Foto: James Harrison auf Unsplash

Testen von KI-Systemen

Teil 2 – Testprozedur

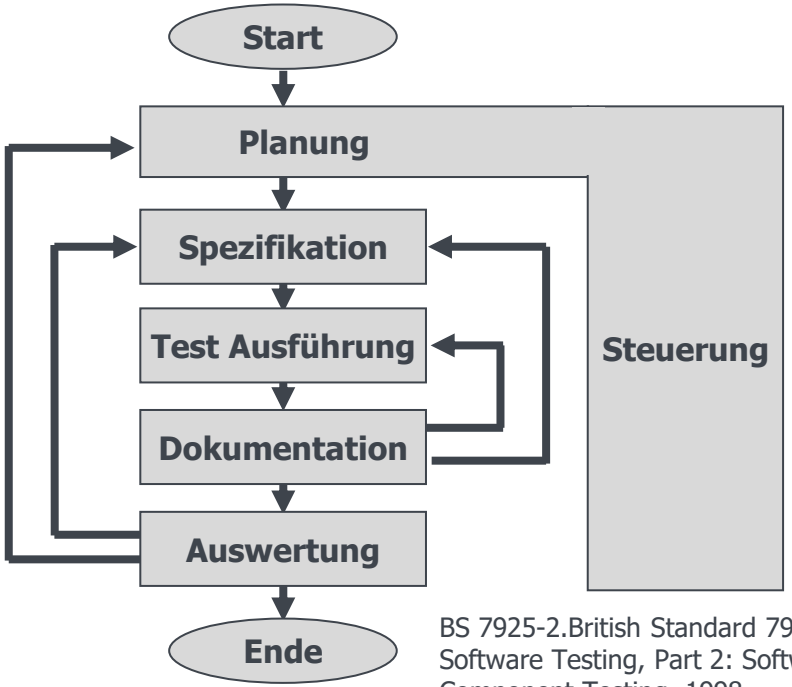
INNOVET KI B³
Künstliche Intelligenz in die berufliche Bildung bringen

INNOVET
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
bibb Bundesinstitut für Berufsbildung

Gefördert als InnoVET-Projekt aus Mitteln des Bundesministeriums für Bildung und Forschung.

Testprozedur – Überblick

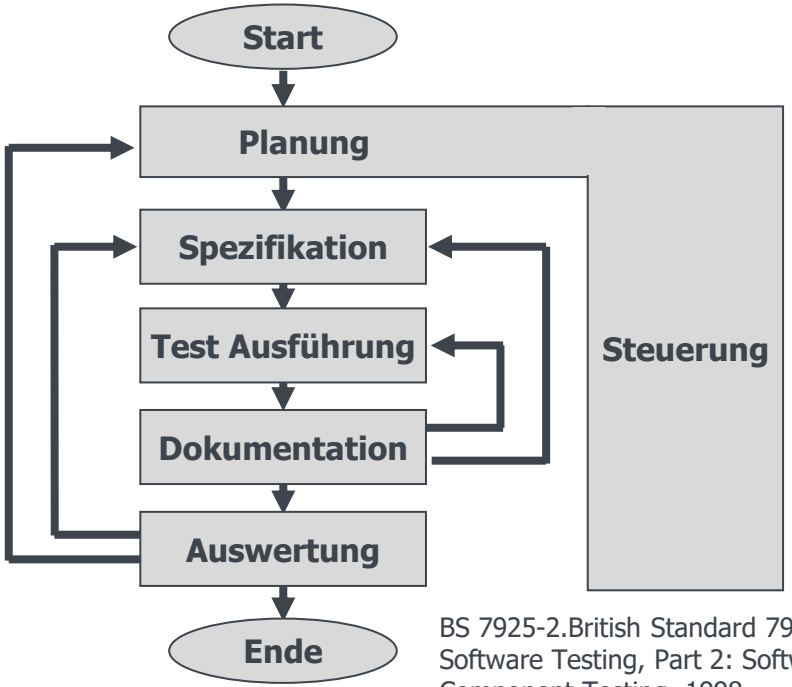
- Der Softwareprozess macht meist keine Vorgaben zum Testprozess.
  - Es ist ein Testverfahren erforderlich.
- Der Prozess sieht auf jeder Ebene anders aus.



