

Software-Qualitätssicherung, Teil I: Grundlagen des Softwaretests

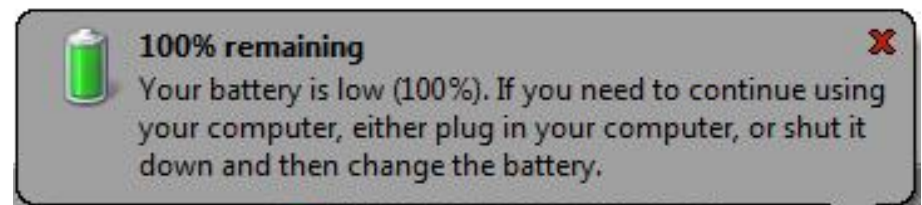
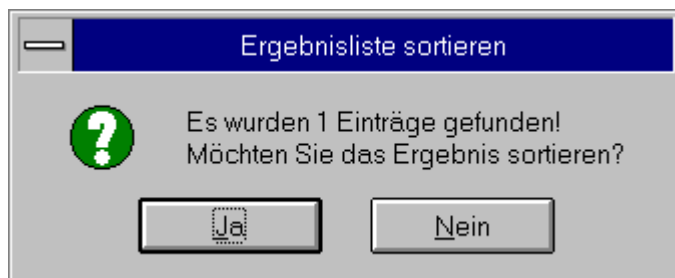
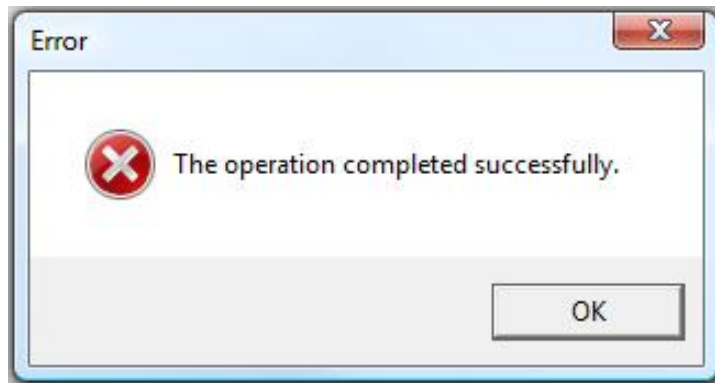
Friedrich-Schiller-Universität Jena, Wintersemester 2017/2018
Ronny Vogel, Xceptance GmbH

Agenda

- Einführung
- Klassifikation von QS-Maßnahmen
- Testfallerstellung
- Ökonomie des Testens
- Systematische Testfallermittlung
- Testausführung
- Fehlerverfolgung
- Kurzüberblick Testmanagement

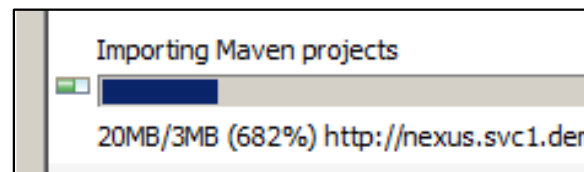
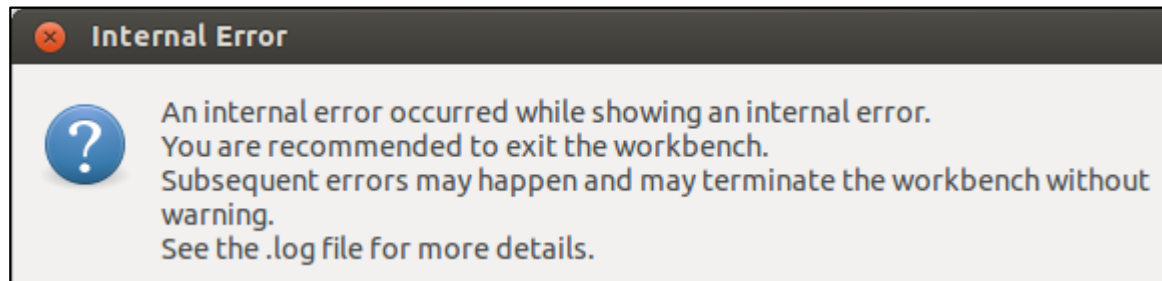
Einführung >

Notwendigkeit von Softwaretests (1)



Einführung >

Notwendigkeit von Softwaretests (2)



The "Select View" control allows the user to change the view they would like to pick up the items at. When the user clicks this control a lightbox is displayed similar to product pages.

