

In der Zwischenzeit

A Tester's Mindset (System Test)

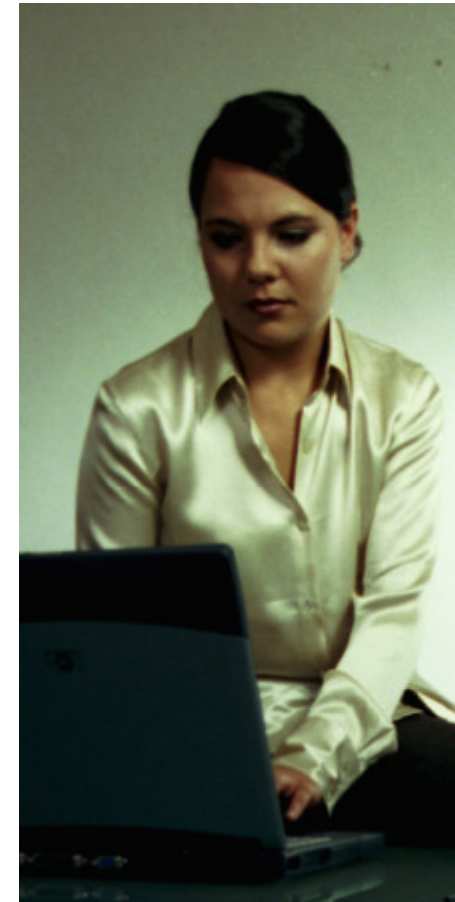
Hans Hartmann

Hans.hartmann@objentis.at

OBJENTIS Software Integration GmbH

Die Agenda

- Einleitung
- Eine falsche Grundannahme
- Konsequenzen für die TesterInnen
- Eine neue Herangehensweise
- Die veränderte Grundeinstellung
- Das hohe Maß an intellektueller Verantwortlichkeit



About OBJENTIS



Objentis arbeitet auf den Gebieten

Banking

- **Versicherung**
- **Industrielle und Telekom-Bereiche**
- **Öffentliche Administrationsbereiche**

Objentis unterstützt mit/bei

- **Software-Test und QA-Beratung**
- **Testwerkzeugen**
- **Datenanalyse, Migration und Szenarienanalyse**
- **Entwicklerunterstützung für Lotus Notes®-basierter Software (KENTUMI)**
- **Training und Fortbildung**
- **ISTQB®, IREB®**

About the Speaker



SALES

Age 16



DEV

Age 22



DEV

Age 25



DEV

Age 28



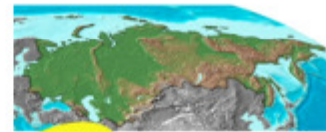
DEV

Age 31



DEV

Age 33



SALES

Age 35



SALES

Age 37



DEV

SALES

EDU

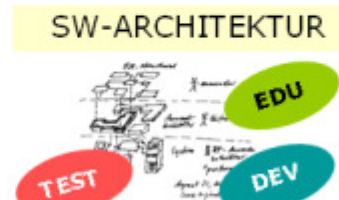
Age 43



TEST

EDU

Age 47



EDU

TEST

DEV

Age 51



TEST-ARCHITEKTUR

Age 56

Einleitung – Die wesentlichen Fragen

Woher kommen wir?

- Wissenschaft
- Religion
- Philosophie

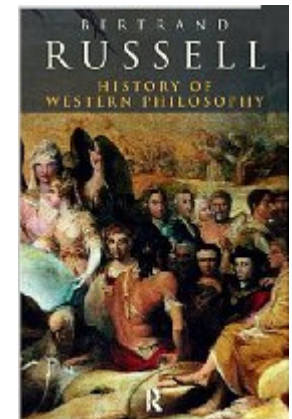


In der Zwischenzeit ...

- Warum gibt es uns?
- Was sollten wir heute essen?
- **Was testen wir inzwischen?**

Wohin gehen wir?

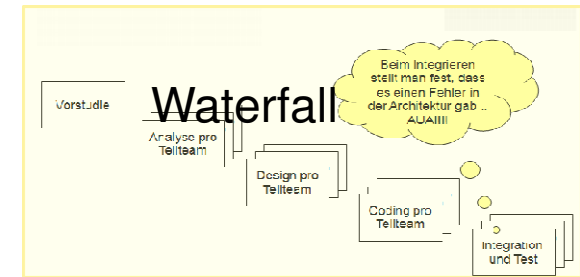
- Wissenschaft
- Religion
- Philosophie



Was bedeutet Zwischenzeit?

Was bedeutet Zwischenzeit

- Eine falsche Annahme

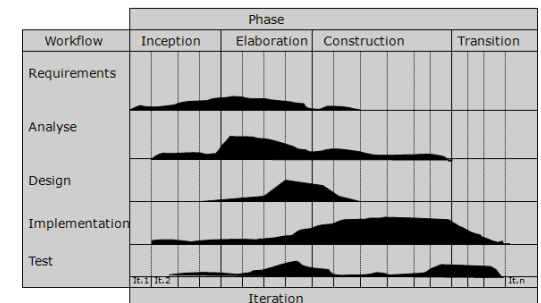


Entwicklung

Testen

Oder besser...

