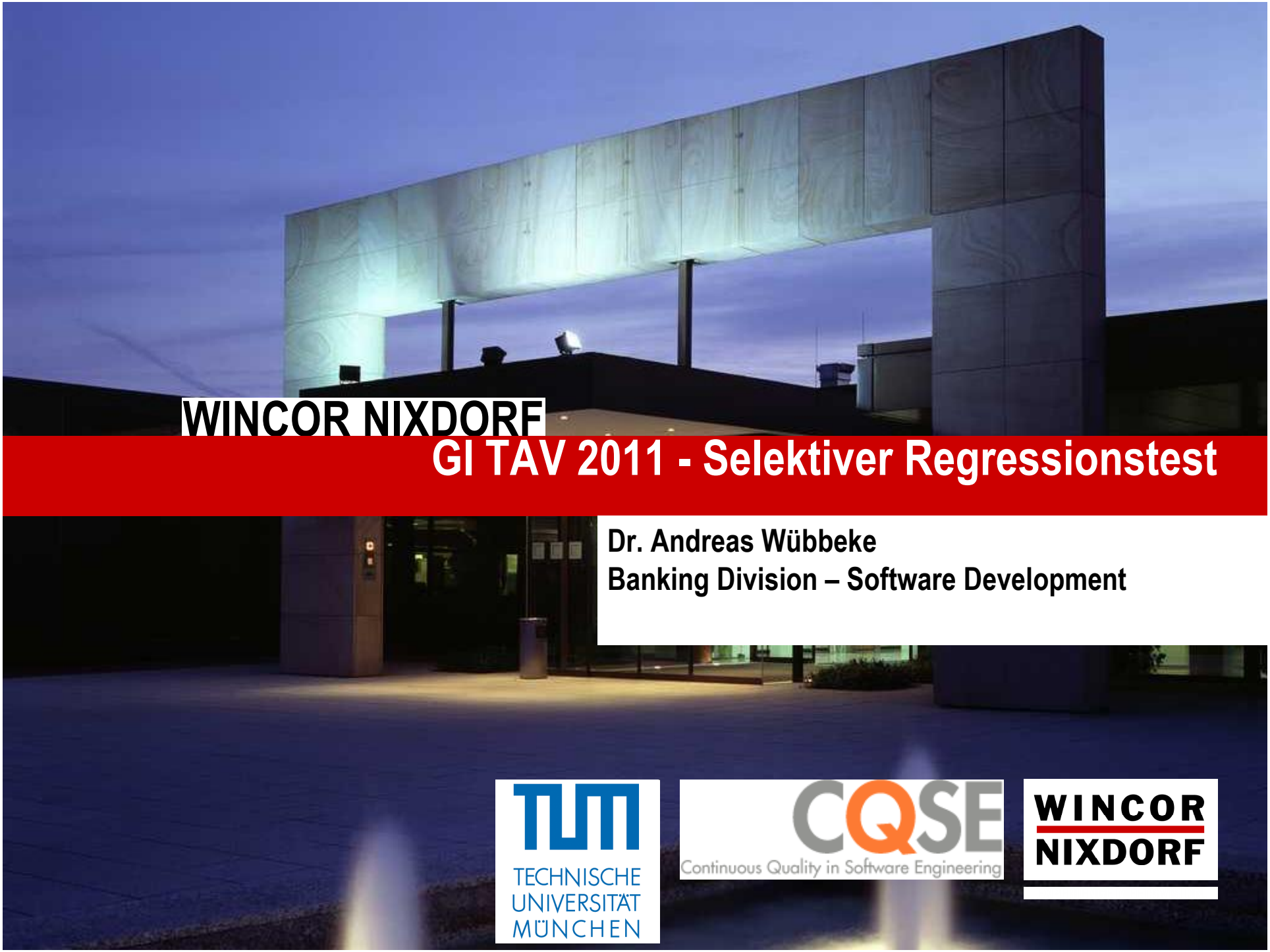


WINCOR NIXDORF

GI TAV 2011 - Selektiver Regressionstest

Dr. Andreas Wübbeke
Banking Division – Software Development

TECHNISCHE UNIVERSITÄT MÜNCHEN

CQSE Continuous Quality in Software Engineering

WINCOR NIXDORF

WINCOR NIXDORF

Agenda

I Problemstellung und Projektziele

II Stand der Forschung: Selektiver Regressionstest (SRT)