Einführung >

Notwendigkeit von Softwaretests (3)

- **1962:** Verlust der Venus-Sonde Mariner 1 (80 Mio. \$) durch fehlenden Bindestrich
- **1971:** Rund um die Erde explodieren 72 von 141 Wetterballons bei einem Experiment aufgrund eines falsch interpretierten Satelliten-Kommandos.
- **1982:** Die Vertauschung der Steuerung zwischen Höhenruder und Seitenruder beim Bau eines F-117-Tarnkappenbombers führte zum Absturz des Flugzeugs.
- **1985:** Die Software eines Roboters bei GM verarbeitete keine Information von schwarzen Autos, die die Montagehalle daher ohne Windschutzscheibe verließen.
- **1985-1987:** Mehrere Verletzte und drei Tote durch Bestrahlungsgerät Therac-25
- **1989:** Innerhalb einer halben Stunde wurden 2000 Mio. Pfund an Kunden in England und USA doppelt überwiesen. Zinsverlust trotz Rückzahlung: bis 0.5 Mio. Pfund/Tag.
- **1991:** Signalsystem verursacht Zusammenstoß zweier Züge in Japan mit 42 Toten
- **1992:** Ein Programm der British Telecom zur Annahme von Notrufen spielte nur "Bitte warten" vor. Einige Anrufer verstarben während dieser Wartezeit.
- **1993:** Ein Airbus A320 rollte in Warschau über die Landebahn hinaus, da das Aquaplaning auf der Landebahn falsch berechnet wurde.
- **1995:** Massive Verzögerungen im bundesweiten Zugverkehr durch einen Stack Overflow in der Software eines Hamburger Stellwerks

Einführung >

Notwendigkeit von Softwaretests (4)

- **1996:** Sprengung des Prototyps der Ariane-5 kurz nach dem Start, weil der von der Ariane-4 wiederverwendete Code aufgrund der jetzt höheren Beschleunigung in einen Fehler bei einer Typumwandlung lief
- **1997:** die „USS Yorktown“ trieb bei einem Manöver drei Stunden hilflos im Mittelmeer, da ein Mitarbeiter einen fehlerhaften Sensorwert in der Datenbank mit 0 überschrieb; es kam zu einer Division durch Null, die nicht korrekt abgefangen wurde; der temporäre Speicher lief in den Speicherbereich des Antriebssystems über; das Netzwerk aus mehreren Windows-NT-Servern fiel aus
- **1999:** Verlust der NASA-Sonde „Mars Climate Orbiter“ beim Marsanflug, weil der Schub in Pfund x Sekunde statt in Newton x Sekunde berechnet wurde (125 Mio. \$)
- **2003:** Verzögerungen bei Toll Collect mit Vertragsstrafen und Einnahmeausfällen in Milliardenhöhe, u.a. aufgrund der fehlenden Kompatibilität von Software-Modulen
- **2005:** Im russischen Plessezsk führte ein Programmfehler zum Fehlstart einer Trägerrakete und zum Verlust des Satelliten CryoSat.
- **2005:** Durch mangelnde Umstellungsmöglichkeiten der Software für das ALG II überwies die BfA 300 Mio. € zu viel an die Krankenkassen, die 20% davon für den zusätzlichen Verwaltungsaufwand behielten.