BS 7925-2.British Standard 7925-2, Software Testing, Part 2: Software Component Testing, 1998

Testprozedur – Überblick

- Der Grundlegende Testprozess läuft fast immer so ab:
  - Planung (und Steuerung)
  - Spezifikation der Testfälle
  - Ausführen der Tests
  - Dokumentation der Testergebnisse
  - Auswertung des Tests und Entscheidung, ob die Tests fortgeführt werden sollen



BS 7925-2.British Standard 7925-2, Software Testing, Part 2: Software Component Testing, 1998

Testprozedur – Planung (und Steuerung)

In dieser Phase werden die Testziele, der Testumfang, die verfügbaren Ressourcen und der Zeitplan festgelegt.

Dazu gehört auch die Festlegung auf eine Teststrategie, und die Planung zum Management der folgenden Schritte.

Planung (und Steuerung)

- Wie soll das KI-System getestet werden? Wann gilt der Test als erfolgreich?
- Beispiel Recommender-System:

- **Ähnlichkeitsbasiert**: Das System empfiehlt Inhalte, die Inhalten ähneln, für die sich der Nutzer in der Vergangenheit interessiert hat, basierend auf Bildern, Beschreibungen oder Titel.
  - **Beispiel**: Wenn der Nutzer sich für Fußball interessiert, sollen Fußballvideos oder verwandte Inhalte vorgeschlagen werden.

Planung (und Steuerung)

- Wie soll das KI-System getestet werden? Wann gilt der Test als erfolgreich?
- Beispiel Recommender-System:

- **Heuristisch**: Das System empfiehlt Inhalte, für die sich andere Nutzer mit ähnlichen Vorlieben interessiert haben.
  - **Beispiel**: Wenn andere Nutzer mit ähnlichem Filmgeschmack auch Film X mochten, wird das System Film X vorschlagen.

Planung (und Steuerung)

- Wie soll das KI-System getestet werden? Wann gilt der Test als erfolgreich?
- Beispiel Recommender-System:

- **Kurzfristig / Sitzungsbasiert**: Das System empfiehlt Elemente, unter Berücksichtigung der Sitzung.
  - **Beispiel**: Auch wenn der Nutzer sich normalerweise eher für Horrorfilme interessiert, werden ihm direkt nach dem Konsum von „König der Löwen“ weitere Kinderfilme vorgeschlagen.

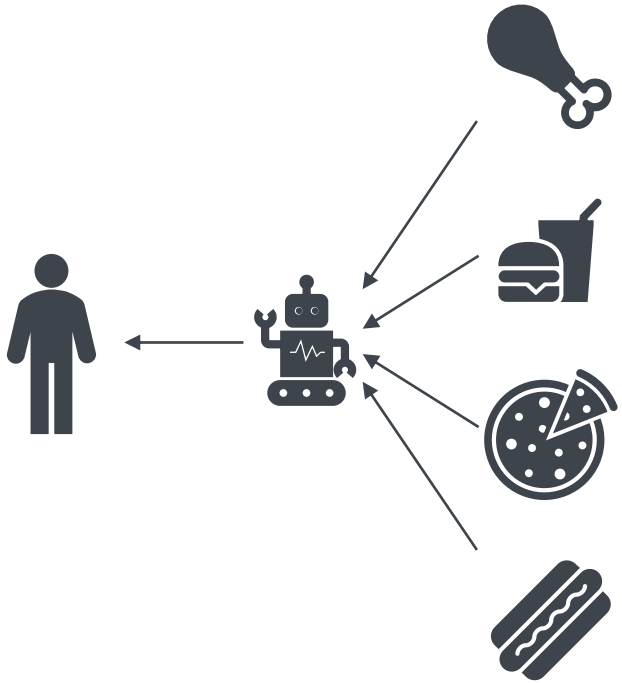
Testprozedur – Spezifikation

In dieser Phase werden die Testfälle und Szenarien festgelegt.

