

Fehlersuche

SoPra 2017

Alexander Kampmann

mit Folien von Andreas Zeller

The Devil's Guide to Debugging

Finde den Defekt durch Raten:

- Versuche gar nicht erst zu verstehen, was das Programm tun soll
- Verstreue überall Debugging-Anweisungen
- Ändere den Code, bis etwas funktioniert
- Mache keine Kopien von alten Versionen

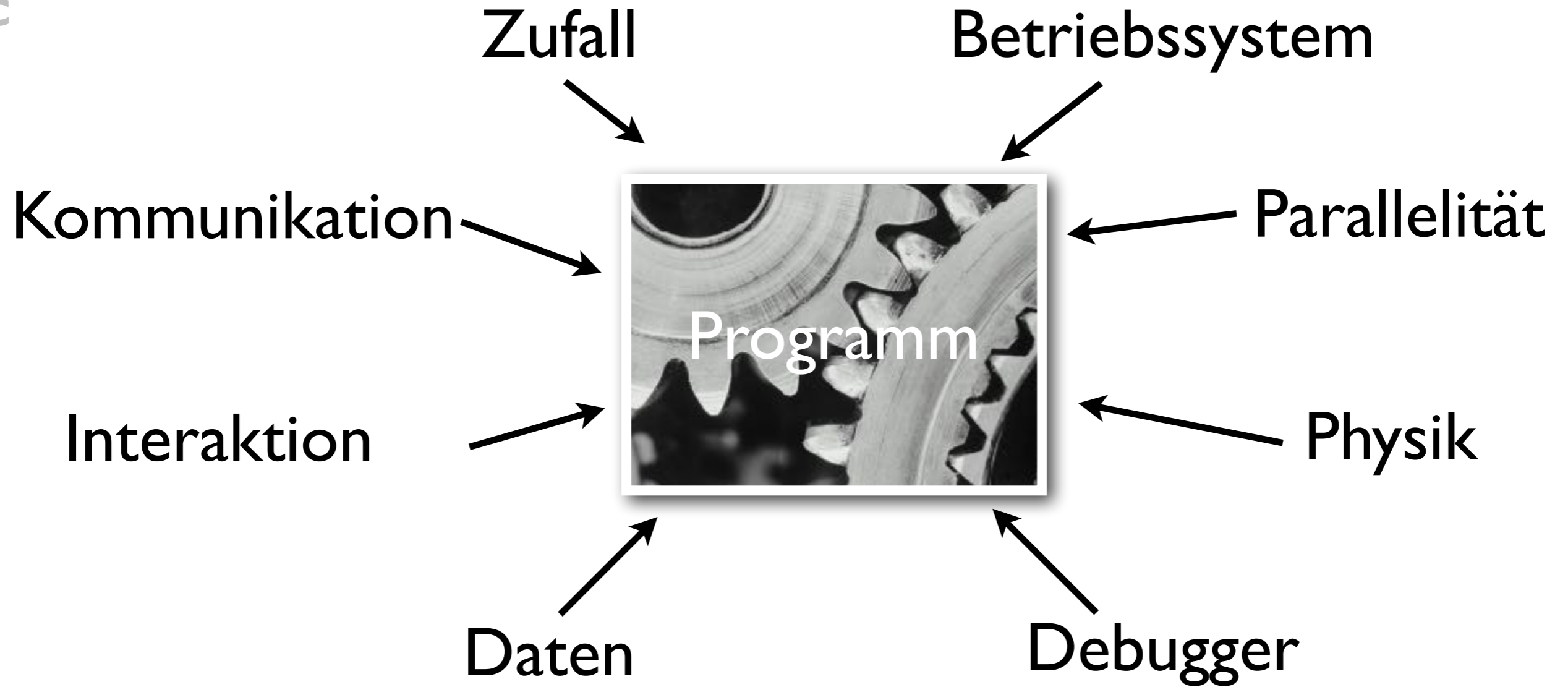
Systematische Fehlersuche

T rack the problem	<i>Problem verfolgen</i>
R eproduce	<i>Reproduzieren</i>
A utomate	<i>Automatisieren</i>
F ind Origins	<i>Ursprünge finden</i>
F ocus	<i>Fokussieren</i>
I solate	<i>Isolieren</i>
C orrect	<i>Korrigieren</i>