- Ein idealer Plan



Entwicklung

Testen (statisch und dynamisch)

Die realistische „Zwischenzeit“

Was heißt Zwischenzeit

- Ein realer Fall

Irgendwas wie Waterfall

Entwicklung

Testen



Unterschiedlicher
Hintergrund

Unterschiedliche
Ausbildung



Test Team

Unterschiedliche Ideen
über das Testen

Die falsche Grundannahme

Warum gibt es Fehler in den Programmen?

- Entwickler arbeiten schlampig
- Die Anforderungen: unvollständig und fehlerhaft
- Unrealistische Zeitvorgaben vom Vorstand



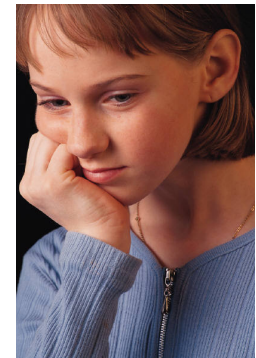
Könnten wir die Fehler verhindern? ... durch

- Kent Beck, Erich Gamma, Ralph Johnson
- Space Rocket Lab Crew
- Der Papst

**Nein
wegen
X**

Wofür steht der X-Faktor?

- Größe
- Komplexität
- Mathematik



Größe, Komplexität und Kurt Gödel



Unvollständigkeitssatz 1931 veröffentlicht

- **Ein konsistentes System kann nicht vollständig sein.**
- **Die Konsistenz von Axiomen kann nicht innerhalb des Systems bewiesen werden.**

Gültigkeit

- In jedem axiomatischen System, welches ausreichend stark ist um die Arithmetik von (z.B.) natürlichen Zahlen zu beschreiben (z.B. Peano-Axiome oder ZFC)

Quelle 1931

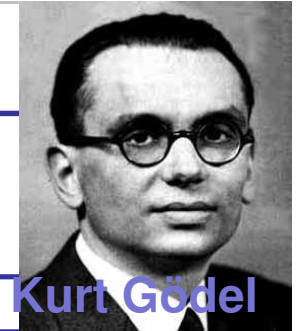
- *"Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme"*
- *"On formally undecidable propositions of Principia Mathematica and related systems"*

A False Assumption

Der Satz von Gödel

Leseempfehlungen

- Gödel, Escher Bach: an eternal golden braid, Douglas R. Hofstadter (dt.)
- Infinity and the Mind, Rudy Rucker



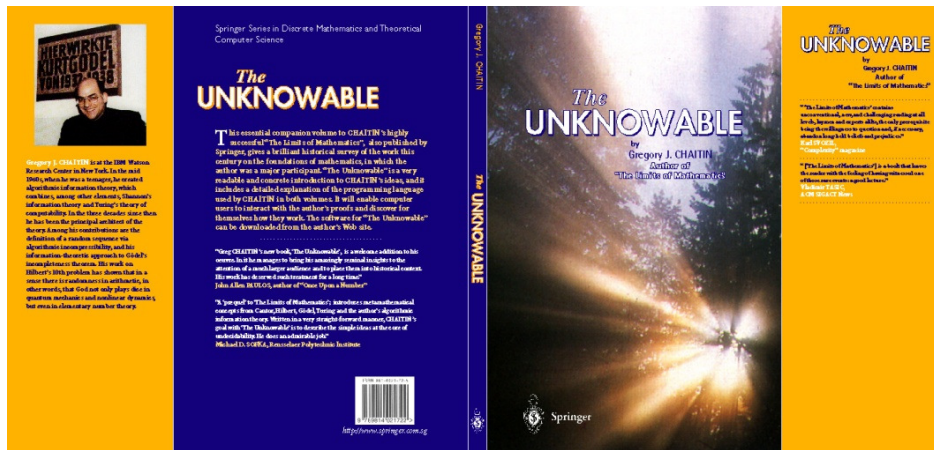
Kurt Gödel

For your pleasure (after R.Rucker)

- 1. Someone introduces Gödel to a UTM, a machine that is supposed to be a Universal Truth Machine, capable of correctly answering any question at all.
- 2. Gödel asks for the program and the circuit design of the UTM. The program may be complicated, but it can only be finitely long. Call the program P(UTM) for Program of the Universal Truth Machine.
- 3. Smiling a little, Gödel writes out the following sentence: "The machine constructed on the basis of the program P(UTM) will never say that this sentence is true." Call this sentence G for Gödel. Note that G is equivalent to: "UTM will never say G is true."
- 4. Now Gödel laughs his high laugh and asks UTM whether G is true or not.
- 5. If UTM says G is true, then "UTM will never say G is true" is false. If "UTM will never say G is true" is false, then G is false (since G = "UTM will never say G is true"). So if UTM says G is true, then G is in fact false, and UTM has made a false statement. So UTM will never say that G is true, since UTM makes only true statements.
- 6. We have established that UTM will never say G is true. So "UTM will never say G is true" is in fact a true statement. So G is true (since G = "UTM will never say G is true").
- 7. "I know a truth that UTM can never utter," Gödel says. "I know that G is true. UTM is not truly universal."