Dies geschieht anhand der Spezifikation der Software.

Spezifikation (Analyse und Entwurf)

- Hierbei muss entschieden werden, was getestet werden soll.
  - **Beispiel**: Es werden Testfälle erstellt, welche die Genauigkeit und Effektivität eines Empfehlungssystems von Fast-Food prüfen sollen.

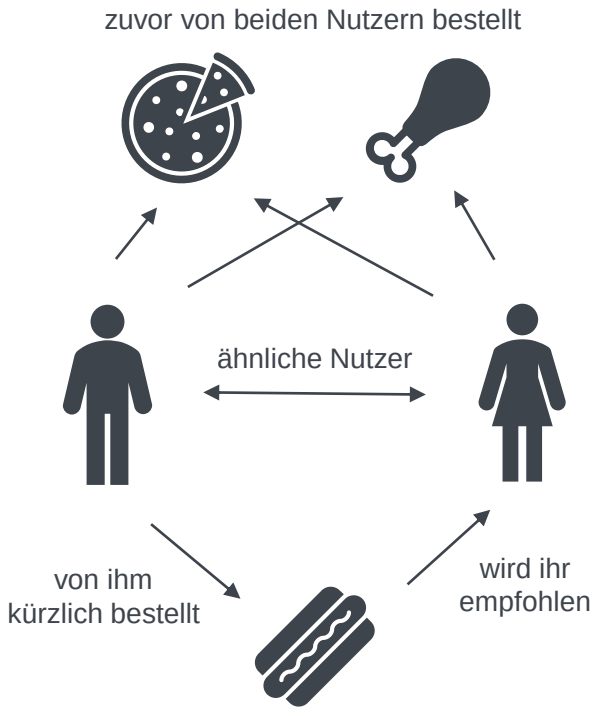


Testprozedur – Testausführung

In dieser Phase werden die Testfälle implementiert und mit festgelegten Testdaten ausgeführt.

Testausführung

- Das KI-System wird anhand der definierten Testfälle getestet.
  - **Beispiel**: Testfälle ausführen, die die Relevanz von Essensempfehlungen anhand von Heuristiken bewerten.



Testprozedur – Dokumentation

In dieser Phase werden die Ergebnisse aus der Testausführung dokumentiert. Dazu gehören aufgetretene Fehler und andere Unregelmäßigkeiten.

Dokumentation der Testergebnisse

- **Beispiel**: Dokumentation von Testergebnissen und Problemen während der Testausführung, nachdem ein KI-Empfehlungssystem eines Onlineshops getestet wurde.

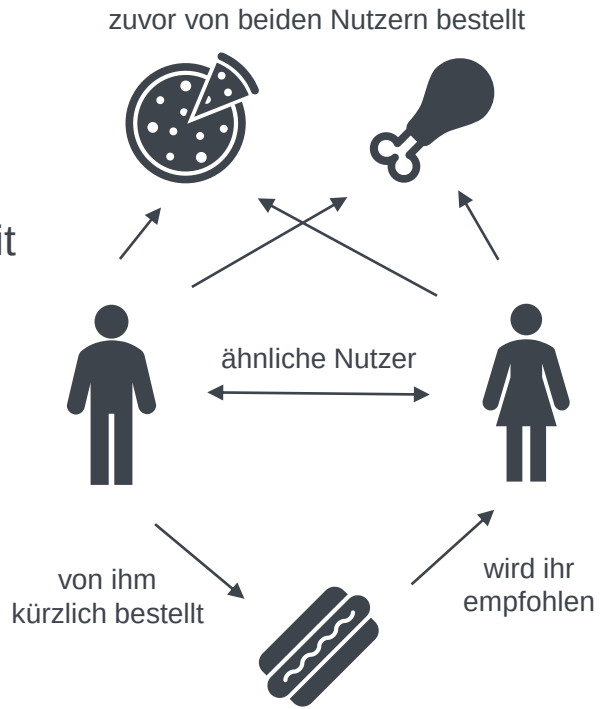
Testprozedur – Testauswertung

In dieser Phase wird die Dokumentation der Testläufe gegen definierte Kriterien geprüft.