Einführung >

Notwendigkeit von Softwaretests (5)

- **2006:** Ein missglücktes Annäherungsmanöver zweier US-Satelliten führte zu einer veränderten Umlaufbahn mit 1 Mio. \$ Schaden. Ursache waren Softwarefehler in einem der Satelliten und im GPS-System.
- **2007:** Zehn Personen der südafrikanischen Armee kamen aufgrund eines Programmfehlers in einem vollautomatisierten 35-mm-Flakgeschütz ums Leben.
- **2008:** Gepäcksystem des Heathrow Airport's Terminal 4 zwei Tage offline aufgrund von Softwareproblemen; Verzögerungen und Beschwerden tausender Passagiere
- **2009:** Ein Airbus A380 musste aufgrund von Problemen mit dem Navigationssystem umkehren und zum Ausgangsflughafen in New York zurückfliegen.
- **2010:** Die Bezahlung und das Holen von Bargeld mit deutschen Girocard- und Kreditkarten waren bei etwa 30 Millionen Karten nicht möglich. Ursache war ein Chip, dessen Software Fehler bei der Verarbeitung der Jahreszahl 2010 aufweist.
- **2010:** Die Amerikanerin Louise Chavez gewann an einem Münzautomaten in Colorado knapp 43 Millionen Dollar. Laut Casinobetreiber handelte es sich um einen Software-Fehler. Eine Untersuchung wurde eingeleitet, das Casino sprach ihr \$20 zu.
- **2010:** Viele kleine Rückbuchungen wurden bei der britischen TUI Travel über Jahre nicht korrekt verbucht. Die TUI AG musste nun 120 Millionen Euro abschreiben. Der Finanzchef trat zurück, das Eigenkapital wurde um diese Summe gesenkt.

Einführung >

Notwendigkeit von Softwaretests (6)

- **2011:** Honda rief weltweit etwa 2,5 Millionen Autos in die Werkstätten zurück, da ein Software-Problem zu Schäden am Automatik-Getriebe führen konnte.
- **2011:** Fehlerhafte Software eines 2,7 Mrd. USD teuren Aufklärungssystems behinderte die US-Armee in Afghanistan und im Irak. Es versagte selbst bei einfachen Aufgaben und reagierte verzögert bei mehreren parallelen Nutzern.
- **2011:** Hackerangriff auf das Playstation-Netzwerk von Sony; Daten von mehr als 70 Millionen Nutzern gestohlen
- **2011:** Hacker haben Daten von hunderttausenden US-Kunden der US-Großbank Citigroup ausgespäht
- **2012:** 45min lang überflutete ein neues Programm des Finanzdienstleisters "Knight Capital Group" die Wall Street mit fehlerhaften Handelsaufträgen; die Firma blieb auf vielen zu teuer gekauften Aktien sitzen; 440 Millionen Dollar Verluste
- **2012:** Ein Fehler in der WLAN-Software bestimmter Telekom-Router ermöglichte jedermann das unbemerkte Eindringen in das Netzwerk.
- **2012:** Das Messsystem verweigerte bei Olympia 2012 die Annahme der Weite von Hammerwerferin Betty Heidler, da es aufgrund des gleichen Messwerts der Werferin vor ihr einen Fehler der Kampfrichter zugrunde legte. Die Medaille war gefährdet.

Einführung >

Notwendigkeit von Softwaretests (7)

- **2012:** Ein Fehler im Internet Explorer ermöglichte Angreifern, Computer unter ihre Kontrolle zu bringen, das BSI warnte.
- **2013:** Ein Kommando flutete das Steuerungssystem der Europäischen Elektrizitätsnetze mit Daten; das führte beinahe zum Zusammenbruch der Elektrizitätsnetze in Österreich; möglicher Blackout in halb Europa
- **2013:** Eine Sicherheitslücke im Heizungssystem Vaillant EcoPower 1.0 ermöglichte Angreifern das Ein- und Ausschalten und die Beschädigung über das Internet
- **2013:** Probleme an der Motorsteuerung des Opel Insignia können zu Motorschäden führen. Opel ruft 61.000 Autos zurück.
- **2013:** Chrysler ruft eine halbe Million Jeeps zurück, da durch ein Softwareproblem die Gangschaltung unbeabsichtigt verstellt werden kann.
- **2013:** Durch Softwareprobleme war drei Stunden lang kein Handel an der New Yorker Tech-Börse NASDAQ möglich. Amerikas gesamter Tech-Markt (2712 Aktienwerte) war nicht handelbar, auch nicht an anderen Börsen.
- **2013:** Xerox-Multifunktionsgeräte ersetzen beim Scannen als PDF (ohne OCR-Texterkennung) klein gedruckte Ziffern willkürlich durch andere, was z.B. in Bauplänen zu falschen Zahlenangaben führte.