III Evaluierung: Möglichkeiten und Grenzen von SRT

IV Werkzeugunterstützung

V Zusammenfassung und Diskussion

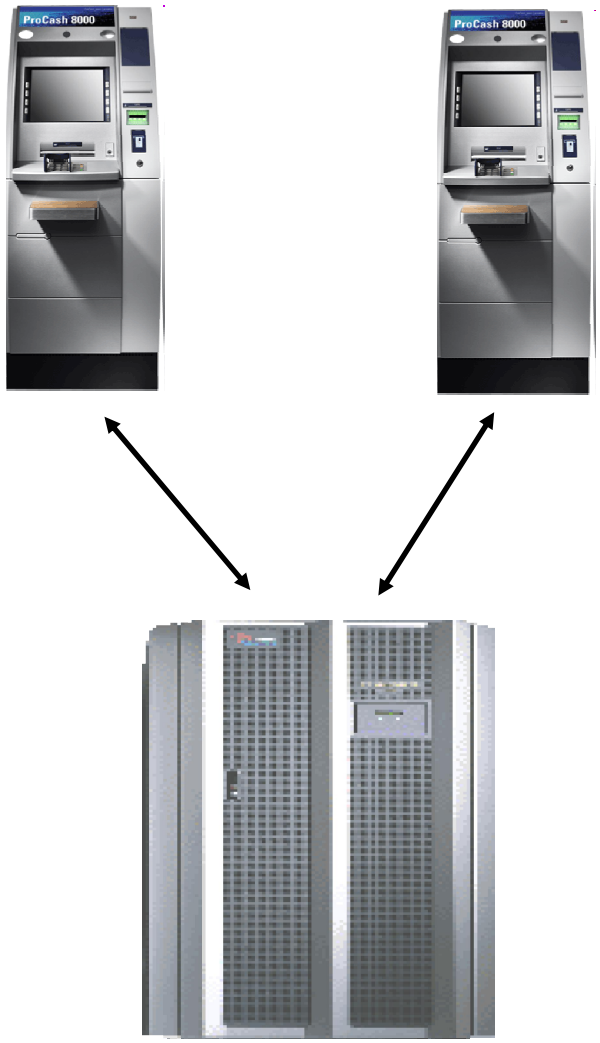
Kontext – Geldautomatennetzwerk als Prüfling

Geldautomaten

- Eingebettetes System
  - mit hoher Variantenvielfalt
  - in heterogenen Systemlandschaften
- Hohe Anforderungen an
  - Sicherheit
  - Robustheit
  - Einfachheit
  - …

