

Trial&Error

Beispiele in Programmierung

- der Klassiker: häufiges Compilieren bei gleichzeitiger Änderung von nur sehr wenig Code, meisten Codeänderungen immer an der gleichen Stelle
- redundante Vorgänge bei nicht-Arbeiten z.B. häufiger Klick auf defekten Link
- kein Überblick über Code vorhanden, z.B. komplexe Verschachtelungen Wichtig: Zurücklehnen und observieren erfolgt erst nach ein paar erfolglosen Versuchen
- Angst vor große Schritten: je besser Annäherung an das Ziel, desto langsamer u. vorsichtiger wird agiert
- Auslagern der Intelligenz auf den Rechner. Bsp: Autovervollständigung in Eclipse: CTRL+Space

Eingabemenge meist begrenzt

- Führt dazu, dass viele Möglichkeiten in kurzer Zeit getestet werden können
- viele Chancen/schritte können allmähliche Annäherung an das Ziel bewirken T&E kann auch als monotone Approximation an ein Ziel verstanden werden

Syntax

- Chance Vorgang des Durchtestens einer Möglichkeit
- most likely Methode, nach der die wahrscheinlichsten Chancen zuerst probiert werden

Fragen

- Ist folgende Methodik auch T&E? Programm Laufen lassen -> testen und evtl. bewusst Fehler erzeugen -> diese Fehler danach analysieren
- T&E scheint SEHR vielfältig und oft im alltäglichen Leben eingesetzt zu werden Bsp: Suchbegriff bei Google eingeben und die wahrscheinlichsten Treffer zuerst besuchen Ist also das gesamte Streben des Menschen auf T&E ausgelegt?

Kategorien von T&E

- laut en.wikipedia.org
 - lösungsorientiert
 - problemspezifisch
 - nicht-optimal
 - kein Vorwissen vorhanden
 - diese vier sind nicht-orthogonal zueinander
- von mir selbst erdachte
 - bewusstes T&E ist als wirkliche Methodik zu verstehen
 - unbewusstes T&E wird von Menschen/Tieren sehr oft durchgeführt evolutionär bedingt
- in Bezug auf Programmierung
 - muss unterschieden werden in:
 - Interaktion zwischen Programmierer und Code
 - Interaktion zwischen Programmierer und Entwicklungsumgebung dieser Punkt wird immer wichtiger, da Entwicklungsumgebungen mehr und mehr eine neue Schicht zwischen Code und Programmierer herstellen
viele T&E Vorgänge bezüglich auf Interaktion mit Code sind so nicht mehr notwendig, werden einem von Entwicklungsumgebung abgenommen
generell werden sehr viel mehr Fehler vermieden als ohne Benutzung einer Benutzungsumgebung