Einführung >

Notwendigkeit von Softwaretests (8)

- **2013:** „Heartbleed“-Sicherheitslücke in der Open-Source-Bibliothek „OpenSSL“; private Daten über verschlüsselte Verbindungen können ausgelesen werden
- **2013:** monatelange Wartezeiten und nicht auffindbare Teile im neuen, von IBM und SAP implementierten Ersatzteil-Logistiksystem „ATLAS“ bei BMW
- **2014:** Ein Softwarefehler legt die Luftverkehrskontrolle um Los Angeles lahm; der Flugplan eines Militärflugzeugs in 20 km Höhe wurde fälschlich mit zivilen Flügen koordiniert, obwohl sich die Routen aufgrund verschiedener Höhen nicht kreuzen
- **2014:** Toyota ruft weltweit 1,9 Millionen Prius III zurück; durch einen Softwarefehler kann es zur Überlastung der Elektronik kommen, das Fahrzeug schaltet in ein Notprogramm und wird langsamer; im Extremfall kann es sogar stehen bleiben
- **2014:** „ShellShock“-Sicherheitslücke in der Unix-Shell „bash“; beliebiger Schadcode über das Netz ausführbar; bedroht sind vor allem Webserver
- **2014:** ein Fehler in einem Tool von RepricerExpress setzt in UK den Preis hunderter Produkte im Amazon Marketplace auf 0,01 Pfund; immense Verluste einiger Anbieter
- **2015:** ein Fehler bei der Preismstellung führte für 2h zu nur 6 ct/l bei Super Plus an 30 Tankstellen um Nürnberg; die zentralen Rechner zeigten den korrekten Preis

Einführung >

Notwendigkeit von Softwaretests (9)

- **2015:** eine bestimmte Tastenkombination beim Roulette in Merkur-Spielautomaten führte immer zu hohen Gewinnen; 10 Mio. € Verlust in einer Nacht; Tage vorher wurde der Softwarefehler vom Hersteller entdeckt und die Spielhallen-Betreiber gewarnt, aber diese hatten keine Zeit mehr, zu reagieren
- **2015:** administrativer Zugriff auf Hosting-Pakete und Domains mit einem beliebigen Passwort im Reseller-Kundenportal von 1und1.de, gmx.de, web.de durch SW-Fehler
- **2015:** potentieller Strom- und Steuerungsausfall in Boeing 787 durch Überlauf eines Zählers in der Generatorsteuerung nach 248 Tagen mit Abschaltung des Generators; vorgeschriebener Neustart pro Maschine alle 120 Tage als temporärer Workaround
- **2015:** Sicherheitslücke bei BMW; mehr als 2 Millionen Fahrzeuge hätten unberechtigt geöffnet werden können
- **2015:** Android-Sicherheitslücke „Stagefright“ ermöglicht das Übernehmen und Fernsteuern hunderter Millionen Smartphones durch manipulierte MMS oder Videos
- **2015:** Absturz eines Airbus A400M in Spanien bei einem Testflug; kurz nach dem Start erhielten drei der vier Triebwerke widersprüchliche Befehle und drosselten die Leistung, vermutlich durch fehlerhaft installierte Software zur Triebwerkssteuerung; vier Tote und zwei Schwerverletzte