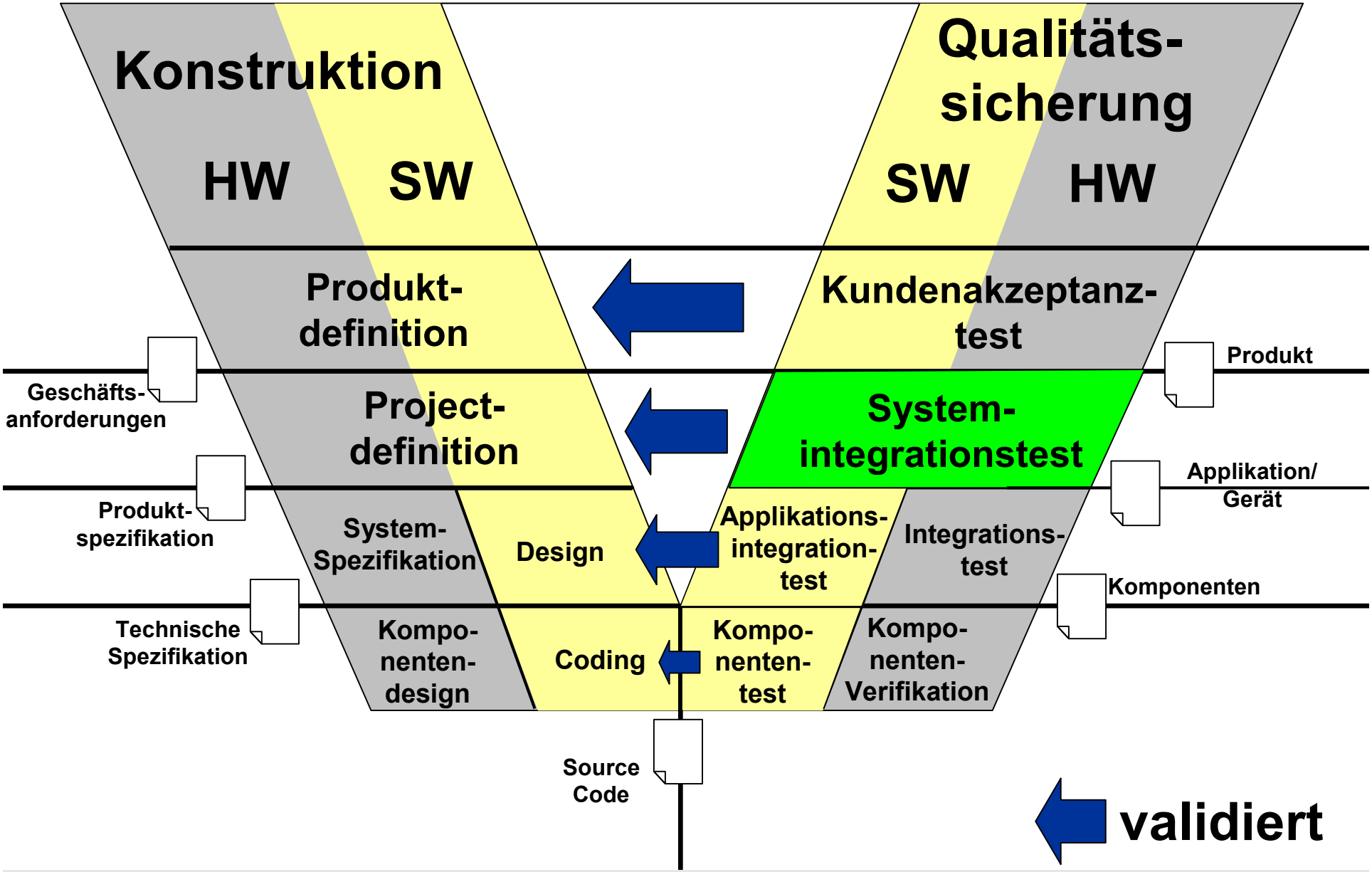
Transaktionsserver

- Informationssystem
  - Verarbeitung von Autorisierung, Transaktionen, Routing
  - Heterogene Systemlandschaften
- Hohe Anforderungen an
  - Sicherheit
  - Robustheit
  - Verfügbarkeit
  - Skalierbarkeit
  - …



WINCOR NIXDORF

Einordnung in den Entwicklungsprozess



WINCOR NIXDORF

Problemstellung

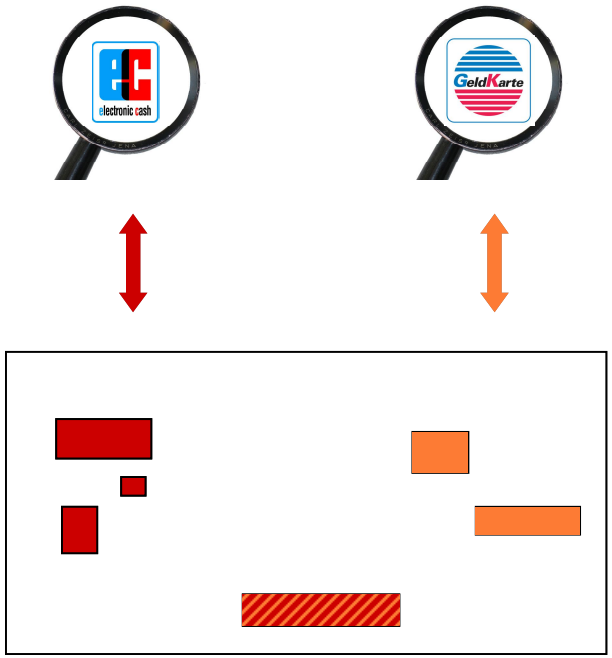
Systemintegrationstest

- Betrachtet das System als Black Box
- Sinnvoll zur Definition funktionaler Tests

Regressionstest

- Spezifikation ändert sich nicht
- Sichert gegenüber unerwünschten Nebeneffekten ab
- Abhängigkeiten im Code können zu Nebeneffekten auf fachlich scheinbar unabhängiger Funktionalität führen

→Black-Box-Sicht ist nicht ausreichend



Herausforderung

- Auswahl notwendig (Ausführung aller Tests nach Änderung zu teuer)
- Aber: Abhängigkeiten zwischen Code und Testfällen manuell nicht überschaubar

WINCOR NIXDORF

Projektziele

Werkzeugunterstütze Auswahl von Regressionstests

- Evaluierung der Möglichkeiten und Grenzen von Werkzeugunterstützung
- Konzeption und prototypische Realisierung
- Evaluierung bei Wincor Nixdorf

WINCOR NIXDORF

Agenda

I Problemstellung und Projektziele

II Stand der Forschung: Selektiver Regressionstest (SRT)

III Evaluierung: Möglichkeiten und Grenzen von SRT

IV Werkzeugunterstützung

V Zusammenfassung und Diskussion

WINCOR NIXDORF

Verfahren in der Literatur

Fragestellung: Was mache ich, wenn ich nicht alle Testfälle ausführen kann?

Selektiver Regressionstest (SRT)

- Wählt Teilmenge der Testfälle bzgl. durchgeführter Modifikationen aus
- ✓ Bietet Einsparungspotential

