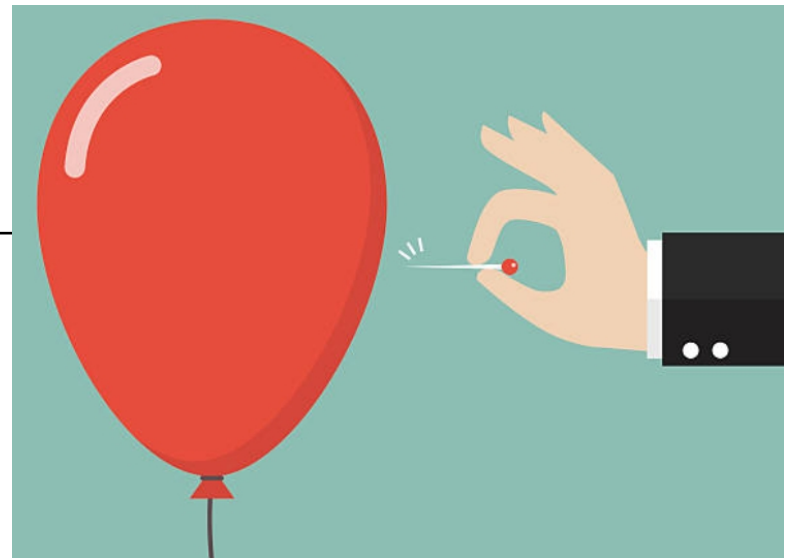


Der systematische Test

Ein systematischer Test ist ein Test, bei dem:

- die Randbedingungen definiert oder präzise erfasst sind
- die Eingaben systematisch ausgewählt wurden
- die Ergebnisse dokumentiert und nach Kriterien beurteilt werden

Testziel



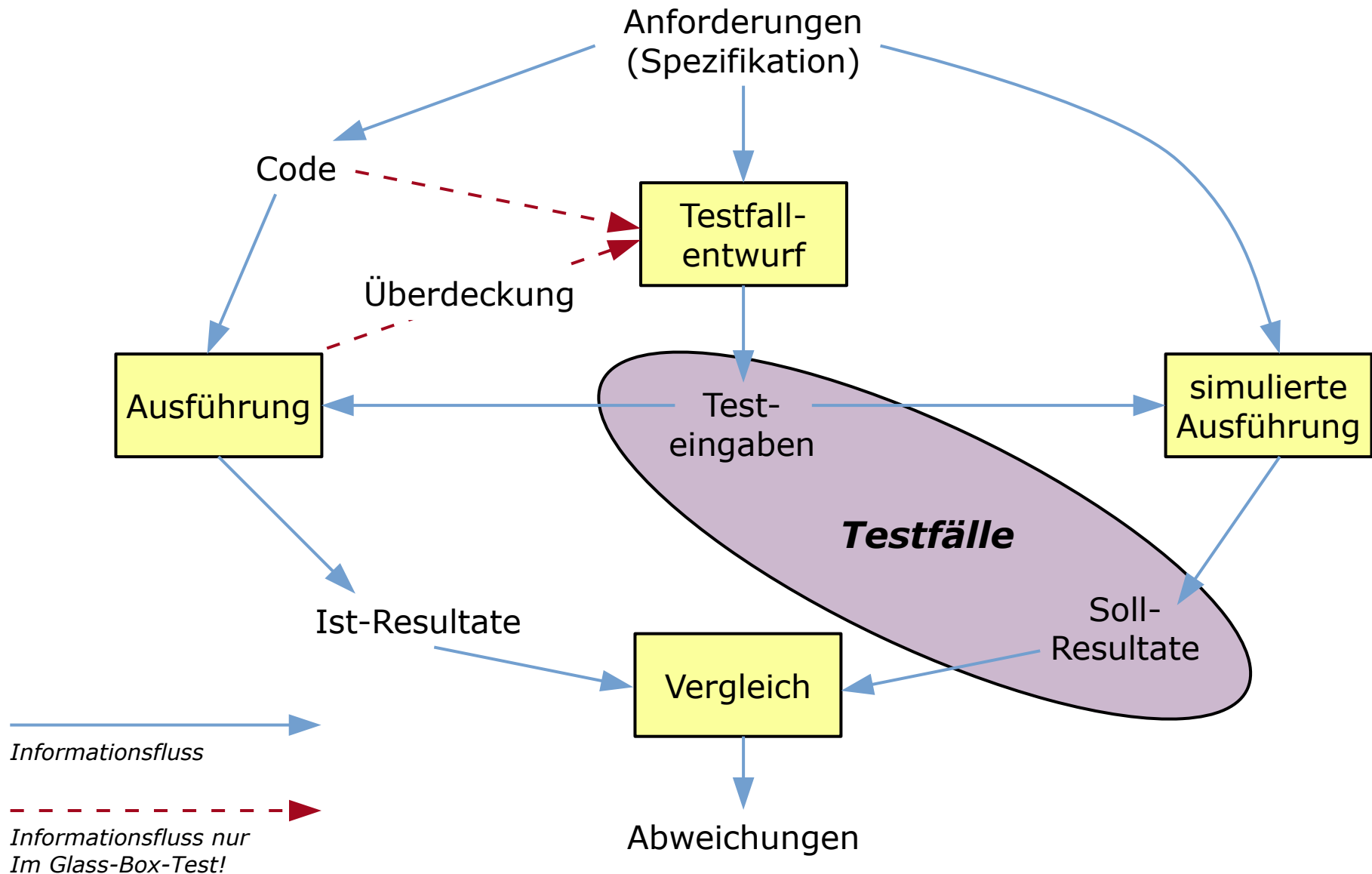
Der Test soll die Einsatzsituation so simulieren, dass aus einem **erfolglosen** („negativ“) **Test** auf einen **erfolgreichen Einsatz** geschlossen werden kann.

