

Proseminar Software Disaster

ARIANE 5
Absturz des Flugs 501

Mathias Riedl

4.12.2002

Gliederung:

- 1. Einleitung: Rückschlag für Raumfahrt**
- 2. Allgemeine Beschreibung des Vorfalls**
 - a. Ablauf des Countdowns
 - b. Erste Fluganalyse
 - c. Bergung von Material und Daten
- 3. Auswertung der Abläufe im technischen System**
 - a. Beschreibung der Rakete
 - b. Auswertung der technischen Abläufe
- 4. Gründe für den Softwarefehler**
 - a. Ungeschützte Variablen
 - b. Fehlerhafte Spezifikation
 - c. Verwendung der Ariane 4 Software
- 5. Tests und Überprüfungen**
 - a. Durchführung einer Simulation fehlt
 - b. Systemtests nur mit Modulen
- 6. Verbesserungsmaßnahmen**
- 7. Zusammenfassung**
- 8. Literaturverzeichnis**



1 Einleitung – finanzieller Rückschlag für europäische Raumfahrt

Am 4. Juni 1996 explodierte die Ariane 5 auf ihrem Jungfernflug. Die unbemannte Rakete sollte vier identische Satelliten in den Orbit befördern. Durch die Explosion entstand ein Schaden von 1,7 Milliarden DM. Die Entwicklungszeit für die Rakete betrug 10 Jahre und die Entwicklungskosten beliefen sich auf 11,8 Milliarden DM.

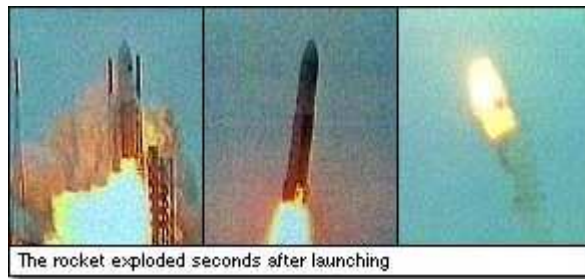
Der erste kommerzielle Flug der Ariane 5 fand erst am 10. Dezember, also 3 Jahre später, statt.

Nach dem Absturz wurde unverzüglich eine Untersuchungskommission eingerichtet, die den Vorfall untersuchen sollte. Sie sollte feststellen was genau geschehen war und was die Gründe für das Scheitern der Mission waren.

2 Beschreibung des Vorfalles

2.a Ablauf des Countdowns

Die Witterungsbedingungen an diesem Tag waren normal. Es gab kein Risiko eines Blitzeinschlages, da das gemessene elektrische Feld im Abschussgebiet vernachlässigbar klein war. Die Sichtverhältnisse waren das einzige Unsicherheitskriterium an diesem Tag. Deswegen wurde der Countdown auch 7 Minuten vor Start angehalten und um eine Stunde verschoben, da die Sichtverhältnisse nicht den Anforderungen entsprachen. Die Sichtbedingungen verbesserten sich in der Zeit wie vorhergesagt und die Zündung der Raketen konnte erfolgen.



2.b Erste Fluganalyse

Bis 36 Sekunden nach dem Start tauchten keine Unregelmäßigkeiten auf. Die Rakete hielt ihre Flugbahn ein und sendete normale Telemetriedaten zur Bodenstation. Dann jedoch traten die ersten Fehler auf. Hier eine Auflistung in chronologischer Reihenfolge:

- Das „back-up Inertial Reference System (SRI)“ schaltete sich ab. Wenig später versagt auch das aktive „Inertial Reference System“.
- Die Rakete wird durch Steurdüsen zu einer Flugbahnänderung veranlasst. Dies bewirkte eine schnelle abrupte Drehung der Rakete
- 42 Sekunden nach dem Start wurde die Selbstzerstörung veranlasst, weil die äußeren Kräfte auf die Rakete zu stark wurden und sie auseinander zubrechen drohte.

Man erkannte sehr schnell, dass der Fehler im Flugkontrollsystem zu suchen war, genauer im Trägheitsnavigationssystem (SRI). Diese zwei Einheiten hatten fast gleichzeitig ihre Arbeit eingestellt.