Darauf aufbauend fällt die Entscheidung, ob weiter getestet werden soll, weitere Testfälle benötigt werden, oder die Tests abgeschlossen sind.

Testauswertung

- **Beispiel**: Ähnlichkeitsbasiertes Empfehlungssystem:
  - **Testziel**: Genauigkeit der Ähnlichkeitsmessung eines Essens-Empfehlungssystems sicherstellen.
  - **Bewertung**: Vergleich von empfohlenen Gerichten mit bereits bestellten Gerichten. Messung des Anteils an relevanten Empfehlungen.
  - **Testerfolg**: Das System gilt als erfolgreich getestet, wenn die Ähnlichkeit der empfohlenen Gerichte zu bestellten Gerichten immer mindestens 90% beträgt.



Terminologie

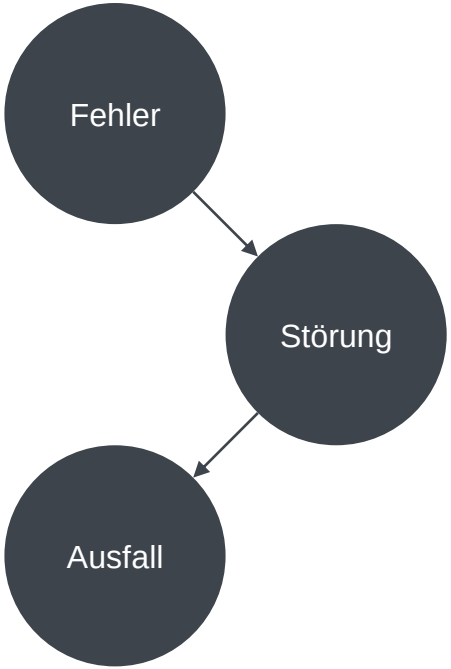
- Die Begriffe „Fehler“, „Bug“, „Störung“ und „Ausfall“ werden oft unpräzise verwendet.
- Im englischen wurden einige Begriffe standardisiert (IEEE 610.12):
  - „Failure“ (Ausfall),
  - „Fault“ / „Defect“ (Störung),
  - „Error“ (Fehler),
  - „Error propagation“ (Fehlerfolgen)
  - „Testing“, „Debugging“,
  - „Validation“, „Verification“.

Terminologie – Ausfall

Ein Ausfall ist eine Nicht-Erfüllung von Anforderungen, also eine Diskrepanz von tatsächlichem Verhalten (beim Testfall) und erwarteten Verhalten (nach Spezifikation)

Terminologie – Ausfall

- Ein Ausfall kann nur als solcher bezeichnet werden, wenn überhaupt ein Sollverhalten definiert wurde.
  - Ausfälle werden beim Testen gefunden.
  - KI-Beispiel: Wenn ein Nutzer trotz vorhandener Nutzerdaten irrelevante Empfehlungen bekommt, ist das ein Ausfall der Recommender-Systems.



Terminologie – Störung

- Eine Störung ist die Ursache für einen Ausfall.
  - Z.B. eine falsche Zeile Code in einer Software.
  - Eine Störung kann mehrere Ausfälle verursachen, aber auch keinen.
    - Wenn die Störung nie ausgeführt wird, verursacht sie keinen Ausfall.
    - Eine Störung im Empfehlungsalgorithmus könnte zur Empfehlung unpassender Produkte führen.
    - In der Praxis kann es schwer sein, die Störung zu finden, die einen Ausfall verursacht.