Einführung >

Notwendigkeit von Softwaretests (10)

- **2015:** ein Jeep Cherokee wurde bei einer Demonstration remote gehackt und ferngesteuert, u.a. Bremsen und Motor; Chrysler ruft 1,4 Millionen Fahrzeuge zurück
- **2015:** im US-Bundesstaat Washington wurden von 2002 und 2015 bis zu 3200 Gefängnisinsassen zu früh entlassen; die genaue Zahl ist nicht bekannt; einige Gefangene wurden kurz danach wieder eingesperrt
- **2016:** die Software der Gepäck-Förderanlage am Flughafen Düsseldorf erkannte den 29. Februar nicht als Tag; etwa 1200 Koffer blieben dadurch liegen; Mitarbeiter mussten das Gepäck von Hand sortieren
- **2016:** das 286 Millionen USD teure japanische Weltraumteleskop Hitomi zerbrach etwa einen Monat nach Inbetriebnahme bei einem Ausrichtungsmanöver im Orbit durch einen Softwarefehler in der Lageregelung
- **2016:** beim Online-Banking von Comdirect sahen Kunden Kontodaten und Postfächer fremder Nutzer nach einem Software-Update
- **2016:** Absturz der Marssonde "Schiaparelli"; sie warf schon beim Landeanflug die Bremsfallschirme ab und schaltete die Bremstriebwerke zu kurz ein, wahrscheinlich durch Überlastung einer Messkomponente, die zeitweilig nicht reagierte, so dass falsche Höhenangaben berechnet wurden (unterhalb der Marsoberfläche)

Einführung >

Notwendigkeit von Softwaretests (11)

- **2017:** Fehlerhaftes Firmware-Update legt über WLAN steuerbare Türschlösser von „RemoteLock“ lahm; keine Öffnung mehr per Tastenkombination; kein Update mehr möglich; die Hardware muss getauscht werden
- **2017:** das cloudbasierte „Smart Garden System“ von Gardena löscht in der Haupturlaubszeit Daten einiger Benutzer, auch Backups und lokale Daten in den gesteuerten Geräten; Neueinrichtung erfordert physischen Zugriff auf die Geräte
- **2017:** ein Krimineller stahl in den USA jahrelang aus mehr als 100 Hotels Wertgegenstände; er nutzte eine Sicherheitslücke in den Schlössern von Onity mit einem selbstgebauten Arduino-basierten Gerät über den zugänglichen Diagnoseport
- **2017:** Firmware-Update für unsichere Herzschrittmacher und Heim-Basisstationen von Abbot nötig; fast 500.000 Patienten betroffen; Sicherheitslücken ermöglichten Konfigurationsänderungen und die Ausführung von Befehlen über Funk
- **2017:** beim Drucken von PDF-Dateien mit dem Browser Microsoft Edge erscheint unter bestimmten Umständen statt "1234" auf dem Ausdruck die Ziffernfolge "1144"

Einführung >

Notwendigkeit von Softwaretests (12)

Beobachtung

- potentielle Auswirkungen von Softwarefehlern nehmen offenbar eher zu statt ab

Einerseits: Potentiell höhere Softwarequalität

- Anstrengungen und Fortschritte in der ingenieurmäßigen Erstellung von Software in den Prozessen, Werkzeugen, usw., ermöglichen höhere Qualität

Andererseits: Potentiell höhere Fehlerauswirkungen

- Software wird essentieller Bestandteil von immer mehr Lebensbereichen
- Software wird immer umfangreicher (Lines of Code), Projekte werden größer, die Anzahl der Beteiligten und deren lokale Verteilung steigt
- immer kürzere Time-To-Market neuer Software
- immer höhere Frequenz von Änderungen in Betrieb befindlicher Software
- immer größerer Einflussbereich einzelner auf Software basierender Systeme, z.B. Online-Systeme mit Millionen oder sogar über 1 Milliarde Nutzer

Einführung >

Notwendigkeit von Softwaretests (13)

Fehler sind trotz aller Anstrengungen nicht vermeidbar!