Ein guter Bug Report

- Name und Id
- genaue Version (zum Beispiel commit-Id oder Versionsnummer)
- Signifikanz und Priorität
- Reporter
- Assigned to
- Datum des Reports
- Umgebung: Betriebssystem, Hardware, ...
- Beschreibung
- Schritte zum Reproduzieren
- Erwartetes und tatsächliches Ergebnis

Reproduzieren



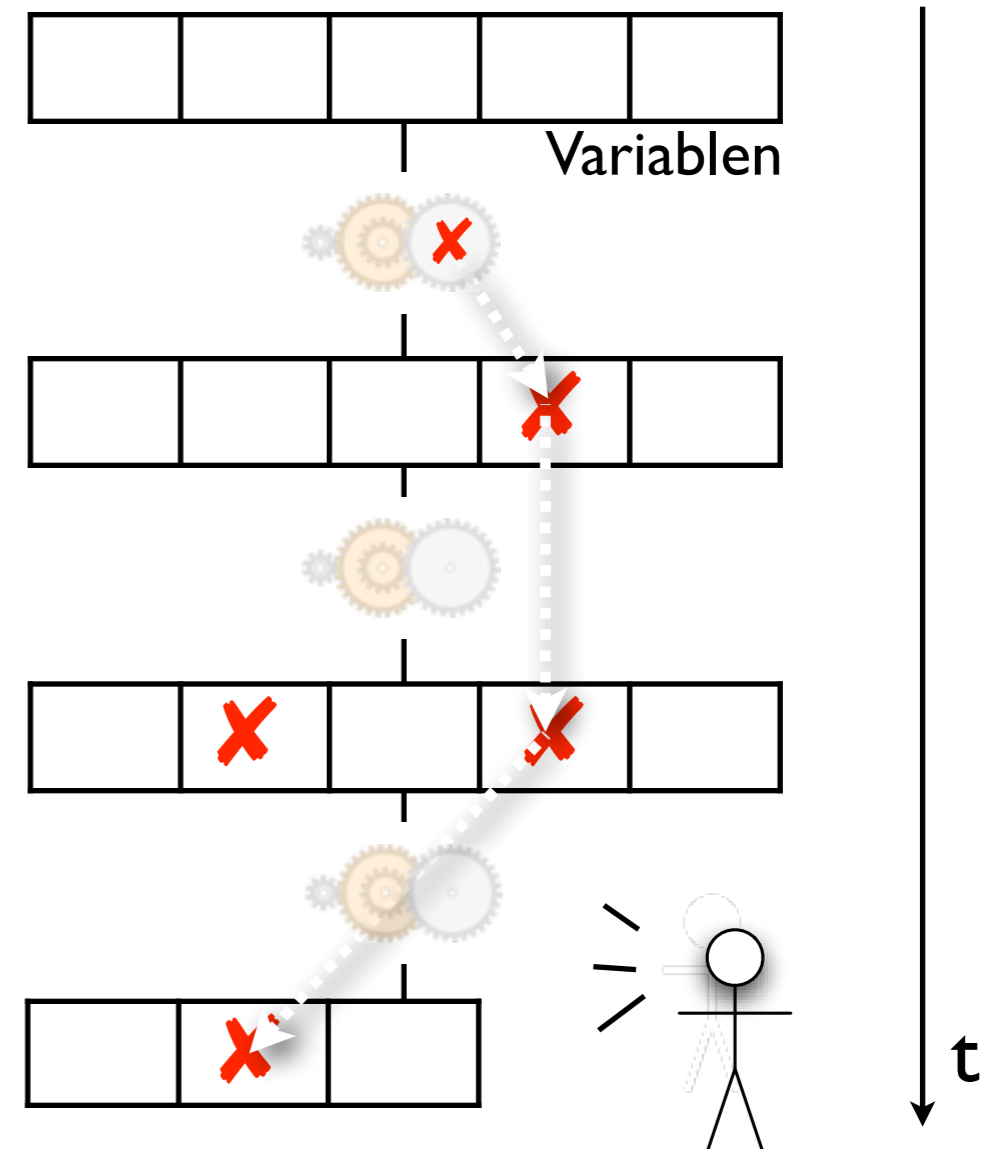
Automatisieren

- Jedes Problem sollte automatisch reproduzierbar sein
- Dies geschieht über geeignete Tests
- Nach jeder Änderung werden die Testfälle ausgeführt

Ursprung finden

1. Der Programmierer erzeugt einen *Defekt* – einen Fehler im Code
2. Der ausgeführte Defekt erzeugt eine *Infektion* – einen Fehler im Zustand
3. Die Infektion breitet sich aus...
4. ...und wird als *Fehlverhalten* sichtbar.

Diese Infektionskette müssen wir *brechen*.



Fokussieren

Bei der Suche nach Infektionen konzentrieren uns auf Stellen im Zustand, die

- *wahrscheinlich falsch* sind (z.B. weil hier früher Fehler aufgetreten sind)
- *explizit falsch* sind (z.B. weil sie eine *Zusicherung* verletzen)

Zusicherungen sind das effektivste Mittel, Infektionen zu finden.

Isolieren

- Fehlerursachen sollen *systematisch* eingeeengt werden – mit Beobachtungen und Experimenten.