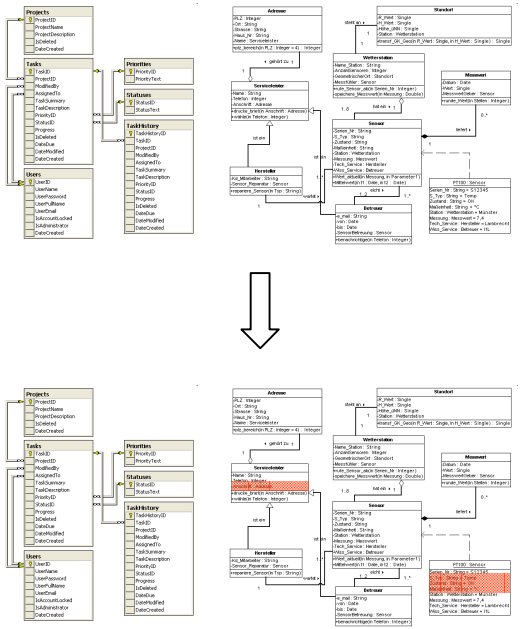
Testfall-Priorisierung

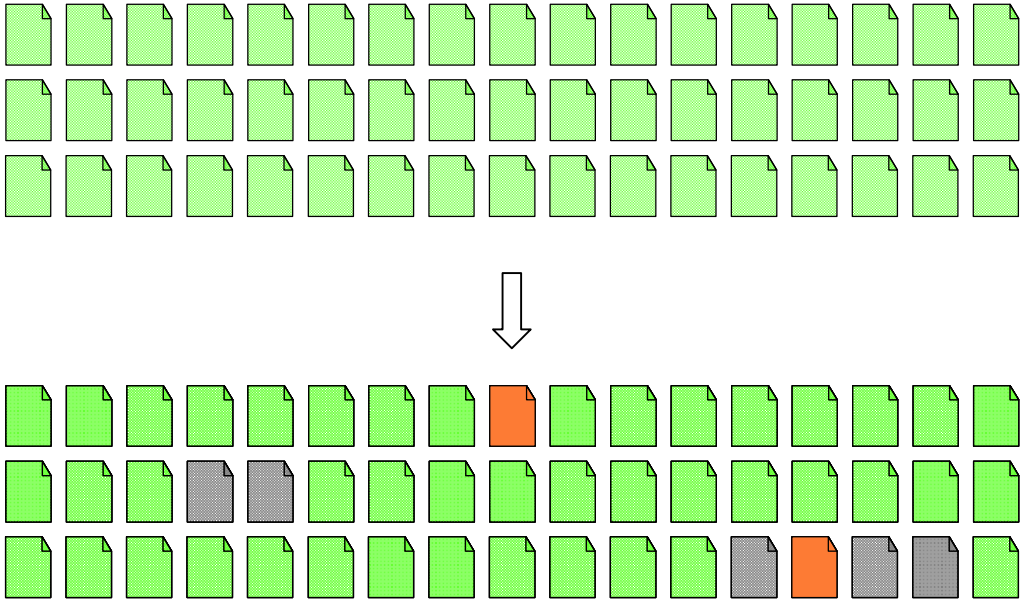
- Ordnet Testfälle nach Wahrscheinlichkeit Fehler aufzudecken
- \- Unabhängig von Modifikationen, daher kein Einsparungspotential

Testfall-Reduktion

- Identifiziert redundante Testfälle zur Reduktion der Gesamttestmenge
- \- Unabhängig von Modifikationen, kein kurzfristiges Einsparungspotential

SRT: Überblick





Ziel

- Ermittlung aller fehleraufdeckenden Testfälle (in Bezug auf Modifikationen)

Ansatz

- Identifikation der modifikations-traversierenden Testfälle

SRT: Ablauf

Mapping Testfall ←→ Code

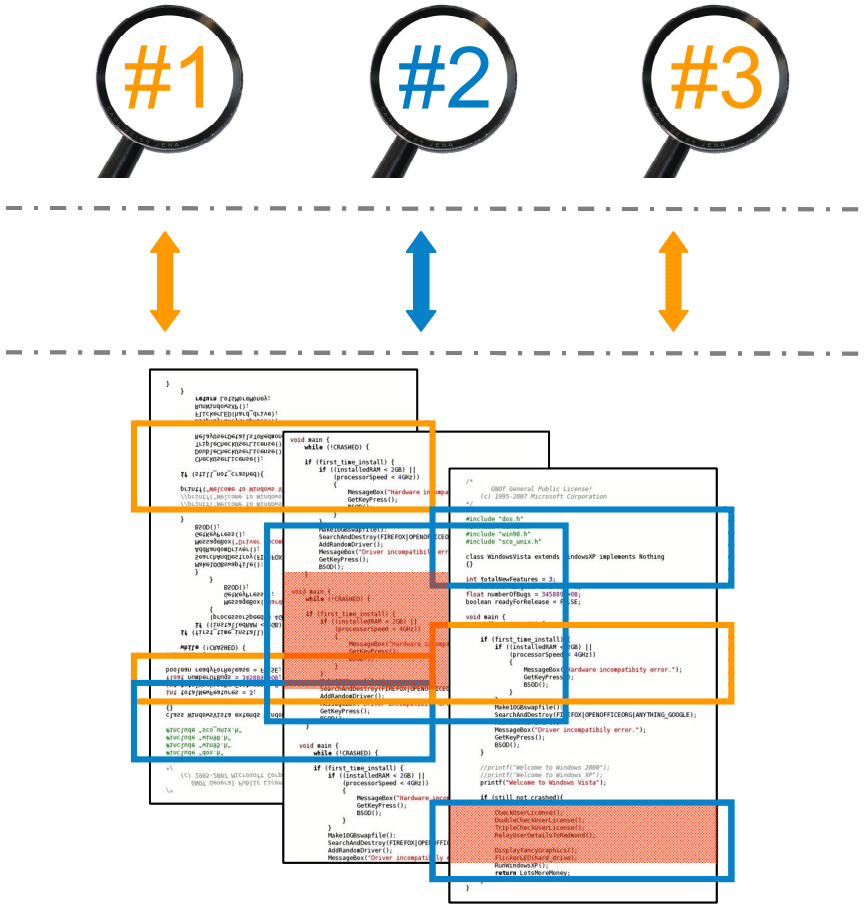
- Tracing während Test
- Granularität variiert

