die Reviewsitzung wird auf zwei Stunden beschränkt.
  - Der Moderator hat das Recht, eine Sitzung abzusagen oder abzubrechen, wenn einer oder mehrere Reviewer nicht erschienen oder ungenügend vorbereitet sind.
  - Das Resultat (also das zu prüfende Dokument, das Arbeitsergebnis) und nicht der Autor steht zur Diskussion.
  - Der Moderator darf nicht gleichzeitig als Reviewer fungieren.
  - Allgemeine Stilfragen (ausserhalb der Prüfkriterien) dürfen nicht diskutiert werden.
  - Die Entwicklung und Diskussion von Lösungen ist nicht Aufgabe des Reviewteams (Ausnahmen s.u.) und soll unterbleiben.
  - Jeder Reviewer muss Gelegenheit haben, seine Befunde angemessen präsentieren zu können.
  - Der Konsens der Reviewer zu einem Befund ist anzustreben und zu protokollieren.
  - Befunde sind nicht in Form von Korrekturanweisungen an den Autor zu formulieren.
  - Die einzelnen Befunde sind zu gewichten (z.B. als kritischer Fehler, Hauptfehler, Nebenfehler und gut).

4.3 Fehlerbehebung und Bericht

- Die abschliessende Aktivität im Reviewprozess ist die Erstellung von Berichten und die Behebung der aufgedeckten Fehler(zustände) bzw. Unstimmigkeiten.
- Oft beinhaltet das Protokoll der Reviewsitzung alle erforderlichen Informationen, sodass keine zusätzlichen Fehlerberichte angefertigt werden müssen, die jeden Fehler(zustand) einzeln dokumentieren.
- Im Regelfall wird der Autor die Fehler(zustände) in seinem Arbeitsergebnis auf Grundlage der Reviewergebnisse beseitigen und eine Überarbeitung vornehmen.
- Die Reviewergebnisse können in Abhängigkeit von der Reviewart und dem Grad der Formalität stark variieren.

4.3.2 Unterschiedliche Vorgehensweisen beim Individuellen Review

- Die grundlegende Vorgehensweise beim Reviewprozess sieht eine individuelle Vorbereitung – ein individuelles Review – vor.
- Für diese Vorbereitung gibt es eine Reihe von Verfahren, die genutzt werden können, um Fehler(zustände) aufzudecken.
- Nachfolgend sind Beispiele für verschiedene individuelle Vorgehensweisen aufgeführt.
  - Keine Vorgaben: Ad-hoc-Vorgehen
  - Strukturiertes Lesen: Vorgehen basiert auf Checklisten
  - Strukturierte Richtlinie: Vorgehen unter Nutzung von Szenarien und «Probeläufen» so genannten «Dry Runs» (z.B. Use Cases, User Stories)
  - Prüfung aus einer Perspektive: Rollenbasiertes und perspektivisches Vorgehen (z.B. aus Sicht Tester)

4.3.2 Beispiel Checklisten

- Checkliste für ein Review eines Anforderungsdokuments:
  - «Ist die Struktur des Anforderungsdokuments standardisiert?
  - Wird erläutert, wie das Anforderungsdokument zu benutzen ist? (einleitendes Kapitel)
  - Wurde eine Zusammenfassung des Projekts im Anforderungsdokument hinterlegt?
  - Ist ein Glossar zur Unterstützung des einheitlichen Verständnisses von Fachbegriffen verfügbar?
  - …»

4.3.3 Rollen und Verantwortlichkeiten im Reviewprozess (1/4)

- **Management**
  - Entscheidet über Durchführung von Reviews und deren Art.
  - Wählt die Prüfobjekte und die zu verwendenden Dokumente aus.
  - Verantwortlich für Planung und stellt die notwendigen Ressourcen (Personen, Budget, Zeit) zur Verfügung und das Review Team zusammen.
  - Überwachung der Kosteneffizienz und das Treffen von steuernden Entscheidungen im Fall von unzureichenden Ergebnissen des Reviewprozesses.
- **Reviewleiter**
  - Reviewleiter trägt die Gesamtverantwortung für das Review, d.h., für Planung, Vorbereitung, Durchführung und Überarbeitung sowie Nachbereitung.
  - Entscheidet, welche Personen am Review beteiligt sind, und organisiert Zeitpunkt und Räumlichkeiten, wenn eine Sitzung stattfindet.