A False Assumption

Alan Turing und Gregory J. Chaitin



Über den Zufall in der (auch numerischen) Mathematik

- In a nutshell, Gödel discovered incompleteness, Turing discovered uncomputability, and I discovered randomness—that's the amazing fact that some mathematical statements are true for no reason, they're true by accident. There can be no "theory of everything" at least not in mathematics. Maybe in physics!*
 (G.J.Chaitin)

A False Assumption

Die Konsequenzen für die TesterInnen

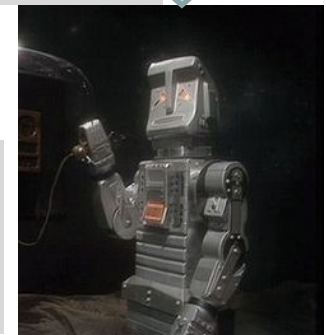


Sie kämpfen nicht
gegen

- Entwickler
- Fachbereich
- Chefs

Sie bekämpfen
Komplexität

„Marvin, the paranoid
android, its brains are the
size of the universe“



The consequences for the testers

In the meantime – A Tester's Mindset

(c) 2009 by OBJENTIS Software Integration GmbH

Seite 12

Die neue Herangehensweise

Ein gefundener Fehler bedeutet nicht eine menschliche Schwäche

- Kein Anlass zum Raunzen
- Kein Anlass zur Herumpitzelei
- Kein Anlass für Beschuldigungen

Ein gefundener Fehler bedeutet einen kleinen Sieg gegen das Unvermeidliche

- 500 Fehler, die in jeder Anwendung vorkommen können
- Ein paar Hundert Fehler, deren Quelle unbekannt ist
- Hunderte Herausforderungen für die TesterInnen

Eine veränderte Grundeinstellung

Die Herausforderung – mache es dem Anwender unmöglich, einen Fehler zu finden

- Alles andere ist weniger wichtig

Aber welche Fehler findet denn der Anwender?

- Die Testerin kann auf diese Frage antworten, wenn sie sich wie ein **Anwender** benimmt.
- Die Testerin weiß sich wie ein Anwender zu benehmen, wenn sie das **Geschäft** des Anwenders kennt
- Die Testerin kennt das Geschäft des Anwenders, wenn sie dessen **Ziele** kennt. Diese sind die Resultate von Geschäftsprozessen bzw. Workflows.
- Die Testerin kennt die Ziele des Geschäfts, wenn sie dessen Anwendungsfälle (**usecases**) kennt.

Die Grundeinstellung ändern – Ausrichten der Anstrengungen

Welche Fehler findet der Anwender?

- Die Testerin weiß die Antwort, wenn sie sich wie ein **Anwender** benimmt.
- Die Testerin kann sich wie ein Anwender benehmen, wenn sie das **Geschäft** des Anwenders kennt.
- Die Testerin kennt das Geschäft des Anwenders, wenn sie dessen **Ziele** kennt. Diese sind die Resultate von Geschäftsprozessen. (**business workflows**)
- Die Testerin kennt die Ziele, wenn sie die Anwendungsfälle (**usecases**) kennt.

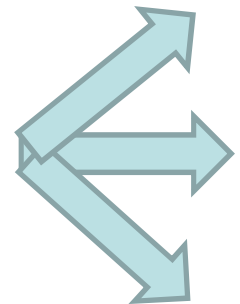


Unterschiedlicher Hintergrund

Unterschiedliche Ausbildung



Test Team



Rechtzeitig lehren!

Einheitliche Zielausrichtung

A new mindset

In the meantime – A Tester's Mindset

(c) 2009 by OBJENTIS Software Integration GmbH

Seite 15

Warum müssen Tester Usecases definieren?

- Müssen sie auch nicht
- Sollten sie nicht müssen
- **Aber sie müssen, wenn es noch keine gibt**

Genügen denn die Testfälle nicht?

- Testfälle müssen **spezielle Bedingungen** austesten
- Spezielle Bedingungen werden aus **Geschäftsregeln** abgeleitet
- Geschäftsregel sind Bestandteil von **Anwendungsfällen**
- **Wenn die Anwendungsfälle bekannt sind, sind die speziellen Bedingungen bekannt**
- **Wenn die Anwendungsfälle nicht bekannt sind, müssen die speziellen Bedingungen erst erraten werden.**

Reichen denn Testfälle nicht aus?

- Wenn es nur um eine beschränkte Anzahl von Testfällen geht, vielleicht ...
- Wenn als Ziel vermieden werden soll, dass der Anwender Fehler findet, so wird die reine Testfallmethodik sehr komplex und zeitaufwändig.
- Wenn allerdings der Anwendungsfall (usecase) bekannt ist, ist die Testfalldefinition ein Kinderspiel

Können denn die TesterInnen die notwendigen Usecases definieren?

- Wenn sich der Testmanager über die intellektuellen Anforderungen im Klaren ist → JA
- Sonst → NEIN

Ein hohes Maß an intellektueller Verantwortung

Fragen?



A high degree of intellectual responsibility