Ermitteln der Modifikationen

- Differenz bzgl. letzter Version, auf der Test durchgeführt wurde
- Nutzung von Source Code / Binary

Identifikation modifikations-traversierender Testfälle

- Traces und Diffs
- Unterschiedliche Algorithmen



WINCOR NIXDORF

SRT: Klassifikation Verfahren

Sichere SRT Verfahren

Fehleraufdeckende Testfälle garantiert in Testmenge
Konservativ (kann zu Überabschätzung führen)
Anwendbarkeit in der Praxis unklar

Heuristische Verfahren

Keine Garantie dass fehleraufdeckende Tests ausgewählt werden
Ermöglichen oft größere Einsparungen (weniger Überabschätzung)
Deutlich bessere Anwendbarkeit

WINCOR NIXDORF

SRT: Annahmen

Entwicklungsprozess

Tests vor Modifikation erfolgreich ausgeführt
Umfang der Modifikationen relativ gering

Kontrolle

Außer der Modifikationen wird an der Testumgebung nichts verändert

Abhängigkeiten

Alle Abhängigkeiten zwischen Software-Artefakten sind dem SRT Werkzeug bekannt

(Inklusive: DB Zugriff, Caches, Konfigurationsparameter, …)

Müssen für sichere SRT Verfahren vollständig erfüllt sein.
Nicht in vollem Umfang für heuristische Verfahren notwendig.

WINCOR NIXDORF

Agenda

I Problemstellung und Projektziele

II Stand der Forschung: Selektiver Regressionstest (SRT)

III Evaluierung: Möglichkeiten und Grenzen von SRT

IV Werkzeugunterstützung

V Zusammenfassung und Diskussion

WINCOR NIXDORF

Evaluierung der Grenzen und Möglichkeiten

Zentrale Entscheidungen für Werkzeugprototypen

Sicheres oder heuristisches Verfahren?
Vollautomatischer Ansatz, oder Expertenunterstützung?

Forschungsfragen

Wie groß ist die Menge an Testfällen, die SRT auswählt (bzgl. aller Testfälle)

Wie wirkt sich Unterspezifikation der Testfälle aus?

Eignen sich Traces zur Vorhersage welche Testfälle modifizierten Code ausführen werden?

WINCOR NIXDORF

Größe der selektierten Testfallmenge

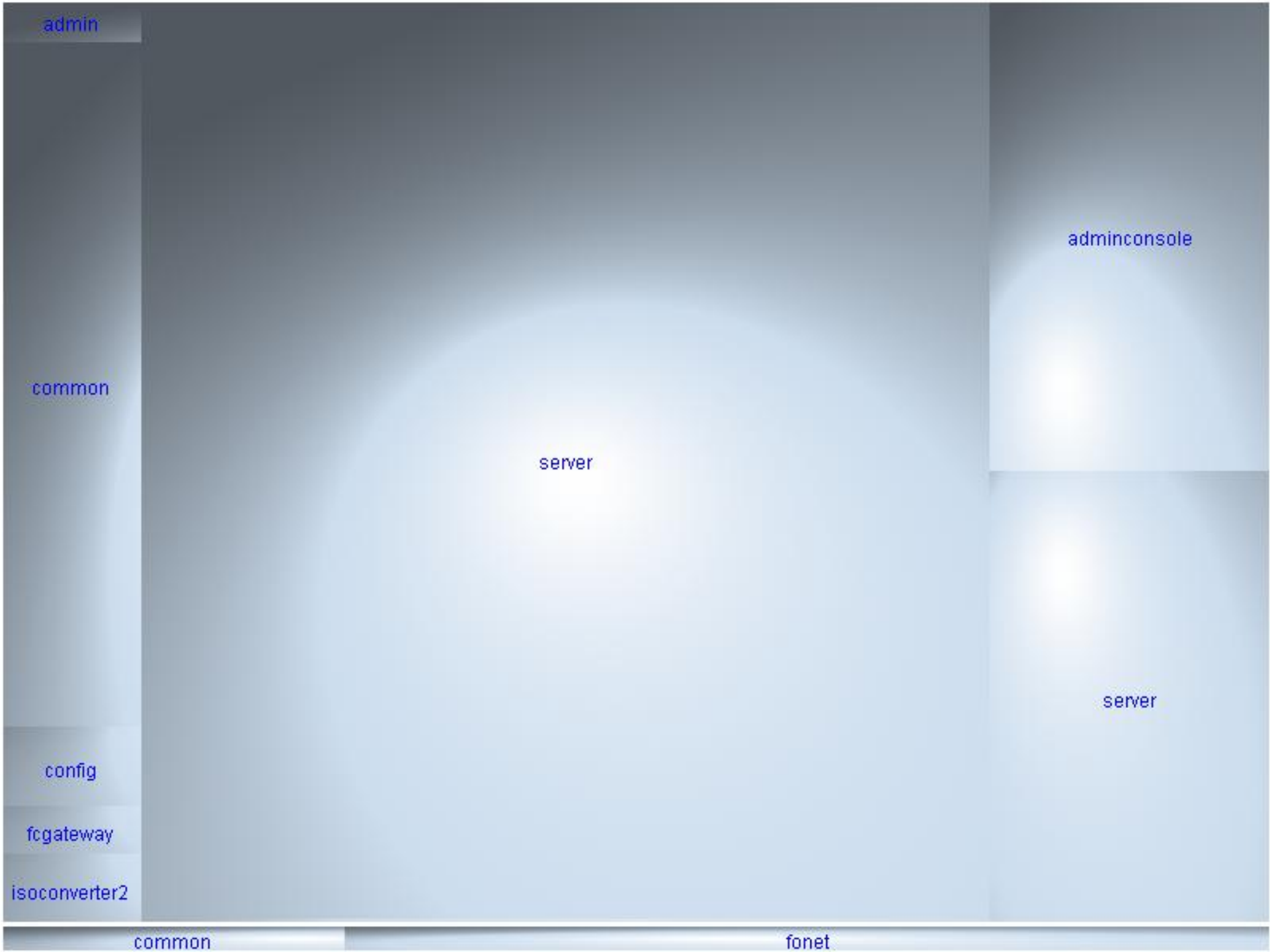
RQ: **Wie groß ist die Menge an Testfällen, die SRT auswählt (bzgl. aller Testfälle)**