4.3.3 Rollen und Verantwortlichkeiten im Reviewprozess (2/4)

- **Reviewmoderator (Moderator)**
  - Leitet die Sitzung und ist die entscheidende Person, was den Erfolg des Reviews betrifft.
  - Muss ein sehr guter Sitzungsleiter sein.
  - Muss bei gegensätzlichen Standpunkten vermitteln können und auch auf «Untertöne» achten.
  - Darf nicht voreingenommen sein und keine eigene Meinung zum Prüfobjekt äussern.
- **Autor**
  - Ersteller des Dokuments bzw. des Arbeitsergebnisses, das einem Review unterzogen wird.
  - Bei mehreren Erstellern sollte ein Hauptverantwortlicher benannt werden, der die Rolle des Autors übernimmt.
  - Wichtig ist, dass der Autor die geäusserte Kritik nicht als Kritik an seiner Person auffasst, sondern dass es ausschließlich um die Verbesserung der Qualität des Prüfobjektes geht.

4.3.3 Rollen und Verantwortlichkeiten im Reviewprozess (3/4)

- **Reviewer (Gutachter, Inspektor)**
  - Mehrere (maximal fünf) Fachexperten, die nach der entsprechenden Vorbereitung an der Sitzung teilnehmen.
  - Haben darauf zu achten, dass als gut befundene Teile der Dokumente auch so benannt werden.
  - Kennzeichnen mangelhafte Teile des Prüfobjektes und dokumentieren Mängel nachvollziehbar, damit deren Beseitigung möglich ist.
  - Reviewer sollten so gewählt werden, dass verschiedene Sichten und Rollen im Reviewprozess vertreten.

4.3.3 Rollen und Verantwortlichkeiten im Reviewprozess (4/4)

- **Protokollant**
  - Dokumentiert alle Ergebnisse, Probleme und offenen Punkte, die im Verlauf der Sitzung identifiziert werden.
  - Muss kurz und präzise formulieren können.
  - Schreibt die während der Sitzung meist nicht ausreichend formulierten Diskussionsbeträge verfälschungsfrei auf.
  - Die Rollen Protokollant und Reviewmoderator sollten nicht von einer Person wahrgenommen werden.
  - Aus pragmatischen Gründen kann es manchmal sogar sinnvoll sein, den Autor selbst das Protokoll führen zu lassen, denn er weiss genau, was im Einzelnen und wie präzise und ausführlich die Anmerkungen der Gutachter zu protokollieren sind, um ausreichend Information für die später von ihm durchzuführenden Änderungen zu haben.

4.4 Reviewarten

- **Informelles Review**
  - Abgeschwächte Form des Reviews
  - Autor-Leser-Feedbackzyklus
- **Walkthrough**
  - Im Mittelpunkt steht eine Sitzung
  - Kein formaler Ablauf
- **Technische Review (auch fachliches Review)**
  - Übereinstimmung des Prüflings mit einer Spezifikation, Eignung für den Einsatz, Einhaltung von Standards («gemeinsame Sicht auf die Dinge»)
  - Hinzuziehen von Fachexperten
- **Inspektion (auch Design- oder Code-/Softwareinspektion)**
  - Formalste Reviewart
  - Folgt einem definierten Ablauf (mit genauen Prüfkriterien)
  - Aufdeckung von Unklarheiten, mögliche Fehler(zustände), Bestimmung der Qualität, Vertrauen in das Arbeitsergebnis schaffen

4.5 Erfolgsfaktoren, Vorteile und Grenzen (1/2)

- Reviews sind ein effizientes Mittel zur Sicherung der Qualität der untersuchten Arbeitsergebnisse.
- Idealerweise sind sie direkt nach der Fertigstellung der Arbeitsergebnisse durchzuführen, um Fehler(zustände) und Unstimmigkeiten kurzfristig festzustellen und frühes Feedback zu liefern.
- Das frühe Finden von Fehler(zustände)n durch Reviews und deren unmittelbar anschliessendes Beseitigen ist in aller Regel kostengünstiger als das spätere Erkennen von Fehler(zustände)n in ablauffähigen Programmen durch dynamisches Testen.
- Da die Überprüfungen im Team durchgeführt werden, ergibt sich daraus ein Wissensaustausch unter den beteiligten Personen.

