erschienen in der FIfF-Kommunikation,
herausgegeben von FIfF e.V. - ISSN 0938-3476
www.fiff.de

**Stefan Hügel**

# Kleine Ursache – große Wirkung

## Software-Qualitätsprobleme und ihre Folgen

*Nicht selten haben technologische Erzeugnisse mit Qualitätsproblemen zu kämpfen. Wir ärgern uns über abstürzende Computer, „verschwundene" Dateien oder Mängel der Benutzbarkeit. Doch richtig ernst wird es, wenn Qualitätsprobleme zu Katastrophen führen oder gar Menschenleben kosten.*

Qualität hat unterschiedliche Aspekte. Um qualitativ hochwertige Arbeitsergebnisse zu schaffen, müssen alle Phasen des Entwicklungszyklus betrachtet werden – von der Analyse der Anforderungen bis zum Test des Endprodukts. Außerdem müssen die Managementprozesse funktionieren, die die Produktentwicklung steuern.

In der Geschichte gibt es eine Reihe von Beispielen, wie Qualitätsprobleme in diesen unterschiedlichen Phasen eines Entwicklungsprozesses zu mangelhaften Endprodukten führen. In manchen der zitierten Beipiele hat dies zum Verlust von Menschenleben geführt, in anderen „nur" zu erheblichem wirtschaftlichen Schaden.

Es werden hier drei Beispiele herausgegriffen, davon zwei, bei denen die Schäden auf Softwareproblemen beruhten. Es ist meist schwierig, solche Mängel auf einzelne Ursachen zurückzuführen; die Beispiele wurden aber so gewählt, dass unterschiedliche Hauptursachen für die letztlich entstandenen Schäden verantwortlich waren:

- Mangelhafter Test – Absturz beim Jungfernflug der Ariane V,
- Mangelhafte Benutzbarkeit – Überdosierung beim Bestrahlungsgerät Therac-25 zur Krebsbehandlung,
- Mangelhaftes Management – Explosion nach dem Start des Space-Shuttles Challenger.

### Mangelhafter Test – Ariane V

Ein Softwarefehler, der beim Test unentdeckt geblieben war, führte am 4. Juni 1996 beim Jungfernflug der Ariane V zum Absturz. Was war passiert?

Ca. 37 Sekunden nach dem Start zerstörte sich die Rakete selbst, nachdem sie durch extreme Kurswechsel aerodynamisch stark belastet wurde, und begann auseinanderzubrechen. Damit waren Trägersystem und Nutzlast verloren; der Schaden belief sich auf ca. 500 Mio US-Dollar – einer der bisher teuersten Softwa-

refehler. Wie in solchen Fällen üblich, wurden die Ursachen für den Unfall durch eine Untersuchungskommission analysiert (Lions et al. 1996).

Teile der bereits beim Vorgängersystem Ariane IV verwendeten Software wurden bei der Ariane V wiederverwendet. Sie umfassten auch Teile des Lenksystems. Die Software enthielt Komponenten, die für das Vorgängersystem Ariane IV entwickelt wurden, die aber – aufgrund eines geänderten Startvorbereitungsprozesses – für Ariane V nicht mehr benötigt wurden. Es wurde dennoch entschieden, die Software unverändert zu übernehmen, um weitere Probleme aufgrund einer Softwareänderung zu vermeiden. („Never change a running system.") Da die Funktionalität aber mit keiner Anforderung mehr verbunden war, wurde sie für die Ariane V nicht vollständig getestet. Für Tests anderer Komponenten wurde sie lediglich simuliert. Übersehen wurde dabei, dass die physikalischen Rahmenbedingungen – in diesem Fall die Beschleunigung beim Start – sich bei der Ariane V im Vergleich zur Ariane IV erheblich geändert hatten.

Die höhere Beschleunigung führte dazu, dass in einer Variablen ein Überlauf entstand. Dieser Überlauf wurde nicht durch die Software abgefangen, sondern die Ausnahmebehandlung des Systems setzte ein. Dadurch wurde die entsprechende Komponente heruntergefahren. Dies geschah sowohl im Haupt- als auch im Ersatzsystem, die beide auf der gleichen Software basierten.