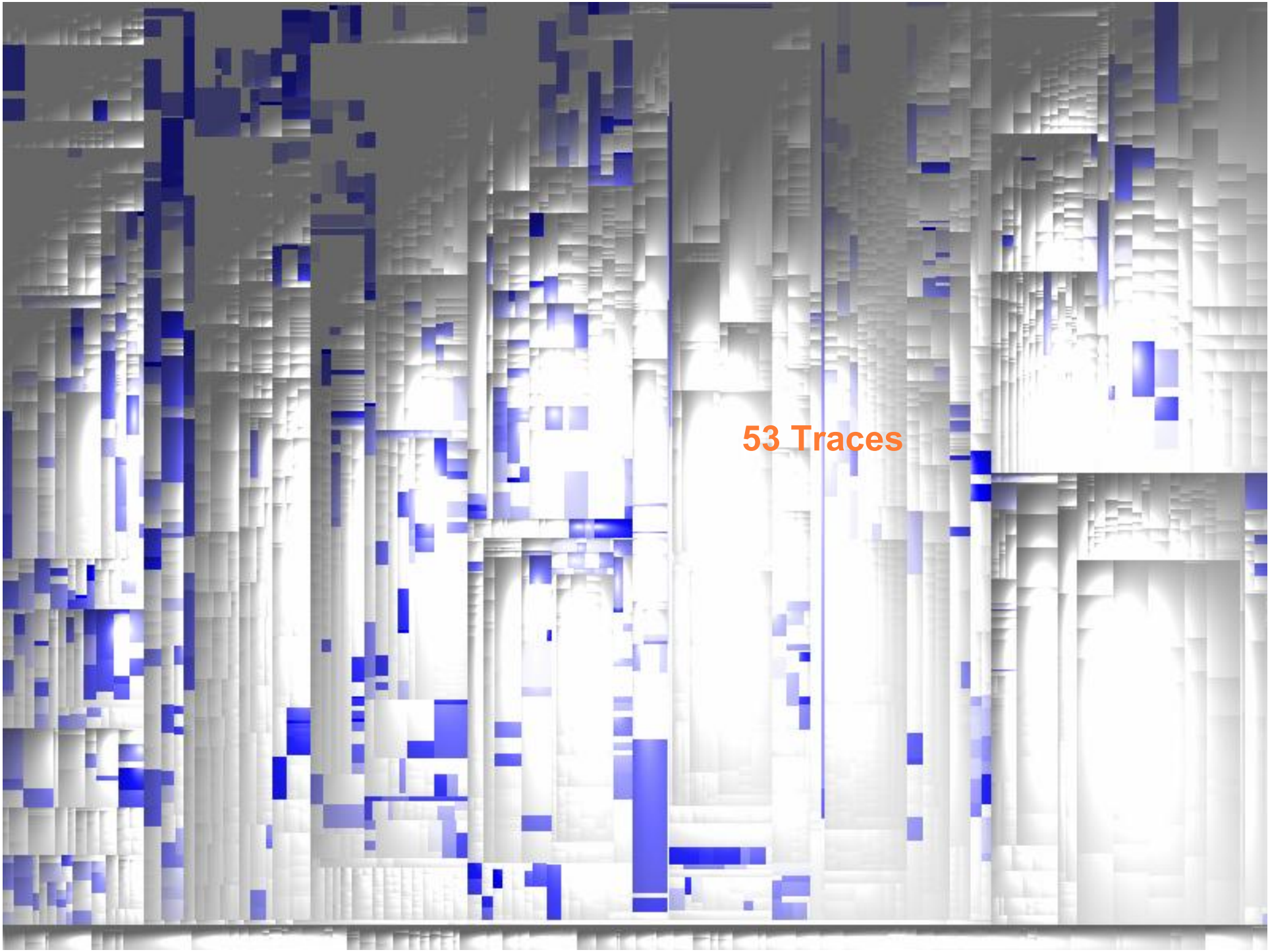
Vorgehen

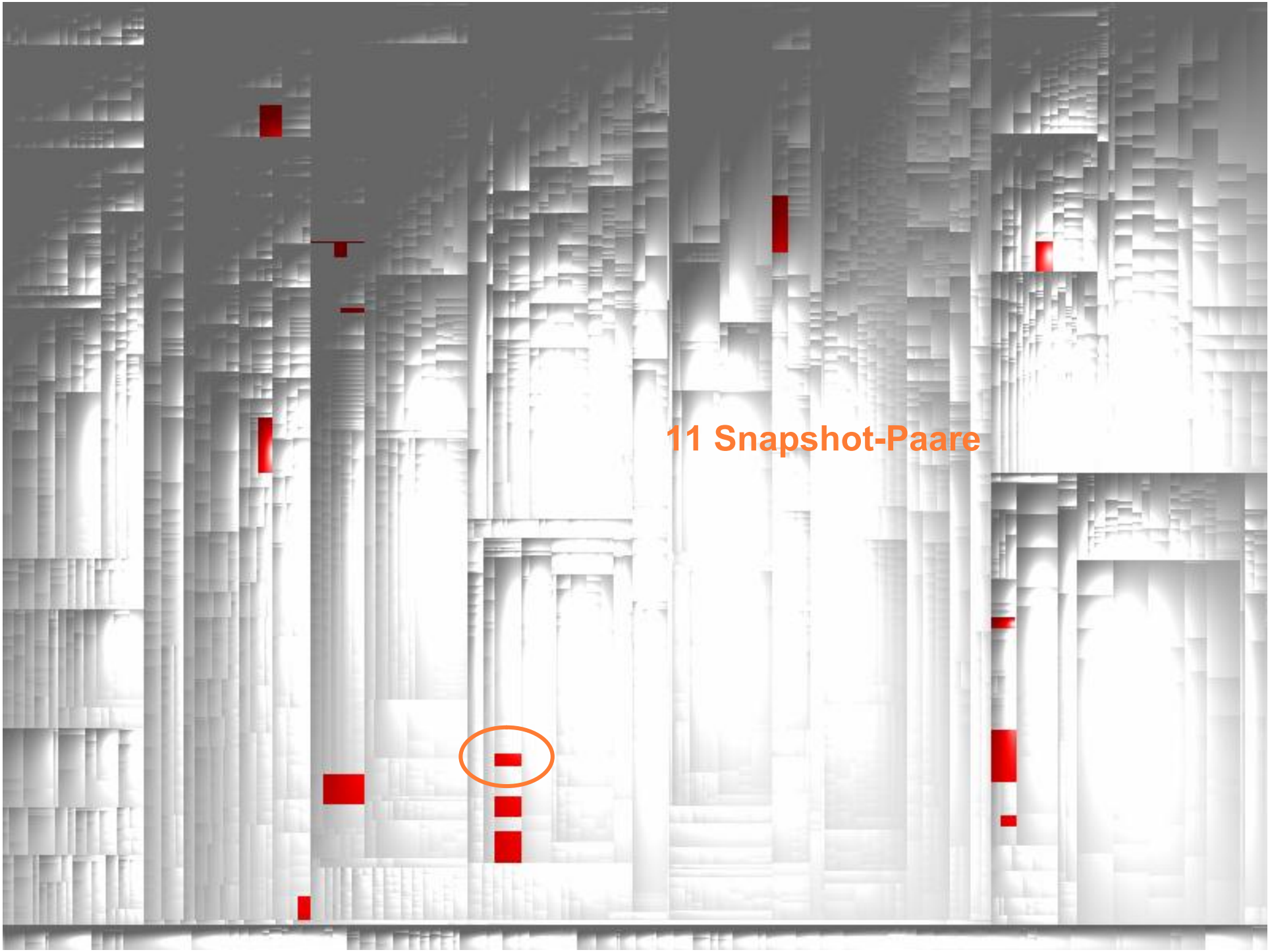
12 Snapshots aus Korrektur-Branch (4157_05 - 4157_16): S_1 bis S_{12}
53 Testfälle auf S_1 ausgeführt und Traces aufgezeichnet

Bestimmung der Code-Modifikationen zwischen S_1/S_2, S_2/S_3, S_3/S_4, …
Ermitteln der Anzahl betroffener Testfälle

Ermitteln der Anzahl betroffener Testfälle für zufällig ausgewählte Methoden


admin
adminconsole
common
server
server
config
fcgateway
isoconverter2
common
fonet