4.5 Erfolgsfaktoren, Vorteile und Grenzen (2/2)

- **Organisatorische Erfolgsfaktoren**
  - Management bzw. Projektleitung unterstützt Reviews
  - Formelle Reviews werden mit angemessener Frist geplant
  - Teilnehmern an einem Review steht genügend Zeit zur Vorbereitung zur Verfügung
  - Reviewart ist dem Kenntnisstand und der beteiligten Personen anzupassen
  - Verwendete Checklisten sind aktuell
  - Umfangreiche Dokumente werden nicht als Ganzes geprüft
- **Personenbezogene Erfolgsfaktoren**
  - Für Reviews sind die «passenden» Personen auszuwählen
  - Reviews dienen zur Qualitätssicherung der untersuchten Arbeitsergebnisse (Aufzeigen von Fehler(zuständen) ist gewünscht!)
  - Teilnehmer nehmen sich ausreichend Zeit und Aufmerksamkeit für Details.
  - Mit Reviews entsteht eine Kultur des «ständigen Lernens»

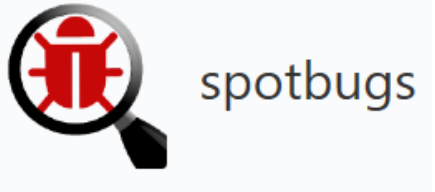
4.6 Unterschiede zwischen statischen und dynamischen Tests

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

- Statischer und dynamischer Test können die gleichen Ziele verfolgen, ergänzen sich aber gegenseitig, indem sie verschiedene Arten von Fehlern finden.
- Statische Tests entdecken Fehler(zustände) in den Arbeitsergebnissen bzw. Dokumenten direkt.
- Dynamische Tests weisen Fehlerwirkungen nach – in aller Regel im Programmcode und nicht in anderen Dokumenten (von ausführbaren Modellen abgesehen).
- Typische Fehler, die mittels statischen Tests erkannt werden:
  - Anforderungsfehler, Entwurfsfehler, Programmierfehler, Abweichungen von Standards, fehlerhafte Schnittstellen-spezifikationen, Schwachstellen in der Zugriffsicherheit, Lücken oder Ungenauigkeit bei der Rückverfolgbarkeit oder Überdeckungsgrad

4.7 Statische (werkzeugbasierte) Analyse (Exkurs)

- Statische Analyse wird oft als Obergriff für alle Prüfverfahren verwendet, bei denen keine Ausführung des Testobjekts erfolgt.
  - Achtung: Viele Defekte werden erst bei der Ausführung erkennbar (dynamischer) Test.
- Ziel ist die Aufdeckung vorhandener Fehler oder fehlerträchtiger Stellen in einem Dokument durch Werkzeuge (Analogie Rechtschreibprüfprogramm).



Statische (werkzeugbasierte) Analyse

- Überprüfung, Vermessung und Darstellung eines Dokuments bzw. eines Programms oder einer Komponente
- Dokumente müssen eine formale Struktur haben
- Typischerweise wird Quelltext analysiert.
  - z.B. Prüfung auf «toten» Code oder Klone
- Prüfung auf Einhaltung von Konventionen und Standards
  - Schachtelungstiefe, Kommentierungsgrad, ...
- Datenflussanalyse
  - Anomalien z.B. ur-, du-, dd- (u: undefiniert, d: definiert, r: referenziert)
- Kontrollflussanalyse
- Erhebung von Metriken

Compiler als statisches Analysewerkzeug

- Alle Compiler führen eine statische Analyse des Quelltextes durch.
  - Prüfung der Syntax
  - Cross Reference List
  - Typgerechte Verwendung
  - Ermittlung von nicht deklarierten Variablen
  - Nicht erreichbarer Code (dead code)
  - Über- / Unterschreitung von Feldgrenzen
  - Prüfung der Konsistenz der Schnittstellen

Prüfung der Einhaltung von Konventionen und Standards

- Einhaltung der Programmierkonventionen durch Analysatoren
- Möglichst nur Richtlinien verwenden die sich automatisch überprüfen lassen
- Werkzeuge: Checkstyle, FxCop

Datenflussanalyse

- Die Verwendung von Daten auf Pfaden wird untersucht
  - **ur-Anomalie:** Ein undefinierter Wert (u) einer Variablen wird auf einem Programmpfad gelesen (r).
  - **du-Anomalie:** Die Variable erhält einen Wert (d), der allerdings ungültig (u) wird, ohne dass er zwischenzeitlich verwendet wurde.
  - **dd-Anomalie:** Die Variable erhält auf einem Programmpfad ein zweites Mal einen Wert (d), ohne dass der erste Wert (d) verwendet wurde.
- Werkzeuge: SonarLint, SpotBugs

