

Übungsgruppe: Qianli Wang und Nazar Sopiha

Unvorsichtigkeit unter Zeitdruck:

- a) Es passiert oft, wenn ich Klausuren schreibe. Falls es nicht viel Zeit aber noch viele Aufgaben bleibt, bearbeite ich dann die Aufgaben nicht so sorgfältig wie früher.
- c) Wenn man unvorsichtig wegen des Zeitdrucks ist, ist es sehr möglich, dass manche hilfreicher Leuten dem helfen würden, was eventuell Zusammenarbeiten oder Zusammenhalten fördern kann. Und mehr wesentlicher Arbeit in kurzer Zeit schaffen
- d) Wenn ein Programm sehr zeit stressig ist und die Entwickler nicht so vorsichtig dafür behandeln, kann es z.B. zum Speicherleck oder zur Unsicherheit führen.

Beibehalten eines einmal eingeschlagenen Weges:

- a) Zum Essen würde ich nur im Supermarkt Produkte kaufen, was ich vorher schon einmal gekauft habe und gut finde. Ich mag also nicht etwas Neues zu probieren.
- c) Konservatismus im Planen ist auch nützlich, wenn wir Entscheidung fällen müssen und Gefahr vermeiden wollen. Wenn wir schon wissen, dass wir schon mal probiert haben, also es existiert einen bestimmten Weg, der zum Ziel erfolgreich erreichen kann. Wir können dann Risiko vermeiden, dadurch dass wir uns wie vorher verhalten, wenn nächstes mal wiederum sehr ähnliche oder gleiche Situation treffen.
- d) Manchmal ist es nicht so passend, einen einmal eingeschlagenen Weg beizubehalten. z.B. Eine Entwicklungsmethode funktioniert aber tatsächlich nicht ganz effizient. Was man machen sollte, ist, dass eigene Entwicklungsmethode sich verbessern lässt, statt dessen man eigene Entwicklungsmethode unverändert bleibt.

Fehlendes Gespür für die Dynamik eines Systems:

- a) Und d) ganz einfaches Beispiel aus dem letzten Semester. Wir sollten 2 Zahlen im Bereich zwischen 10^8 und 10^9 finden, die den gleichen Rest beim Teilen durch 1111 ergeben und diese Zahlen sollen möglichst ähnlich sein (ähnlich heißt hier z.B: 77777 und 70777, also Ziffer Unterschiede). Wir haben einen Algorithmus geschrieben, der alle mögliche Varianten prüft, aber das hatte $O(n^3)$ Komplexität und für so viele Zahlen sollte es zu lange dauern. Im Alltag von Programmierer soll dies oft passieren, Effizienz ist immer die Frage. Kleine uneffiziente Funktionen, die mehrmals verwendet werden, führen skaliert zu großen Problemen.
- c) Manchmal weiß niemand wie sich das System weiter entwickeln würde. Man kann aber bei ersten Schritten, wo das Wachstum nicht so groß ist, profitieren und dann das Ganze beenden. Es kann viel nützlicher, als ein systematischer Ansatz sein, weil z.B. Gewinn beim exponentiellen Wachstum schon nicht so groß sein kann.

Theoretisches, aber nicht eigenständig erfahrenes Wissen führt nicht immer zu entsprechenden Handlungen:

- a) Wenn das Auto ins Schleudern gerät, kann man theoretisch wissen, was man machen sollte, aber falls man noch nie in solchen Situationen war, ist einfache Angst, Stillstand und Nichtstun die einzige Reaktion
- c) Wir sind nie in idealen Bedingungen, sondern in einer realen Welt, wo viele Faktoren eine Rolle spielen und vorgeschriebenes Verhalten führt nicht immer zu den besten

Ergebnisse. Spontane Entscheidungen in echter Zeit können zu anderen Ergebnisse führen, die man nie vorhersagen könnte, weil theoretisch hat man das nicht gerechnet. Und noch Zität dazu: "In theory, there is no difference between theory and practice; in practice, there is." © Lawrence Peter "Yogi" Berra

d) Man kann theoretisch wissen, wie man ein Problem lösen kann, aber fast immer bei der genauer Implementation tauchen nicht besprochene Probleme oder Kleinigkeiten auf und man muss die kämpfen. Diese zu vermeiden ist fast unmöglich, weil Softwareentwicklung sehr komplex ist und man kann leider nicht alle Einzelheiten besprechen.