2.c Bergung von Material und Daten

Es wurden noch folgende Daten zur Analyse des Absturzes herangezogen:

- Telemetriedaten, die bis 42 Sekunden nach dem Start zur Bodenstation gesendet wurden
- Flugbahndaten von Radarstationen
- Visuelle Beobachtungen (Infrarotkameras, Filme)
- Untersuchung der geborgenen Materialien

Auf Grund der Explosion der Rakete verstreute sich das ganze Material über eine Fläche von 12 Quadratkilometern. Das machte es schwierig alle wichtigen Überreste zu bergen. Es gelang jedoch die zwei Navigationssysteme zu bergen und aus ihnen wichtige Daten zu erhalten. Von besonderem Interesse war das aktive System, welches sich später abschaltete. Es besaß daher einige Informationen, die nicht durch Telemetriedaten zugänglich waren.

3 Auswertung der Abläufe im technischen System

3.a Beschreibung der Rakete

Das SRI, das sogenannte Trägheitsnavigationssystem, ist dafür zuständig, die aktuellen Flugdaten zu ermitteln und an den Hauptcomputer weitergeben. In diesem System kam es auch zu dem verhängnisvollen Fehler. Das SRI hat einen eigenen Rechner, der Winkel und Geschwindigkeiten berechnet. Die Daten erhält das SRI von einer Trägheitsplattform mit Laser-Gyroskopen und Geschwindigkeitsmessern.

Die errechneten Werte werden dann über einen Datenbus an den On-Board-Computer (OBC) weitergegeben. Der OBC ist sozusagen der Hauptcomputer der Rakete, der das Flugprogramm ausführt, die Flugbahn berechnet und die Steurdüsen der Triebwerke steuert.

Um die Zuverlässigkeit des ganzen Systems zu verbessern hat man eine Redundanz auf Hardwareebene eingebaut. Jede Hardwarekomponente existierte in doppelter Ausführung an Bord der Rakete. So gab es zwei, in Hardware und Software identische, Navigationssysteme (SRI). Eines davon arbeitete in „hot“ Stand-by-Modus, während das andere aktiv war. Hätte der OBC erkannt, dass das aktive SRI einen Fehler verursacht hat, hätte der Hauptcomputer sofort auf das Ersatzsystem umgeschaltet, vorausgesetzt dieses funktioniert richtig.

Es muss noch erwähnt werden, dass der größte Teil der Software von der Ariane 4 übernommen wurde und nahezu identisch auf der Ariane 5 zum Einsatz kam.

3.b Auswertung der technischen Abläufe während des Starts

Auf Basis des Datenmaterials, welches zur Verfügung stand konnte folgender Ablauf der Vorgänge rekonstruiert werden:

- Der Wert für die horizontale Geschwindigkeit war viel höher als erwartet, denn die Flugphase der Ariane 5 ist eine andere als bei der Ariane 4. Dies resultierte in eine beträchtlich höheren Horizontalgeschwindigkeit
- Der Operandenfehler passierte in einer internen Kalibrierungsfunktion („alignment function“). Es gab einen unerwartet hohen Wert im sogenannten BH, Horizontal Bias, der die Geschwindigkeitswerte ausgibt, die von der Trägheitsplattform gemessen wurden.
- Die Kalibrierungsfunktion ist noch 40 Sekunden nach dem lift-off aktiv. Diese Zeitspanne war für die Anforderungen der Ariane 4 festgelegt worden, wäre aber bei der Ariane 5 nicht mehr nötig gewesen.
- Der Fehler tauchte in einem Teil des Systems auf, welches zur Kalibrierung der Plattform diente. Die Funktion lieferte nur vor dem Start wichtige Werte, aber sobald die Rakete abhob, hatte die Funktion keinen Sinn mehr.
- Der Softwarefehler wurde durch eine Konvertierung eines 64-bit floating point Wertes in 16-bit integer Wert verursacht. Der Wert der Gleitpunktzahl hatte einen größeren Wert als man mit einer 16-bit Integerzahl hätte darstellen können. Daraufhin folgte ein Operandenfehler. Obwohl andere vergleichbare Variablen gegen einen Überlauf geschützt waren, hatte man an dieser Stelle darauf verzichtet.
- Der Hauptcomputer konnte nicht auf das Ersatzsystem umschalten, da dies wenige Sekunden zuvor, aus demselben Grund, schon abgestürzt war.
- Kurz bevor sich das aktive SRI abschaltete sendete es auf Grund eines Softwarefehlers noch ein Diagnosebitmuster an den OBC.
- Der On-Board-Computer interpretierte dieses Bitmuster jedoch als Flugdaten und gab daraufhin den Befehl einer unnötigen Flugbahnänderung.
- Durch die totale Änderung der Flugbahn drohte die Rakete, wegen zu großen aerodynamischen Kräften, auseinander zubrechen. Die Selbstzerstörung der Rakete wurde daraufhin eingeleitet.

