NASA: Teurer Totalverlust durch simplen Softwarefehler

Es war wohl ein Softwarefehler, der im Oktober 2016 zum Absturz der Marssonde ‚Schiaparelli' führte. Das Radar-Höhenmessgerät und das Navigationsprogramm der Sonde haben nach Aussagen des ESA-Direktors „nicht richtig miteinander kommuniziert".

Wahrscheinlich ist, dass die Sonde den Landefallschirm bereits in vier Kilometern Höhe absprengte. Danach prallte sie ohne Fallschirm mit einer Geschwindigkeit von **ca. 300km/h ungebremst** auf die Mars-Oberfläche.

Die Schäden, die sich hieraus ergaben, waren nicht nur der Totalverlust der Sonde, sondern auch zeitliche Verzögerungen im ESA-Marsprogramm. Ein neuer Versuch kann erst 2020 unternommen werden. Hinzu kommt, nach mehreren gescheiterten Missionen, ein nicht zu beziffernder Schaden in der Reputation der Mars-Programme:

- 1993 Verlust des eine Milliarde Dollar teuren ‚Mars Observer'
- 1999 Verlust der Sonde ‚Climate Orbiter' durch einen „peinlichen Rechenfehler"
- 2001 Verlust der Sonde ‚Polar Lander' auch durch einen Umrechnungsfehler, der zum verfrühten Abschalten der Bremstriebwerke führte
- 2003 Verlust der Sonde ‚Beagle 2'

Bei solchen Raumfahrtmissionen drohen der Verlust von Milliarden von Forschungsgeldern sowie eine jahrelange Verzögerung des wissenschaftlichen Erkenntnisprozesses.

Wie hätte das Softwaretesten nach ISTQB® dies verhindern können?

- Komponententests: Testen der Komponenten auf korrekte Funktionalität, d.h. korrekte Berechnungen und Messungen.
- Integrationstests: Prüfung der Kommunikation einzelner Module oder Komponenten miteinander. In dieser Teststufe werden insbesondere Umrechnungsfehler gefunden, z.B. von km/h im europäischen Umfeld in mph im amerikanischen Gebrauch, die beim Review der Spezifikation nicht gefunden wurden.
- Code Reviews, manuelle Reviews und statische Analyse zum Auffinden „peinlicher Rechenfehler" (ESA, 1999, Verlust der Sonde ‚Climate Orbiter'), Umrechnungsfehler und zur Qualitätssicherung der Spezifikationen und des Codes
- Risikobasiertes Testen: Welche Phasen sind die risikoreichsten bei Gebrauch der Software? (z.B. Manöver- und Landephasen)
- Error Guessing und erfahrungsbasiertes Testen
- Initialisierung eines Testmanagements
- Pflege einer standardisierten Testdokumentation, z.B. nach IEEE 829
- Lessons Learned zur langfristigen Verbesserung und Optimierung der (Test-) Prozesse. Hiermit wird u.a. verhindert, dass sich (gleichartige) Fehler wiederholen.

qSkills GmbH & Co. KG
Süd-West-Park 65
D-90449 Nürnberg

Tel. +49 (0)911 80 103-31
info@qskills.de
www.qSkills.de