Der Zweck des Tests ist, **Fehler** zu **entdecken**.

Ein **Testfall** ist also **gut**, wenn er hohe Chancen hat, einen vorher noch nicht bekannten **Fehler anzuzeigen**. Tritt dieser Fall ein, so war der Test erfolgreich.

Ist kein Fehler entdeckt worden, dann war der Test erfolglos.

Informationsfluss beim Test



Soll-Resultate

Soll Resultate können oft nicht eindeutig angegeben werden, weil

- es mehrere oder viele richtige Ergebnisse gibt
- das Ergebnis nicht präzise spezifiziert ist, weil Anforderungen weich oder nur vage definiert sind
- es kein mathematisch exakt richtiges Ergebnis gibt, das auf dem Rechner darstellbar wäre
- das richtige Ergebnis einfach nicht bekannt ist

Stattdessen Prädikate, z. B. Junit

```
assertTrue (917.1 <= x & x <= 917.3);
```

Definition Regression Testing

Selective **re**testing of a system or component to verify that modifications have not caused unintended effects and that the system or component **still** complies with its specified requirements.

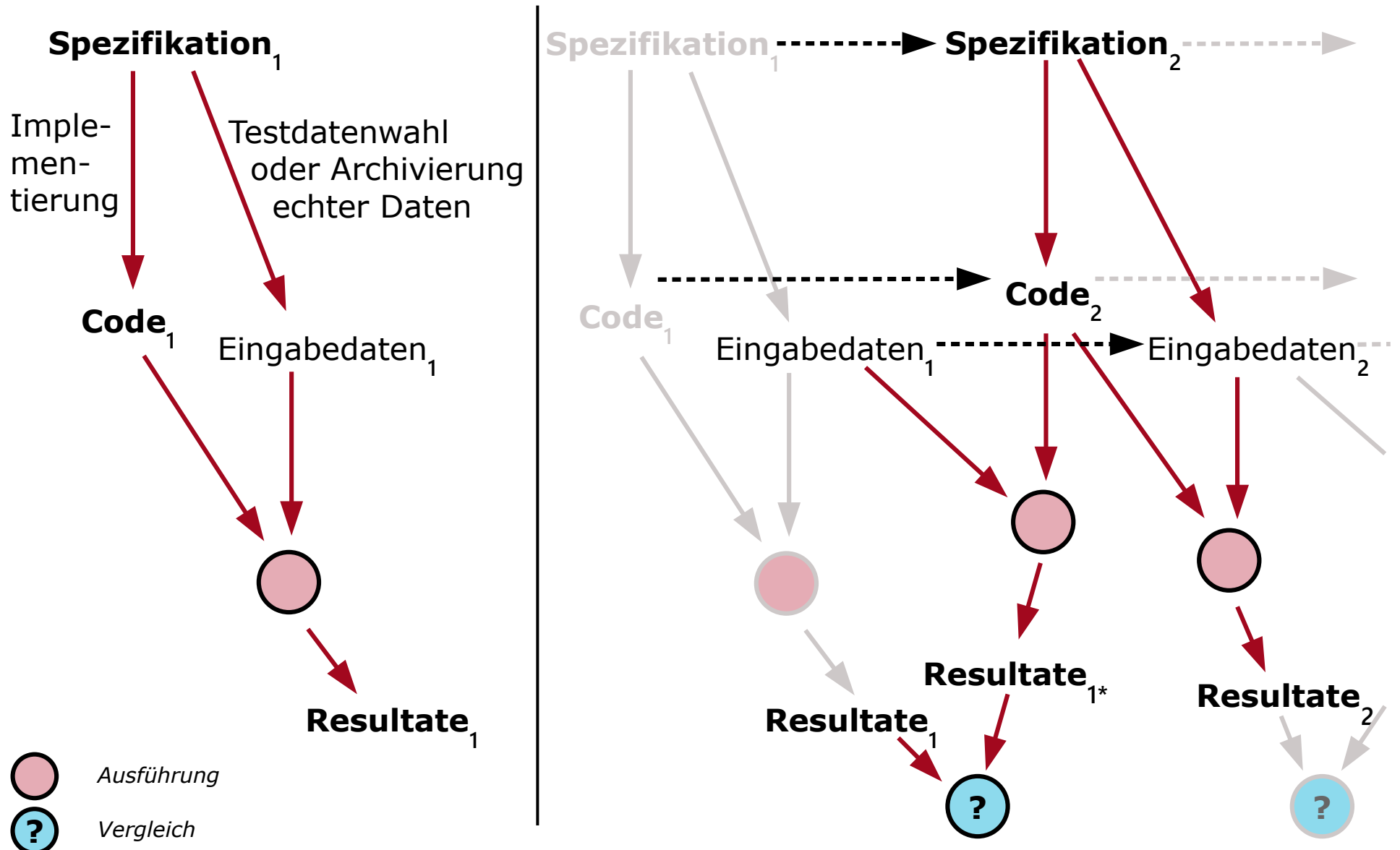
IEEE Std 610.12 (1990)