Terminologie – Störung

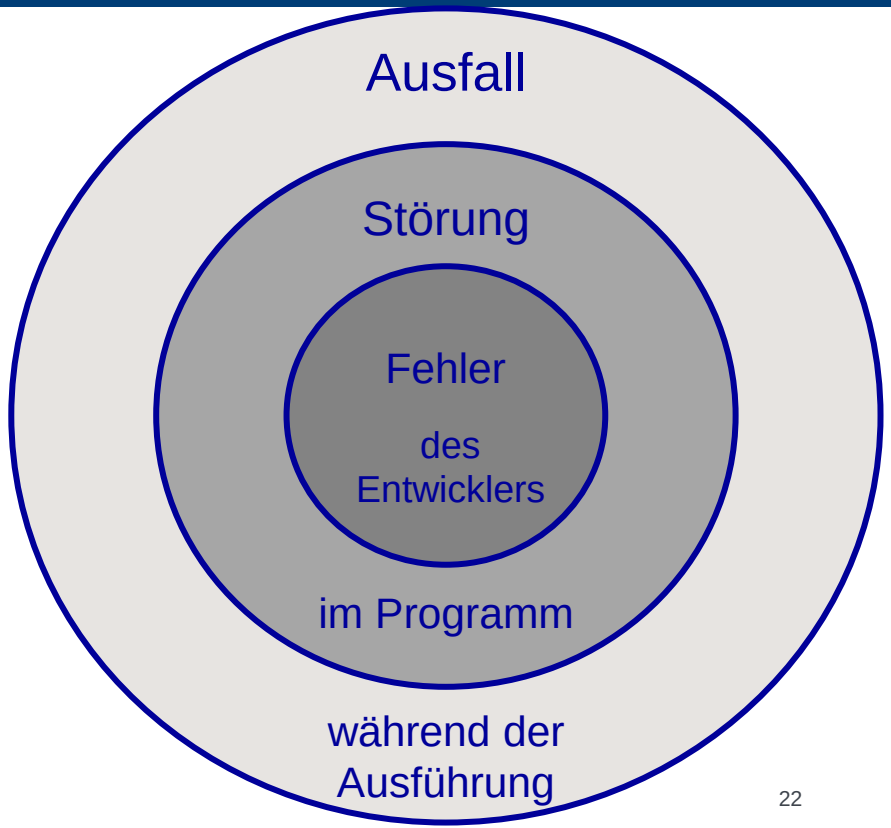
- Verdeckte Störung: Manchmal wird eine Störung von einer anderen Störung verdeckt.
  - Ein Ausfall, der von einer Störung A verursacht wird, führt dazu, dass der Code mit Störung B nie ausgeführt wird.
    - Sobald die Störung behoben ist, tritt ein weiterer Ausfall wegen Störung B auf.
  - In seltenen Fällen können zwei Störungen sich gegenseitig verdecken, sodass das Verhalten auf den ersten Blick korrekt aussieht.