Wissenschaftliche Methode

Fehlerbericht

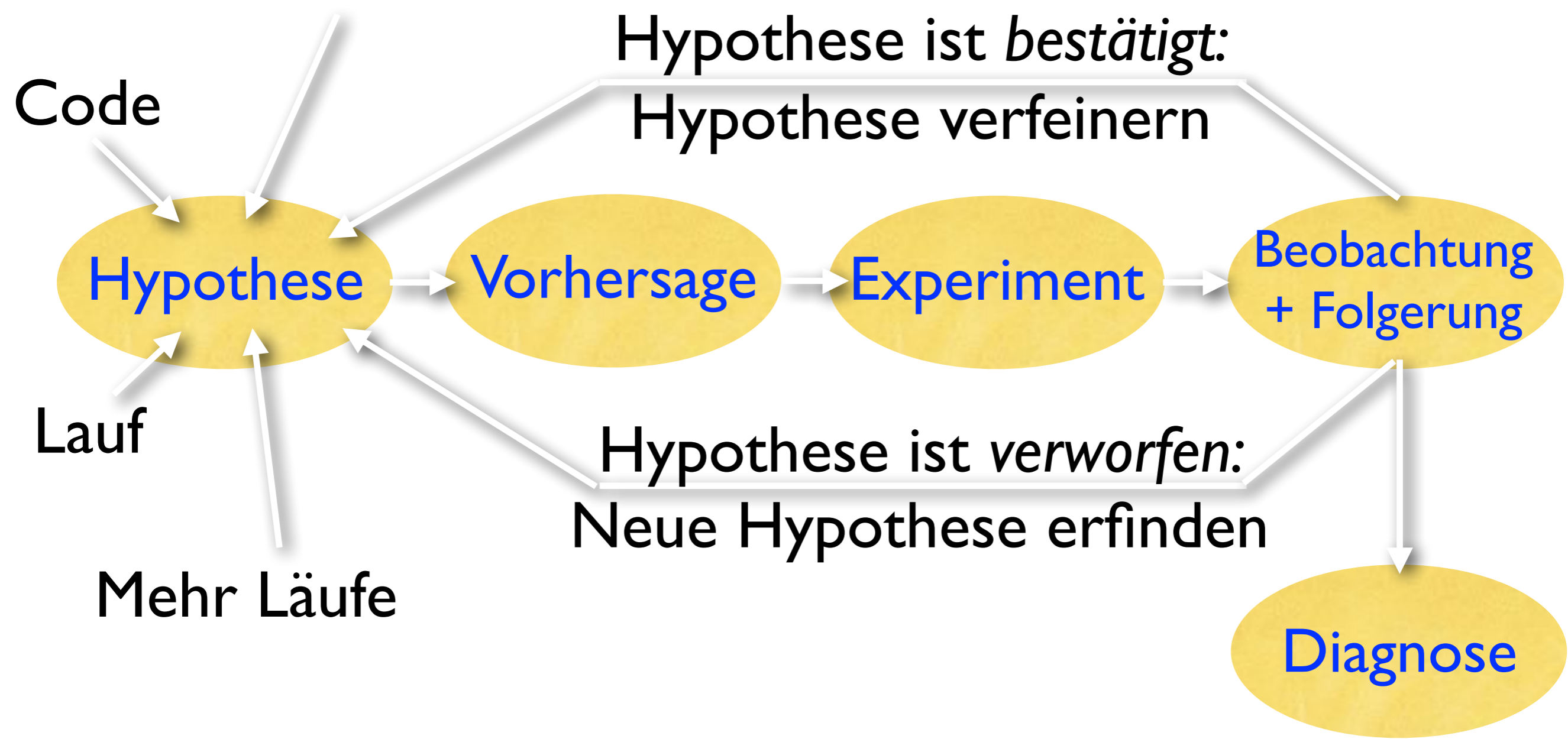
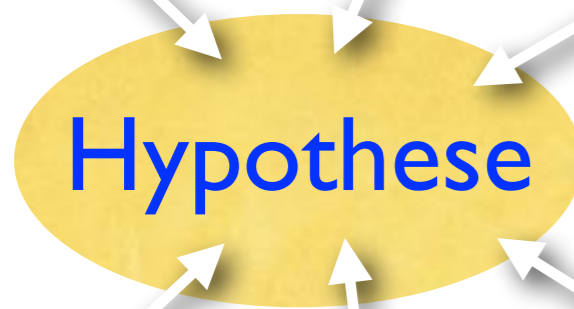
Code

Lauf

Mehr Läufe

Hypothese ist *bestätigt*:
Hypothese verfeinern

Hypothese ist *verworfen*:
Neue Hypothese erfinden



Isolieren

- Wir wiederholen die Suche nach Infektions-Ursprüngen, bis wir den Defekt gefunden haben.
- Wir gehen *systematisch* vor – im Sinne der wissenschaftlichen Methode
- Durch *explizite* Schritte leiten wir die Suche und können sie jederzeit nachvollziehen

Korrektur

Vor der Korrektur müssen wir prüfen, ob der Defekt

- tatsächlich ein *Fehler* ist und
- das Fehlverhalten *verursacht*

Erst wenn beides verstanden ist, dürfen wir den Fehler korrigieren.

Hausaufgaben

- Tritt das Fehlverhalten nicht mehr auf?
(Falls doch, sollte dies eine große Überraschung sein)
- Könnte die Korrektur neue Fehler einführen?
- Wurde derselbe Fehler woanders gemacht?
- Ist meine Korrektur ins Versionsmanagement und Problem-Tracking eingespielt?

Demo

CircleTest(Rennwagen) schlägt fehl

Nützliche Helferlein

- Logging
- Debugger
 - Breakpoint
 - Conditional Breakpoint
 - Exceptional Breakpoint
- Tests
- Assertions
 - Preconditions
 - Postconditions
- Fehlermeldungen