Datenflussanalyse

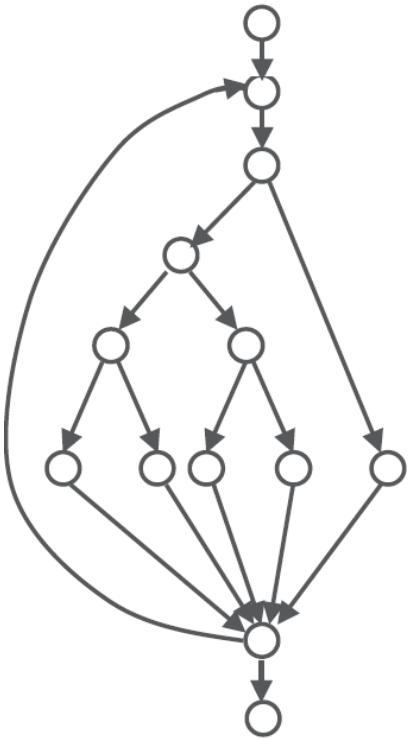
```
void tausch (int& Min, int& Max) {
    int Hilf;
    if (Min > Max) {
        Max = Hilf;
        Max = Min;
        Hilf = Min;
    }
}
```

C++

- **ur-Anomalie** der Variablen Hilf
  - Zuweisung ohne gültigen Wert
- **dd-Anomalie** der Variablen Max
  - 2 Zuweisungen nacheinander ohne dass die erste Zuweisung gebraucht wird
- **du-Anomalie** der Variablen Hilf
  - Letzte Zuweisung wird nirgendwo gebraucht

Kontrollflussanalyse

- Ein Programmstück kann als Kontrollflussgraph dargestellt werden.
- Anweisung + Sequenz von Anweisung ist ein Knoten.
- Änderungen werden durch Verzweigungen und Schleifen erreicht.
- Durch die Visualisierung der Kontrollflussgraphen lassen sich Anomalien leicht erfassen.
  - Sprünge aus Schleifen
  - Mehrere Ausgänge



Ermittlung von Metriken

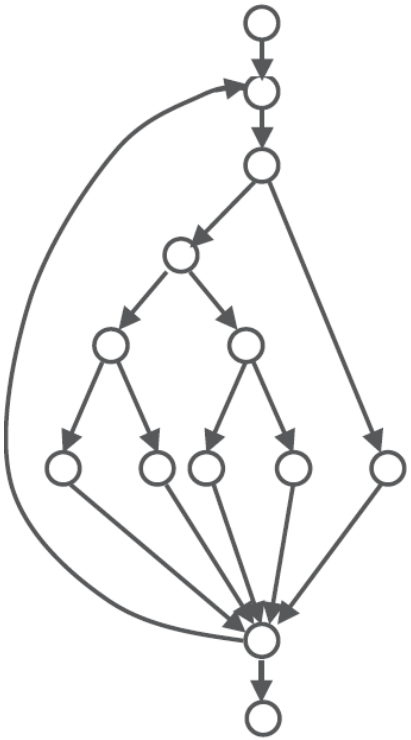
- Werkzeuge der statischen Analyse liefern auch Messwerte (Masszahlen oder Metriken)
  - Siehe dazu ISO DIN 25010
  - Eine Metrik liefert eine Aussage eines untersuchten Aspekts
- Werkzeuge: SonarQube, Eclipse Metrics

Einige Metriken für objektorientierte Programme

Kürzel	Bezeichnung	Erläuterung
NOV	Number of Variables	Anzahl der Instanzvariablen (member variables) einer Klasse
NOM	Number of Methods	Anzahl der Methoden (Operationen) einer Klasse
WMC	Weighted Method Complexity	Komplexität der Methoden einer Klasse WMC = ΣC(i) mit C(i) = Komplexität von Methode i
LCOM	Lack of Cohesion of Methods	Anzahl der durch die Methoden einer Klasse gemeinsam benutzten Instanzvariablen
NORM	Number of Redefined Methods	Anzahl der in einer Klasse redefinierten Methoden
NOC	Number of Children	Anzahl der direkten Unterklassen
DIT	Depth of Inheritance Tree	Maximale Tiefe der Generalisierungshierarchie