Terminologie – Fehler

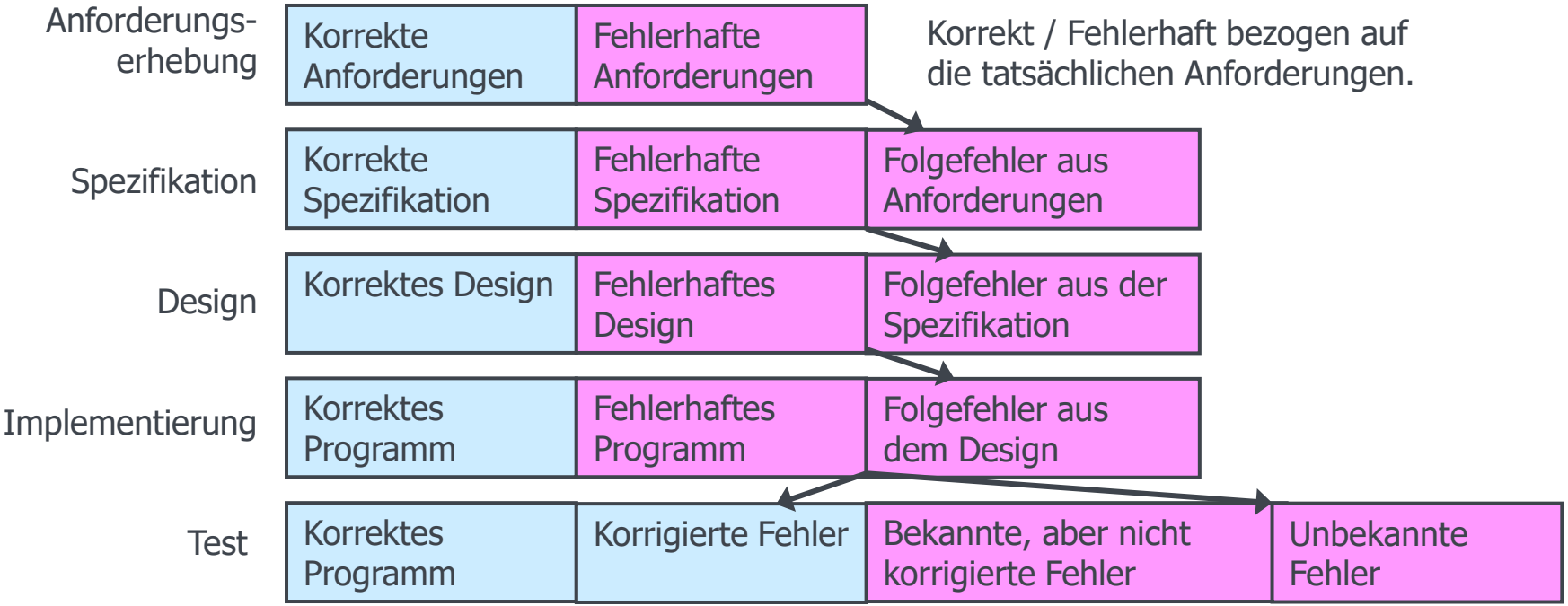
- Ein Fehler ist die Ursache einer Störung.
  - Der Fehler einer Person führt zu der Störung.
    - Ein Entwickler versteht die Spezifikation falsch und schreibt Code mit einem Fehler.
    - Der Datenanalyst trainiert das Empfehlungssystem versehentlich mit voreingenommenen Daten, was zu voreingenommenen Empfehlungen führt.

Terminologie – Ausfall, Störung, Fehler

- Zusammenfassung:
  - Fehler bei der Entwicklung führen zu Störungen der Software, die bei der Ausführung Ausfälle verursachen können.
- Hinweis:
  - In der Praxis werden die Begriff oft mit anderer Bedeutung verwendet.
    - Z.B. „Fehlercode“ und „Fehlerbehandlung“, statt „Störungscode“ und „Störungsbehandlung“



Terminologie – Propagation of defects/errors



Terminologie – Arten von Fehlern und Störungen

Je nach Problemquelle kann es sich um Fehler oder Störungen handeln!

- **Berechnungsfehler:** Eine Funktion berechnet ein falsches Ergebnis.
  - Z.B. wegen eines Rundungsfehlers.
- **Schnittstellenfehler:** Inkonsistenz zwischen aufrufendem und aufgerufenem Code.
  - Z.B. falsche Parameter oder falsche Reihenfolge der Parameter.
  - Z.B. Verletzung der Vorbedingungen für den Aufruf, bspw. Übergabe von `null` als Parameter.

Terminologie – Arten von Fehlern und Störungen

- **Kontrollflussfehler:** Ausführung des falschen Programmpfades.
  - Z.B. wegen vertauschter Statements.
  - Z.B. wegen einer fehlerhaften Bedingung, etwa < anstelle von <=.
- **Initialisierungsfehler:** Falsche oder fehlende Initialisierung.
  - Z.B. wird eine Variable nur in einem der möglichen Pfade initialisiert.

Terminologie – Kinds of defects/errors

- **Datenflussfehler**: Fehlerhafter Zugriff auf Daten und Datenstrukturen.
  - Z.B. falscher Zugriffsindex.
  - Z.B. Zuweisung an eine falsche Variable.
  - Z.B. Zugriff über eine ungültige Referenz.
  - Z.B. Zugriff auf bereits freigegebenen Speicher.