Da sich das SRI abschaltete wäre die Rakete auch ohne das falsch interpretierte Bitmuster abgestürzt, da ohne SRI keine Flugdaten mehr beim OBC ankommen würden. So war die Mission nach dem Fehler im SRI zum Scheitern verurteilt.

4 Gründe für den Softwarefehler

Ada-Programm des Trägheits-Navigationssystems (Ausschnitt):

```
...
declare
  vertical_veloc_sensor: float;
  horizontal_veloc_sensor: float;
  vertical_veloc_bias: integer;
  horizontal_veloc_bias: integer;
  ...
begin
  declare
    pragma suppress(numeric_error, horizontal_veloc_bias);
  begin
    sensor_get(vertical_veloc_sensor);
    sensor_get(horizontal_veloc_sensor);
    vertical_veloc_bias := integer(vertical_veloc_sensor);
    horizontal_veloc_bias := integer(horizontal_veloc_sensor);
    ...
  exception
    when numeric_error => calculate_vertical_veloc();
    when others => use_irs1();
  end;
end irs2;
```

4.a Ungeschützte Variablen

Nun hatte man also den Fehler im Navigationssystem gefunden. Es stellte sich aber die Frage wie so ein Fehler überhaupt passieren konnte. Der primäre technische Grund hierfür war das Fehlen eines Schutzes der Konvertierung der Variable BH. Dadurch schaltete sich das SRI ab und die Rakete war sozusagen orientierungslos.

Ziel war es, die Systemauslastung des SRI-Computers auf 80% zu halten. Deswegen hat man auch nicht alle Variablen auf einen Überlauf überprüft. Um eventuelle Probleme bei dem ungeschützten Code auszuschließen, wurde eine Analyse für jede Variable, die eine Exception verursachen könnte, durchgeführt. Die Gefahr eines Überlaufs bestand bei sieben Variablen, dabei wurden vier geschützt, während drei ungeschützt blieben. Bei den drei ungeschützten Variablen meinte man, dass sie entweder physikalisch limitiert waren und deswegen keinen Überlauf verursachen konnten, oder man hatte einen großen Sicherheitsspielraum gelassen, so dass die Werte die obere Grenze nie erreichen konnten. Jedoch wurden für die Analyse keine Flugdaten der Ariane 5 verwendet, sondern ging von Werten der Ariane 4 aus. Es sollte auch erwähnt werden, dass die Entscheidung einige Variablen nicht zu schützen in Übereinstimmung mit den Projektpartnern in mehreren Verträgen erfolgte.

4.b Fehlerhafte Spezifikation

Es wurde übereinstimmend beschlossen, in die Spezifikation für das SRI keine Flugbahndaten der Ariane 5 aufzunehmen. Des Weiteren wurden in der Spezifikation die Beschränkungen in den Operationen weggelassen, die die obere Grenze der Variablen verdeutlicht hätte. Es gibt keine Anhaltspunkte das aktuelle Flugbahndaten benutzt wurden, um das Verhalten der ungeschützten Variablen zu testen. Die Aufnahme der Grenzen der Variablen, hätte geholfen Unstimmigkeiten mit der Flugbahn der Ariane 5 zu erkennen. (Abstract Interpretation)

Desweiteren trug auch die Spezifizierung des exception-handling zum Scheitern der Mission bei. Falls es zu irgendeiner Art von Softwarefehler käme, sollte der Fehler auf den Datenbus angezeigt werden und der Fehlerkontext in einem EEPROM-Speicher abgelegt werden. Anschließend sollte sich der fehlerhafte SRI-Computer abschalten. Die Entscheidung die Prozessoroperation einzustellen war fatal. Ein Neustart des Prozessors machte keinen Sinn, weil es zu schwer gewesen wäre die Höhe neu zu berechnen. Der Grund für dieses drastische Vorgehen liegt in der Natur des Ariane-Programms, bei dem man nur von zufälligen Hardwarefehlern ausging. Aus dieser Sicht wurde nur ein exception-handling für zufällig auftretende Hardwarefehler kreiert, das leicht verständlich durch eine Back-up-System abgefangen werden konnte.