Falsches Erhöhen von einzelnen Erfahrungen zu allgemeinen Handlungsleitlinien, obwohl wichtige Ausnahmen zu bedenken wären:

a) In vielen großen Städten in China ist die Fahrtrichtung von der linkensten Spur wechselbar. Auf einer Straße, auf der ich fast täglich fahre, habe ich schon daran gewöhnt, dass die linkeste Spur von meiner Richtung gehört, weil es bis damals niemals verändert hat. Und nach der Gewohnheit habe ich die linkeste Spur benutzt. Aber was ganz gefährlich ist, ein PKW wollte den LKW vor ihm gegen meiner Richtung überholen und das PKW hat mich übersehen, weil vor ihm fahrendes LKW zu groß und zu hoch ist, was fast zum Unfall führen.

c) Die einzelne Erfahrungen können uns Möglichkeit anbieten, dass wir in ein unbekanntes Gebiet einsteigen können, wenn wir in diesem Gebiet darüber gar nicht kennenlernen und die gesammelten Erfahrungen bisher noch richtig sind.

d) Greedy-Algorithmus. Im Algorithms betrachtet man immer lokale Extremum und am Ende bekommt trotzdem ein globales Maximum. Aber das Problem ist, dass man nur unter manchen Umständen Greedy Algorithmus benutzen darf. Wir können nicht so daran denken, dass Greedy Algorithmus richtig funktioniert, wenn in vielen Fällen es gut funktioniert. In der Tat ist es nicht so. Also es gilt nicht als eine allgemeine Handlungsleitlinie. Wir müssen natürlich einige Ausnahmen wie negative Kanten betrachten.

Simple Ursachen-Wirkungs-Denken:

a) Ganz einfaches Beispiel: Ich will die Tür öffnen, kann aber nicht. Ich weiß, dass man Kraft dafür braucht. Simple Schlussfolgerung - ich brauche mehr Kraft. Als Folgerung - gebrochenen Griff. Weil etwas aus anderer Seite der Tür stört oder Schlüssel nicht richtig funktioniert oder noch etwas.

c) Nebenwirkungen existieren fast immer und es ist manchmal sehr hilfreich sich an das Ziel zu konzentrieren und alles anderes zu vernachlässigen. Dazu gibt es auch Spruch: "Der Zweck heiligt die Mittel"

d) Man sollte fast immer möglichst global und abstrakt denken. Funktionen oder sogar Architecture Entscheidungen können später für andere Ziele auch angewendet werden und wenn man vorher daran nicht denkt, muss man alles umschreiben. Auch hilft es bei Debugging um schneller zur Lösung zu kommen.

b)

1. C

Begründung: In der Regel darf der Kraftwerk nicht unter 20% gefahren werden. Wegen Nichtverstehen des exponentiellen Verhaltens in solchen Bedingungen hat der Ingenieur übersteuert.

2. D

Begründung: Die Mitarbeiter haben den Reaktor stabilisiert, wussten aber nur theoretisch was weiter passieren kann, weil niemand zuvor aus 7% Leistung gearbeitet hat. Theoretisch könnte man weiterführen.

3. F

Begründung: offensichtlich, alle Pumpen angemacht, aber an Nebenwirkung nicht gedacht.

4. E

Begründung: Um Wasserdruck zu erhöhen - Wassermenge zu erhöhen. Normalerweise funktioniert es, in dem Fall aber nicht, was "Falsches Erhöhen von einzelnen Erfahrung zu allgemeinen Handlungsleitlinien" entspricht.

5. A

Begründung: Tatsächlich entfernten schon mehrere Grafitbremsstäbe aus dem Reaktor. Aber trotzdem haben Mitarbeiter daran dafür gehalten, dass es noch 6 bis 8 Grafitbremsstäbe darin gab. Man sollte sehr schnell entscheiden, ob das Experiment weitergeführt werden soll. Wegen der Tatsache, dass vorher schon ganz viel Arbeit gemacht wurde und unter dem Zeitdruck hat man den Stand des Reaktors als stabil eingeschätzt und weitergeführt

6. B

Begründung: Obwohl der Zustand schon gefährlich war, machte man noch weiter, wie das Experiment geplant wurde.