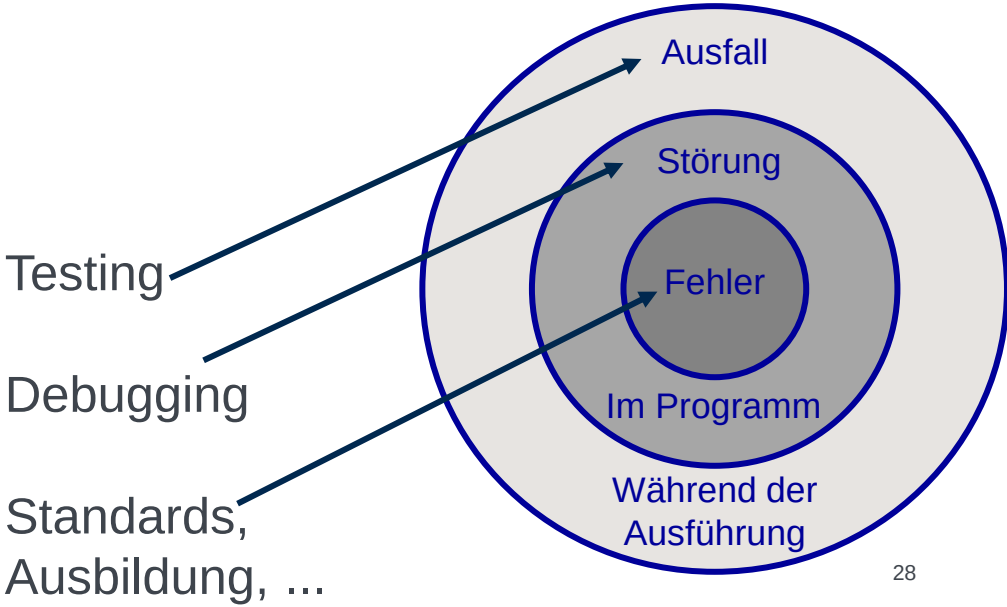
Terminologie – Arten von Fehlern und Störungen

- **Timing-Fehler**: Anforderungen bzgl. Timing nicht erfüllt.
  - Z.B. Implementierung nicht ausreichend effizient.
  - Z.B. wichtiger Interrupt zu lange deaktiviert.
- **Vererbungsfehler:** Vererbte Operation nicht passend implementiert.
  - Die Vor- und Nachbedingungen einer Elternklasse gelten nicht für eine Unterklasse.

Terminologie – Testing und Debugging

- **Testing ist kein Debugging**!
  - Testing entdeckt Ausfälle.
    - Beim Testing eines Empfehlungssystems wird geprüft, ob die Empfehlungen relevant sind.

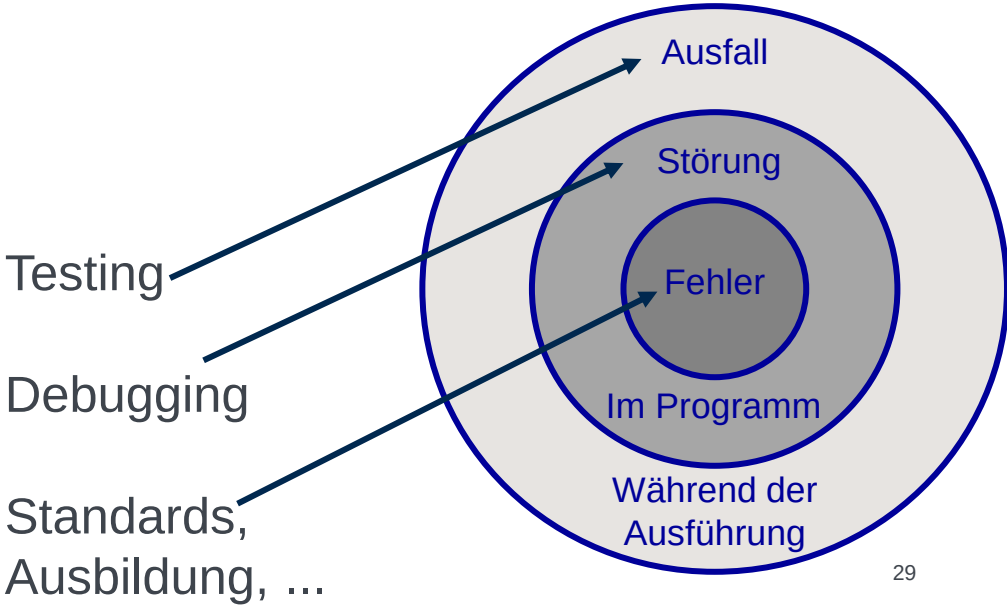
In seltenen Fällen kann der Debugger die Programmausführung beeinflussen und dabei Fehler erzeugen oder verdecken!