4.c Verwendung der Ariane 4 Software

Die Entscheidung, die Kalibrierungsfunktion auch nach dem Start weiterlaufen zu lassen, wurde vor mehr als 10 Jahren für die früheren Modelle der Ariane beschlossen. Dabei wollte man einen eher unwahrscheinlichen Countdownabbruch bewältigen. Deswegen wurde die Zeitspanne von 50 Sekunden (40 Sekunden nach lift-off) für das Fortsetzen des Programms nach dem Start eingeführt. Bei einem Abbruch hätte dann die Bodenstation in der Zeit, wieder die Kontrolle über die Rakete übernehmen können. Hätte man das ganze Programm von neuem starten müssen, hätte es bei dem Vorgängermodell, der Ariane 4, zu lange, etwa 45 Minuten, gedauert neu einzukalibrieren. Dabei war die Gefahr zu groß, dass das Startfenster, in dem die Rakete starten kann, sich schließt und eine Verlegung des Starts nach sich ziehen könnte.

Diese Sonderausstattung wäre aber auf der Ariane 5 nicht nötig gewesen, da diese eine andere Vorbereitungsphase hat, aber trotzdem wurde dieses Programm beibehalten. Man wollte eine gut funktionierende Software nicht verändern, solange es sich nicht als unbedingt notwendig erwies.

Man vertrat auch die Ansicht, dass die Software als fehlerfrei erachtet werden sollte, solange das Gegenteil nicht bewiesen war. Jedoch sollte man es genau andersherum sehen. Man sollte Software erst als korrekt erachten, wenn alle möglichen Tests und Verifizierungsmethoden ergeben haben, dass die Software ordnungsgemäß funktioniert.

5 Unvollständige vorherige Tests

5.a Durchführung einer Flugsimulation fehlt

Es gab noch eine Möglichkeit, im Vorfeld den Fehler zu entdecken und zwar während der unzähligen Tests, die mit dem System gemacht wurden. Jedoch deckten hier die Tests nicht alle Bereiche ab.

Als man jede einzelne Komponente testete, achtete man beim SRI nur auf Umweltfaktoren. Nur äußere Einflüsse auf das System wurden getestet. Es wurde kein Test durchgeführt, der überprüfen sollte, ob das System während des Countdowns und der Flugphase richtig funktioniert. Man hätte das Navigationssystem testen können, indem man simulierte Beschleunigungswerte eingibt und die Bewegung der Rakete nachstellt. Dieser Test hätte schon beim Zulieferer der Komponente passieren sollen. Jedoch wie schon oben erwähnt, standen in der Spezifikation keine Flugbahndaten, somit war es auch für den Hersteller des SRI nicht notwendig diese Art von Tests durchzuführen.

5.b Systemtests nur mit Softwaremodulen

Ein Großteil der Tests wurde in einer speziellen Einrichtung durchgeführt, der Functional Simulation Facility (ISF). In den Tests, in denen auch die Flugbahn und Flugdaten der Ariane 5 berücksichtigt wurden, kam jedoch das tatsächliche SRI nicht zum Einsatz. Das Navigationssystem wurde durch zwei speziell entwickelte Softwaremodule simuliert.

Es gab zwei Möglichkeiten das tatsächliche SRI in die Tests mit einzubeziehen:

- Man hätte einen Tisch benötigt, der in alle drei Richtungen beweglich ist, um das Laser-Gyroskop zu simulieren. Um die Beschleunigungswerte zu bekommen, hätte man extra Geräte anfertigen müssen, welche diese Werte simulieren.
- Man hätte auch beide Messwerte, Beschleunigungs- und Bewegungswerte, durch Simulationen ersetzen und in das SRI einspeisen können.