Zyklomatische Zahl

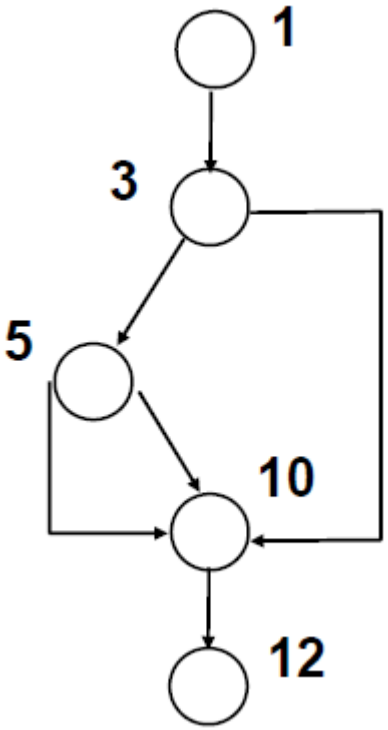
- Die zyklomatische Zahl oder Komplexität (nach McCabe) misst die strukturelle Komplexität des Quellcodes.
- Grundlage für die Berechnung ist der Kontrollflussgraph.
- Die zyklomatische Zahl kann genutzt werden, um Abschätzungen in Bezug auf die Testbarkeit und die Wartbarkeit des Programmteils vorzunehmen!



Zyklomatische Zahl – Ein Beispiel

Gegeben:

```
int example(int i, int j){
   int x = 0;
   if (i<j){
     int k = i+2;
     if (i < 10 || j != k){
        x=x-j;
     }else{
        x=x+j;
     }
   }
   return x;
}
```



Gesucht: v(G) = e - n + 2

v(G) = zyklomatische Zahl des Graphen G

n = Anzahl der Knoten

e = Anzahl der Kanten

Lösung:

n = 5, e = 6

v(G) = e - n + 2
= 6 - 5 + 2 = **3**

Zyklomatische Zahl

- Zyklomatische Zahl höher als 10 ist nach McCabe nicht tolerabel.
  - Überarbeitung des Programmteils bzw. der Methode in einem objektorientierten Programm!
  - Messwerte 6 und 3 in den Beispielen auf Folie 35 und 36 liegen im Bereich, den McCabe für akzeptabel hält.

- Für die Wartbarkeit ist die Verständlichkeit eines Programmstücks von entscheidender Bedeutung.
  - Je höher die ermittelte zyklomatische Zahl, je schwieriger ist ein Nachvollziehen des Ablaufs des Programmstücks und je schlechter ist damit die Verständlichkeit.

Zyklomatische Zahl

- Zyklomatische Zahl gibt Auskunft über den Testaufwand
  - Zyklomatische Zahl = Anzahl der linear unabhängigen Pfad**e** im Programmstück
  - Oft wird die 100%-ige Ausführung aller Anweisungen und Verzweigungsmöglichkeiten eines Programms verlangt – dafür könnten z.B. alle unabhängigen Pfade durch den Kontrollflussgraphen einmal «durchlaufen» werden.
  - Die zyklomatische Zahl gibt somit eine obere Grenze für die Anzahl der benötigten Testfälle zur Erreichung dieses Kriteriums an.

Wrap-up

- Reviews sind statische Tests, bei denen das Arbeitsergebnis – im Gegensatz zum dynamischen Test – nicht auf einem Rechner zur Ausführung kommt.
- Mehrere Augenpaare sehen mehr als ein Augenpaar – auch in der Softwareentwicklung.
- Der Reviewprozess besteht aus den folgenden Aktivitäten: Planung, Initiierung, individuelles Review, Reviewsitzung, Fehlerbehebung und Bericht.
- Rollen beim Review und damit zu verteilende Aufgaben sind: Manager, Reviewleiter, Moderator, Autor, Reviewer (Gutachter) und Protokollant.
- Es gibt eine ganze Reihe von unterschiedlichen Ausprägungen oder Arten von Reviews: informelle Reviews, Walkthrough, technische Reviews und Inspektion.
- Mittels werkzeuggestützter, statischer Analyse können Fehler(zustände) vor dem dynamischen Testen eruiert werden (kostengünstiger).

Ausblick

- Das Thema der nächsten Vorlesung ist:
  - Kap. 5 Dynamischer Test