Terminologie – Testing und Debugging

- Debugging findet Störungen die zu Ausfällen führen.
  - Z.B. Störungen zu Ausfällen, die im Testing gefunden wurden.
  - Debugging enthält meist auch die Korrektur der Software, sodass die Störung nicht mehr auftritt.

In seltenen Fällen kann der Debugger die Programmausführung beeinflussen und dabei Fehler erzeugen oder verdecken!



Terminologie – Zweck von Testing

- Testing kann mögliche Ausfälle finden.
  - Testing kann die Anwesenheit von Fehlern zeigen, aber niemals die Abwesenheit, da vollumfängliches Testing in der Praxis unmöglich ist – selbst für kleinste Programme dauert das bereits Jahre.
- Testing kann Qualität messen.
  - Als Entscheidungsgrundlage – ist das Produkt bereit für ein Release?
- Testing kann Vertrauen schaffen.
  - Wenn nur wenige (oder keine) Ausfälle entdeckt werden, erhöht sich das Vertrauen in das Produkt.

Terminologie – Validierung und Verifikation

- Validierung
  - Prüft, ob das Endresultat der Entwicklung die Erwartungen des Nutzers erfüllt.
  - „Haben wir das richtige System gebaut?“
- Verifikation
  - Prüft, ob das Resultat der Entwicklung die Anforderungen erfüllt.
  - „Haben wir das System richtig gebaut?“

Terminologie – Validierung und Verifikation

- Ein verifiziertes System könnte sich für den Nutzer unerwartet verhalten, obwohl alle Anforderungen erfüllt sind!
  - Allerdings waren die Anforderungen eventuell ungeeignet

- In einem Empfehlungssystem bestätigt die Validierung die Relevanz der Vorschläge für den Nutzer, die Verifikation bestätigt dagegen, ob die Vorschläge den in den Anforderungen definierten Kriterien genügen.

INNOVET
KI B³
Künstliche Intelligenz in die berufliche Bildung bringen

Tobias Eisenreich

Universität Stuttgart
Institut für Software Engineering
Empirisches Software Engineering

Umm-e-Habiba

Universität Stuttgart
Institut für Software Engineering
Empirisches Software Engineering

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung
Institut für Software Engineering

IHK Industrie- und Handelskammer Reutlingen
Reutlingen | Tübingen | Zollernalb

IHK Region Stuttgart

IHK Industrie- und Handelskammer Karlsruhe

LMU Ludwig-Maximilians-Universität München

GEFÖRDERT VOM
INNOVET
Bundesministerium für Bildung und Forschung
bibb Bundesinstitut für Berufsbildung

Gefördert als InnoVET-Projekt aus Mitteln des Bundesministeriums für Bildung und Forschung.

Lizenzbestimmungen

„Testprozedur“ von Umm-e-Habiba und Tobias Eisenreich, KIB3 / Universität Stuttgart

Quellenverzeichnis

https://unsplash.com/de/fotos/vpOeXr5wmR4

BS 7925-2.British Standard 7925-2, Software Testing, Part 2: Software Component Testing, 1998

https://unsplash.com/photos/white-printer-paper-close-up-photography-0LaBRkmH4fM