Die erste Möglichkeit ist ziemlich teuer, aber man könnte eine sehr genaue Simulation des Fluges durchführen. Die zweite Alternative wäre billiger, aber die Durchführung hängt sehr stark von der Genauigkeit der Simulation ab. In beiden Fällen würde aber ein großer Teil der Elektronik und die gesamte Software getestet.

Zu Beginn hatte man sich auch darauf geeinigt, die zweite Methode bei den Tests zu verwenden. Später wurde diese Entscheidung jedoch geändert. Aus

folgenden Gründen wurde das tatsächliche SRI aus den Tests genommen und durch Module ersetzt:

- Das SRI sei schon ausreichend getestet worden und deshalb sei es nicht mehr nötig das tatsächliche SRI nochmals zu testen.
- Die Navigationssoftware des On-Board-Computers hängt maßgeblich von den Messungen des SRI ab. In der Testeinrichtung (ISF), könnte diese Präzision nicht durch die Elektronik, welche die Signale erzeugt, erreicht werden.
- Das Simulieren von Fehlern ist nur mit den Modellen möglich und nicht mit den richtigen Geräten.
- Die Taktzeit des SRI ist eine Millisekunde, während die Taktzeit der Simulation 6 Millisekunden beträgt. Dies würde zu weiteren Ungenauigkeiten in der Simulation führen.

6 Verbesserungsmaßnahmen

Nachdem alle Fehler aufgedeckt worden waren, wurden folgende Verbesserungsvorschläge für das Arianeprogramm gemacht:

- Man schaltet die gesamte Software, die während dem Start nicht benötigt wird, ab. Darunter fällt auch die Kalibrierungsfunktion.
- Es darf nicht mehr vorkommen, dass sich ein wichtiges System abschaltet und keine Daten mehr sendet.
- Softwarefehler in Betracht ziehen.
- Die Flugdaten mit in die Spezifikation und Testanforderungen aufnehmen.
- Die maximalen Werte jeder Variable im System ermitteln. (Abstract Interpretation)
- Mehr Telemetriedaten von Fehlern übermitteln, so dass geborgenes Material nicht so wichtig ist.
- Externe Personen bei der Überprüfung der Spezifikation und des Codes mit einbeziehen.

7 Zusammenfassung

Der Absturz der Ariane 5, auf ihrem Jungfernflug, wurde durch einen Operandenfehler in einem Unterprogramm des Navigationssystems verursacht. Die Gründe für diesen Fehler sind jedoch im Vorfeld zu suchen. Die Spezifikation der Ariane 5 war unvollständig. Die Tests waren unzureichend und deckten nicht alle zu testenden Bereiche ab. Man wollte eine hohe Genauigkeit bei den Tests erzielen, jedoch vernachlässigte man dabei in wie weit andere Komponenten, wie das SRI, sich während eines Fluges verhalten würden. Man sollte jedoch auch erwähnen, dass durch die unzähligen Tests viele andere Fehler ausgebessert wurden und es nicht einfach ist, diese Art von Programmfehler aufzuspüren. Allerdings wurden die Grenzen der Software nicht vollständig analysiert und man erkannte nicht, dass die Tests unzureichend waren, um einen Operandenfehler festzustellen.

8 Literaturverzeichnis

Prof. J. L. Lions: ARIANE 5, Flight 501 Failure, Report by the Inquiry Board

<http://java.sun.com/people/jag/Ariane5.html>

<http://www.wikiservice.at/dse/wiki.cgi?Ariane5Absturz>

Presseveröffentlichungen der ESA:

N° 33-1996: Ariane 501 - Presentation of Inquiry Board report

http://www.esa.int/export/esaCP/Pr_33_1996_p_EN.html

„Softwarebug ließ Ariane abstürzen“, Artikel in der Rhein -Zeitung

http://rhein-zeitung.de/on/96/07/24/topnews/fehler_1.html

<http://www-aix.gsi.de/~giese/swr/ariane5.html>

<http://www3.informatik.tu-muenchen.de/studienberatung/info-tag/huckle.pdf>

<http://archive.eiffel.com/doc/manuals/technology/contract/ariane/page.html>