Anstatt der erwarteten Steuerinformation wurden nun Diagnoseinformationen an die Steuerung übermittelt. Diese – physikalisch sinnlosen – Daten wurden als Steuerinformation interpretiert. Die Steuerung versuchte, die damit vermeintlich gemeldeten Kursabweichungen auszugleichen, was zu extremen Steuerkorrekturen führte. Die daraus resultierenden Belastungen führten letztlich zur Selbstzerstörung der Rakete.

Wenn der Absturz auch vordergründig durch einen Softwarefehler verursacht wurde, liegt die Ursache letztlich in einem Mangel des Testprozesses. Der Untersuchungsbericht kommt zu dem Ergebnis, dass durch einen ausreichenden Test des Lenksystems der Fehler gefunden worden wäre. Die Untersuchungskommission empfiehlt unter anderem, die gesamte Software mit Blick auf die tatsächlichen Rahmenbedingungen zu überprüfen – insbesondere implizite Annahmen wie die erwarteten Wertebereiche von Variablen. Besonderes Augenmerk muss auf die Testabdeckung gelegt werden.

Der Absturz der Ariane V zeigt, dass bereits ein scheinbar kleiner Fehler – zudem in einer Komponente, die eigentlich gar nicht benötigt wird – zum Totalausfall des Gesamtsystems führen kann.

## Mangelhafte Bedienbarkeit – Therac-25

Therac-25 war ein Linearbeschleuniger für die Strahlentherapie, der vor allem bei Krebserkrankungen eingesetzt wurde. Er ermöglichte Elektronenstrahlen für die oberflächliche Behandlung und Röntgenstrahlung für die Behandlung tiefer liegenden Gewebes. Ab 1983 wurden 11 Therac-25-Geräte installiert. Zwischen 1985 und 1987 kam es zu sechs Unfällen durch erhebliche Strahlenüberdosis, zum Teil mit tödlichem Ausgang.

Die Steuerung von Therac-25 erfolgte durch Software, die im Laufe mehrerer Jahre von einem Programmierer in PDP-11-Assembler entwickelt wurde. Über Ausbildung und Erfahrung dieses Programmierers ist nichts bekannt; Dokumentation des Systems, wie Spezifikation oder Testplan, ist nur wenig vorhanden. Im Gegensatz zum Vorgängermodell Therac-20 wurden beim Therac-25 Sicherheitsfunktionen nicht durch Hardware, sondern mittels Software gesteuert. Fehlfunktionen der Software wirkten sich damit unmittelbar auf die Sicherheit des Systems aus – vorher wurden kritische Betriebszustände durch mechanische Verriegelungen verhindert.

Die Unfälle mit Therac-25 wurden im Wesentlichen durch zwei Softwarefehler verursacht, die mit der Bedienung des Geräts im Zusammenhang standen (Prechelt 2005, Pfeifer 2003):

- Das „Cursor-Up-Problem": Vor der Behandlung wurden Geräteeinstellungen mit einem Cursor eingegeben. Sobald dieser alle Eingaben einer Bildschirmmaske durchlaufen hatte und am rechten unteren Ende des Bildschirms angekommen war, wurde ein Flag „Data Entry Complete" gesetzt. Danach wurden die Parameter der Maschine eingestellt. Dennoch konnten durch Positionieren des Cursors („Cursor-Up") noch Änderungen – bspw. der gewählten Energie – vorgenommen werden, die aber während der 8 Sekunden dauernden Einstellung der Maschine nicht mehr verarbeitet wurden. Da die Umstellung der Betriebsart von einer anderen Routine durchgeführt wurde und keine Konsistenzprüfung mehr stattfand, konnte es zu einer falschen Kombination von Betriebsart und Energie kommen: Der Patient erhielt eine Überdosis.

- Das „Kollimatortest-Problem": Abhängig von einem Flag in der Software wurde die Einstellung des Strahlungsbündlers (Kollimators) geprüft. War dieses Flag = 0, so konnte die Behandlung beginnen. War es ungleich 0, so musste die Einstellung korrigiert werden. Die Routine, die das Flag setzte, wurde bei jeder Behandlung bis zu einigen Hundert Mal aufgerufen. Da bei falscher Einstellung das Flag nicht auf einen Wert ungleich 0 gesetzt (Flag = 1), sondern – aus Speichergründen – inkrementiert wurde (Flag++) konnte die 8-Bit-Variable periodisch überlaufen. Der Wert 0 konnte so zufällig eintreten; die Behandlung konnte beginnen, ohne dass sich das Gerät in der richtigen Position befand.