- komplexe Systeme sind nicht vollständig durch Menschen erfassbar
 - Vielzahl von Eigenschaften und Kombinationen nicht überschaubar
 - Ursache-Wirkungs-Beziehungen nicht mehr überschaubar
- komplexe Technologien, komplexe fachliche Fragen, viele Beteiligte, ...

Gründe sind unter anderem

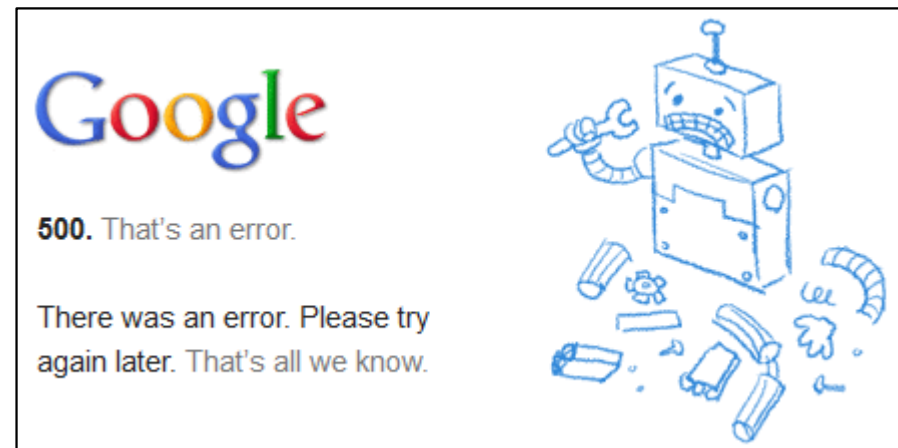
- Kommunikationsprobleme
 - extern, zwischen Menschen
 - intern, bei der Transformation von Sprache in Verständnis
- begrenztes Gedächtnis
- fachliches Problemverständnis

→ Tests sind unbedingt notwendig!

Fehlerbegriff (1)

Fehlerwirkung („failure“)

- auch: „äußerer Fehler“, „Fehlfunktion“
- Wirkung eines Fehlerzustands, die bei der Ausführung des Testobjekts nach außen in Erscheinung tritt
- Abweichung des Ist-Verhaltens vom Soll-Verhalten
- Beispiele:
 - System reagiert nicht mehr auf Eingaben
 - unerwarteter Programmabbruch
 - unter hoher Last sieht ein Nutzer Daten fremder Nutzer
 - fehlende Fehlermeldung
 - falsche Ausgabe



Fehlerbegriff (2)

Fehlerzustand („fault“)

- auch: „innerer Fehler“, „Defekt“
- Ursache für einen äußeren Fehler
- Zustand einer Softwarekomponente, der unter spezifischen Bedingungen eine geforderte Funktion beeinträchtigen kann
- Beispiele:
 - falsche Anweisung im Code
 - fehlende Fehlerbehandlung
 - Race Condition
 - inkorrekte Datendefinition



© Giant Microbes

Fehlerbegriff (3)

Fehlhandlung einer Person („error“)

- menschliche Handlung, die zu einem falschen Ergebnis führt, beispielsweise zu falschem Code
- Beispiele:
 - inkorrektes Verständnis einer Anforderung
 - fehlerhafte Programmierung durch Copy&Paste von Codeteilen
 - Einfügen einer falschen Vergleichsoperation in den Code
 - Entwickler vergisst Fehlerbehandlung
 - Einchecken von Änderungen in einen falschen Code-Zweig



Copyrights

Alle für die Vorlesung zur Verfügung gestellten Unterlagen unterliegen dem Copyright und sind ausschließlich für den persönlichen Gebrauch im Rahmen der Vorlesung „Qualitätssicherung von Software“ freigegeben. Die Weitergabe an Dritte und die Nutzung für andere Zwecke sind nicht erlaubt.