Ziel: Fehler entdecken, die bei einer Wartung entstanden sind

Annahmen beim Regressionstest

- Die Änderung des Programms ist geringfügig, das Verhalten des Programms ist nach der Änderung überwiegend wie vorher.
- Das Programm wurde vor der Änderung eingesetzt und hat dabei keine oder nur wenige falsche Resultate geliefert; es war also grundsätzlich in Ordnung, selbst wenn die Änderung eine Korrektur war.
- Nach einer Korrektur liefert das Programm unter gleichen Bedingungen gleiche Resultate (mit Ausnahme der bislang falschen Resultate).
- Nach einer Erweiterung liefert es gleiche Resultate, denn die neue Funktionalität kann bislang noch nicht verwendet worden sein. Die alte Funktionalität darf sich nicht geändert haben.
- Nach einer Modifikation dürfen die neuen Resultate dort und nur dort von den alten abweichen, wo dies gewünscht war.

Vorgehensweise beim Regressionstest



Wer einen Löwen jagt, muss wissen, wie ein Löwe aussieht.

Leider wissen wir aber über die Fehler fast nichts, nur dies:
Es gibt mindestens einen Testfall, der den Fehler anzeigt,
bei dem also Soll und Ist-Resultate nicht übereinstimmen.

Punktfehler

Bereichsfehler

Program testing can be used to show the presence of bugs, but never to show their absence!

E.W. Dijkstra (1970)

Vorteile des Testens

- Testen ist ein „natürliches“ Prüfverfahren. Denn wir probieren alles aus, um zu sehen, ob es funktioniert, oft auch, um es (im ursprünglichen Sinne) zu begreifen.
- Der (systematische) Test ist reproduzierbar und damit objektiv, wenn die Anfangssituation reproduzierbar ist und die Testumgebung deterministisch arbeitet. Sein Erfolg hängt also nicht von der Tagesform des Testers ab.
- Den investierten Aufwand kann man mehrfach nutzen. Denn ein Test lässt sich, einmal vorbereitet und sauber dokumentiert, mit geringem Aufwand wiederholen. Das ist vor allem in der Software-Wartung ein entscheidendes Argument.
- Die Testumgebung wird mitgeprüft. Fehler in den beteiligten Komponenten (Übersetzer, Bibliotheken, Betriebssystem usw.) fallen möglicherweise im Test auf.
- Das Systemverhalten wird sichtbar gemacht. Auch wenn der Test in der Regel keine Prüfung der Effizienz oder der Bedienbarkeit einschließt, fallen Mängel in diesen und anderen Punkten doch wahrscheinlich auf.

Nachteile des Testens

- Ein Korrektheitsnachweis durch Testen ist praktisch unmöglich.
- Im Test kann man nicht alle Anwendungssituationen nachbilden, sei es, weil sie sich nicht einfach herbeiführen lassen (z.B. eine bestimmte rasche Folge von Interrupts), sei es, weil die Anwendungssituationen noch nicht bekannt sind.
- Im Test wird nur die Funktionalität geprüft. Eventuell werden (eher zufällig) auch Mängel der Bedienschnittstelle oder des Verhaltens unter Last bemerkt. Alle anderen Eigenschaften des Codes, vor allem die Wartbarkeit, sind völlig ausgeblendet.
- Nur der Code ist dem Test zugänglich, fast alle anderen Dokumente bleiben beim Testen ungeprüft. Testen fördert also die früher übliche Fixierung der Entwickler auf den Code und die Geringschätzung der übrigen Dokumente.
- Der Test zeigt die Fehlerursache nicht. Bis zur Analyse der Testresultate ist nicht einmal klar, ob der Fehler im Prüfling liegt.