**Stefan Hügel**

**Stefan Hügel** ist stellvertretender Vorsitzender des FIfF. Er arbeitet als IT-Berater und lebt in München.

Da diese Einstellungen teilweise bedienungsabhängig waren, die Zeit bei den Unfällen eine Rolle spielte und die vom System produzierten Fehlermeldungen unverständlich waren, konnten sie nur sehr schwer reproduziert werden. Das führte dazu, dass die Geräte nach den Unfällen – ohne wirkliche Behebung der Fehler – wieder in Betrieb genommen und dadurch weitere Unfälle verursacht wurden. Die mangelhafte Dokumentation und die unzureichenden Tests des Systems taten ein Übriges.

### Mangelhaftes Management – Challenger

Auch die Qualität der Managementprozesse hat mitunter einen direkten Einfluss auf die Qualität technologischer Produkte. Kritisch wird es dann, wenn Qualitätsprobleme zwar auf der operativen Ebene erkannt werden, auf dem Kommunikationsweg nach oben aber nicht bis zu den Entscheidern durchdringen.

Ein Beispiel dafür ist die Challenger-Katastrophe, bei der im Januar 1986 alle sieben Astronauten ums Leben kamen. Der Ablauf der Katastrophe wurde im Nachhinein rekonstruiert, der Untersuchungsbericht umfasst rund 170.000 Seiten (siehe dazu Steinmann, Schreyögg 1997).

Im Grunde war es ein technisches Problem: Gummidichtungen in den Hauptraketen wurden aufgrund der extrem niedrigen Temperaturen vor dem Start spröde und konnten dem Druck während der Startphase nicht standhalten. Durch ein Leck trat Treibstoff aus und geriet in den Feuerstrahl; dies führte letztendlich zur Explosion.

Dieses Problem wurde vom Hersteller – der Firma *Morton Thokiol* – auch erkannt. Ingenieure der Firma warnten in der entscheidenden Video-Konferenz auf der unteren Entscheidungsebene des Startfreigabeprozesses die NASA auch vor den zu erwartenden Problemen. Doch der zuständige Manager der NASA wollte nichts davon wissen: „The eve of a launch is a hell of a time to be inventing new criteria. My God, Thokiol, when do you want me to launch, next April?“ (zit. nach Steinmann, Schreyögg 1997)

In der weiteren Diskussion zogen sich die Ingenieure immer mehr zurück. Zum Schluss teilten Manager von Morton Thokiol mit, dass man mittlerweile zu einer anderen Einschätzung gekommen war – der Start wurde befürwortet.

In den weiteren Phasen des hierarchisch aufgebauten Startfreigabeprozesses wurden die Bedenken nicht mehr thematisiert. Die kritische Information war „weggefiltert“ worden, der Start fand letztendlich statt. In diesem Beispiel ist klar erkennbar, dass Qualität nur mit der Unterstützung des Managements möglich ist. Und hier wird deutlich, dass Qualitätsfragen immer auch ethische Fragen sind.

### Ethik

Dies zeigt auch ein geradezu klassisches Beispiel der Ethik, wie es in (Lenk 1987) zitiert wird. Von einem amerikanischen Fahrzeughersteller wurde ein Kleinwagen in großer Eile entwickelt. Das Fahrzeug war so ungünstig konstruiert, dass bei einem Auffahrunfall häufig der Benzintank aufgerissen wurde; aufgrund des auslaufenden Benzins kam es dann mit hoher Wahrscheinlichkeit zu einem Brand. Eine Nachbesserung wurde vom Hersteller abgelehnt, da seine Kosten-Nutzen-Analyse ergab, dass die Kosten der Nachbesserung höher liegen würden als Schadensbegleichung und Prozesskosten aufgrund der zu erwartenden Unfälle (dabei wurden jährlich 180 Todesopfer zugrunde gelegt).

### Weitere Fälle

Die beschriebenen Fälle sind natürlich nur Beispiele für eine ganze Reihe bekannt gewordener Softwarefehler und Qualitätsprobleme. Großen Imageschaden für Intel verursachte beispielsweise der Pentium-Bug, der bei Divisionen zu Rundungsfehlern führte (Knieschewski 2003, FIfF 1995). Bekannt wurden auch die Probleme beim Gepäckabfertigungssystem am Flughafen von Denver (Glass 1998, Walber 2003), bei dem Softwarefehler dazu führten, dass Fließband und Gepäckwagen nicht synchronisiert waren und in der Folge Koffer nicht aufgefangen wurden, Wagen aus den Schienen sprangen und Gepäck an der falschen Stelle abgeladen wurde. Auch hier spielten zu enge Terminpläne und in der Folge eine zu kurze Testphase eine Rolle – am Flughafen München, der über ein ähnliches System verfügt, erfolgten vor der Inbetriebnahme umfangreiche Tests. Als Softwarefehler kann sicherlich auch das „Y2K-Problem“ gesehen werden, bei dem Softwaresysteme nicht auf den Jahrtausendwechsel vorbereitet waren und umfangreiche Nacharbeiten notwendig wurden.

### Fazit

Die Beispiele zeigen, wie schnell auch scheinbar kleine Qualitätsprobleme zu erheblichen Folgen führen können – von wirtschaftlichem Schaden bis zum Verlust von Menschenleben. Dabei sind die Fehlerquellen vielfältig: Sowohl technische Fehler als auch Bedienungsfehler – letztlich verursacht durch ungenügende Voraussage des Nutzerverhaltens – können solche Folgen verursachen. Häufig spielen zu enge Terminpläne eine Rolle – der Test des Systems ist davon meist am stärksten betroffen. Das Beispiel der Challenger-Katastrophe zeigt dabei auch, dass letztendlich die Akzeptanz im Management von entscheidender Bedeutung ist.

## Literatur


FIfF (1995): Schluss-PFIfF – Q&A: The Pentium FDIV Bug. FIfF-Kommunikation 1/1995

Simson Garfinkel (2005): History's worst Software Bugs. Wired, http://www.wired.com/software/coolapps/news/2005/11/69355 (Abruf 2. August 2008)

Robert L. Glass (1998): Software-Runaways – Lessons learned from massive Software Project Failures. Upper Saddle River: Prentice Hall

Sebastian Knieschewski (2003): Berühmt berüchtigte Softwarefehler – Der Pentium-Division-Bug. Seminar Dr. Beckert, Universität Koblenz, http://www.uni-koblenz.de/~beckert/Lehre/Seminar-Softwarefehler/Ausarbeitungen/knieschewski.pdf (Abruf 2. August 2008)

Hans Lenk (1987): Ethikkodizes für Ingenieure. In: Hans Lenk, Günter Ropohl (Hg.): Technik und Ethik. Stuttgart: Reclam

Jacques-Louis Lions et al. (1996): Ariane 5 Flight 501 Failure. Report by the Inquiry board. http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html (Abruf 2. August 2008)

Martin Pfeifer (2003): Berühmt berüchtigte Softwarefehler – Therac-25. Seminar Dr. Beckert, Universität Koblenz, http://www.uni-koblenz.de/~beckert/Lehre/Seminar-Softwarefehler/Ausarbeitungen/pfeifer.pdf (Abruf 2. August 2008)

Lutz Prechelt (2005): Vorlesung "Anwendungssysteme" - Sicherheit: Therac-25. Freie Universität Berlin, http://www.inf.fu-berlin.de/inst/ag-se/teaching/V-AWS-2005/22kurz_Sicherheit.pdf, basierend auf Nancy Leveson, Clark Turner (1993): An investigation of the Therac-25 accidents, IEEE Computer (Abruf 2. August 2008)

Horst Steinmann, Georg Schreyögg (1997): Management – Grundlagen der Unternehmensführung. 4. Auflage. Wiesbaden: Gabler

Tina Walber (2003): Berühmt berüchtigte Softwarefehler – London Ambulance Dispatch System und Gepäcktransport am Flughafen Denver. Seminar Dr. Beckert, Universität Koblenz, http://www.uni-koblenz.de/~beckert/Lehre/Seminar-Softwarefehler/Ausarbeitungen/walber.pdf (Abruf 2. August 2008)

Wikipedia.de, Stichwort Ariane 5 (Abruf 2. August 2008)

Wikipedia.de, Stichwort Therac-25 (Abruf 2. August 2008)