Nach Aufwand

- Laufversuch
- Wegwerftest
- Systematischer Test

Nach Komplexität

- Einzeltest
- Modultest
- Integrationstest

Nach Eigenschaft

- Funktionstest
- Installationstest
- Wiederinbetriebnahmetest
- Verfügbarkeitstest
- Last- und Stresstest
- Regressionstest

Nach Rollen

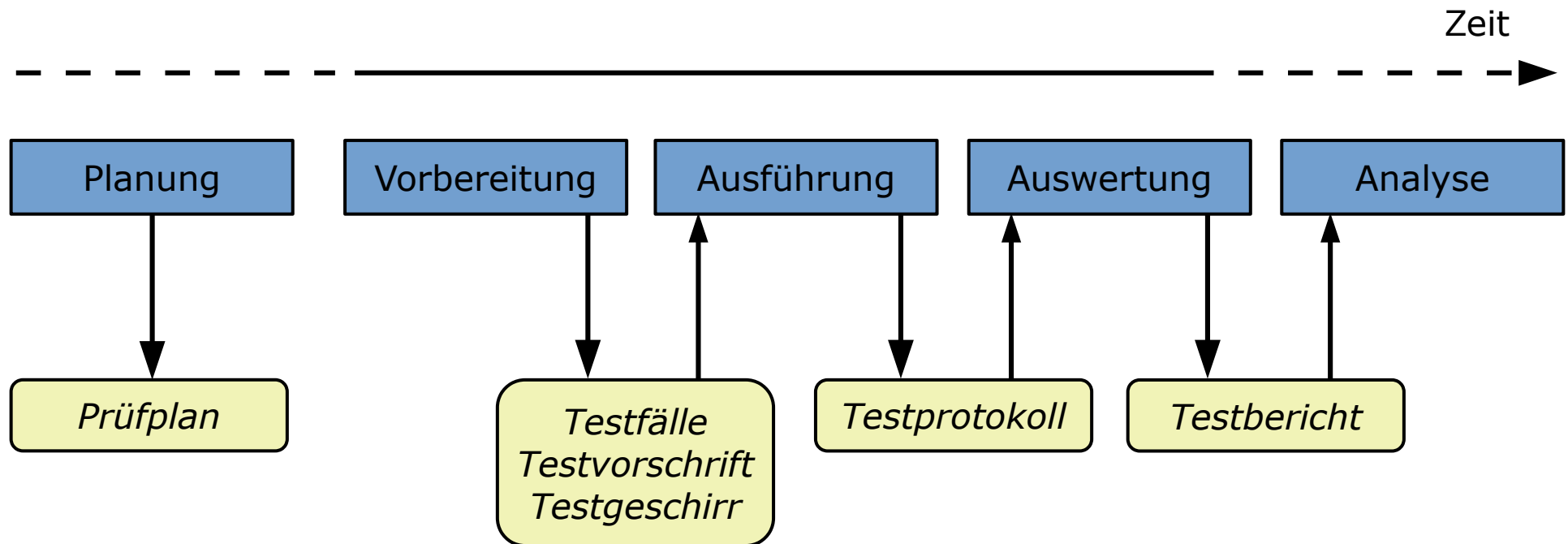
- Alpha- und Beta-Test
- Abnahmetest

Testablauf

Jeder systematische Test besteht im Kern aus den Schritten:

- Vorbereitung
 - Ausführung
 - Auswertung
-
- Planung und Analyse sind nicht minder wichtig, stehen aber getrennt davon an Anfang und Ende
-
- Aufgrund des hohen Aufwands müssen Tests sorgfältig geplant und aufeinander abgestimmt werden

Prinzipieller Testablauf



Testausführung

- Keine spezielle Testvariante des Prüflings!
- Kein Abbruch des Tests, wenn Fehler erkannt wurden!
- Keine Modifikationen des Prüflings im Test!
- Kein Wechseln zwischen Test und Debugging!

Die Einhaltung dieser Regeln hat folgende Vorteile:

- Der Aufwand für den Test lässt sich recht genau abschätzen.
- Der Test deckt die grundsätzlichen Mängel auf.
- Es werden keine kumulativen Korrekturen durchgeführt.
- Das Testgeschirr wird effizient genutzt.
- Der Prüfling wird nicht durch vergessene Zusätze für die Fehlersuche beschädigt.
- Die erzielten Ergebnisse betreffen ein ganz bestimmtes Programm, im letzten Test das später auszuliefernde Produkt.