53 Traces


11 Snapshot-Paare

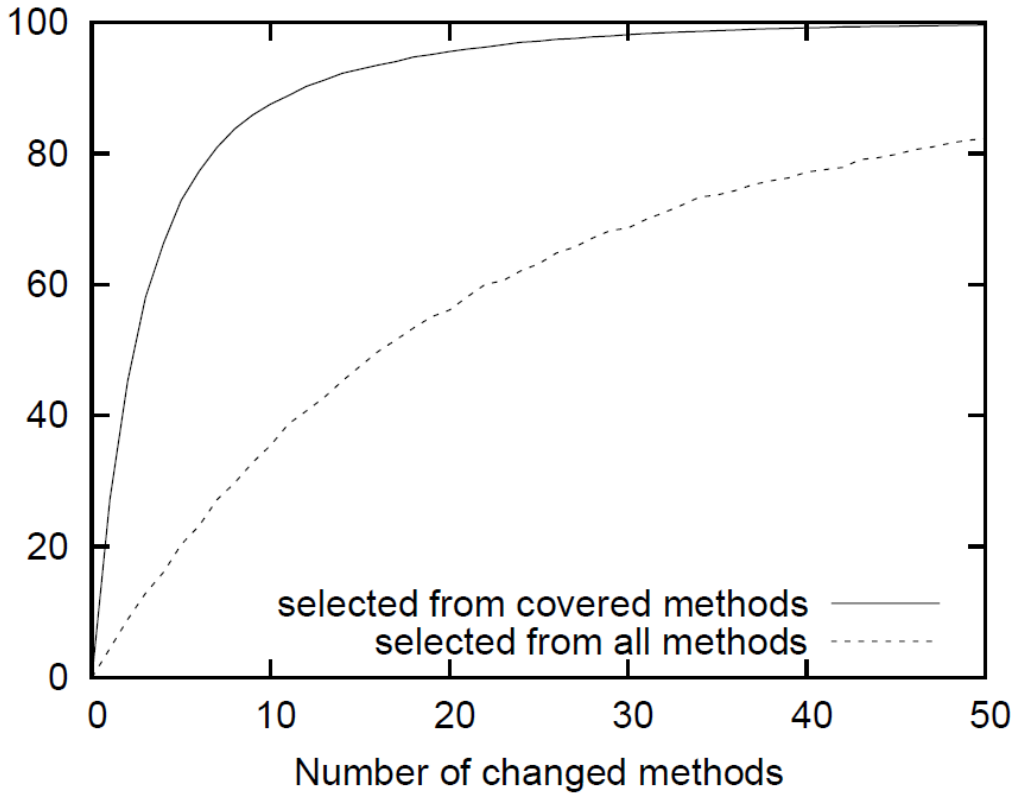
WINCOR NIXDORF

Ergebnis: Größe der selektierten Testfallmenge

Bei S_6, S_8 und S_9 Änderungen am Code
41/12/30 Methoden geändert,
13/4/2 überdeckt,
62%/62%/6% Tests betroffen

Allgemeine Änderungen

Zufällige Auswahl von Methoden
Bei 3 geänderten Methoden 50% der Tests betroffen



WINCOR NIXDORF

Auswirkung von Testfall-Unterspezifikation

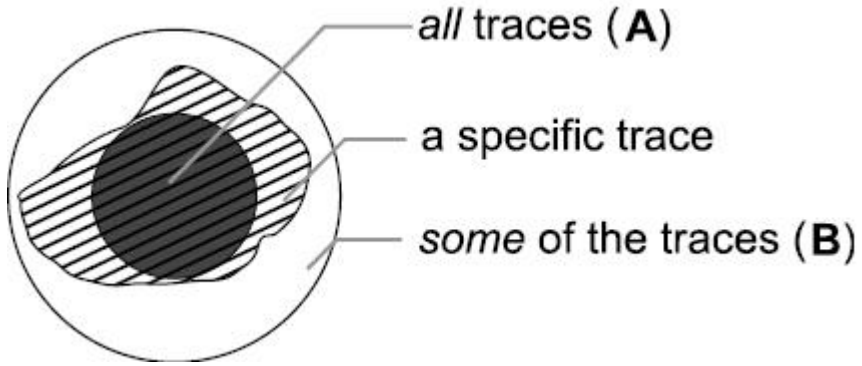
RQ: **Wie wirkt sich Unterspezifikation der Testfälle aus?**

Vorgehen

T_1, T_2, T_3 je 6x auf S_1 getraced
Jede Ausführung konsistent zu Spec.
Wieviele Methoden werden immer / manchmal durchlaufen?

Ergebnis

Im schlechtesten Fall: nur 1/3 der Methoden immer überdeckt
Sichere Verfahren nicht einsetzbar



test case	$\|A\|$	$\|B\|$	$\|A\|/(\|A\|+\|B\|)$
T_1	3692	325	92%
T_2	4211	314	93%
T_3	1605	3295	33%

WINCOR NIXDORF

Interpretation der Ergebnisse

Größe der Selektierten Testfallmenge

Schon bei 3 unabhängigen Änderungen >50% der Tests betroffen
Deutlich zu viele für Einsatz während Hotfix
Kein vollautomatischer Ansatz für Werkzeug

Unterspezifikation von Testfällen

Kontroll-Annahme nicht erfüllt.
Ursache: Blackbox / Whitebox Diskrepanz
Vollständige Spezifikation in der Praxis nicht machbar
Kein sicheres Verfahren

Ausrichtung Werkzeug:

Fokus auf Expertenunterstützung
Einsatz von heuristischen SRT Verfahren

WINCOR NIXDORF

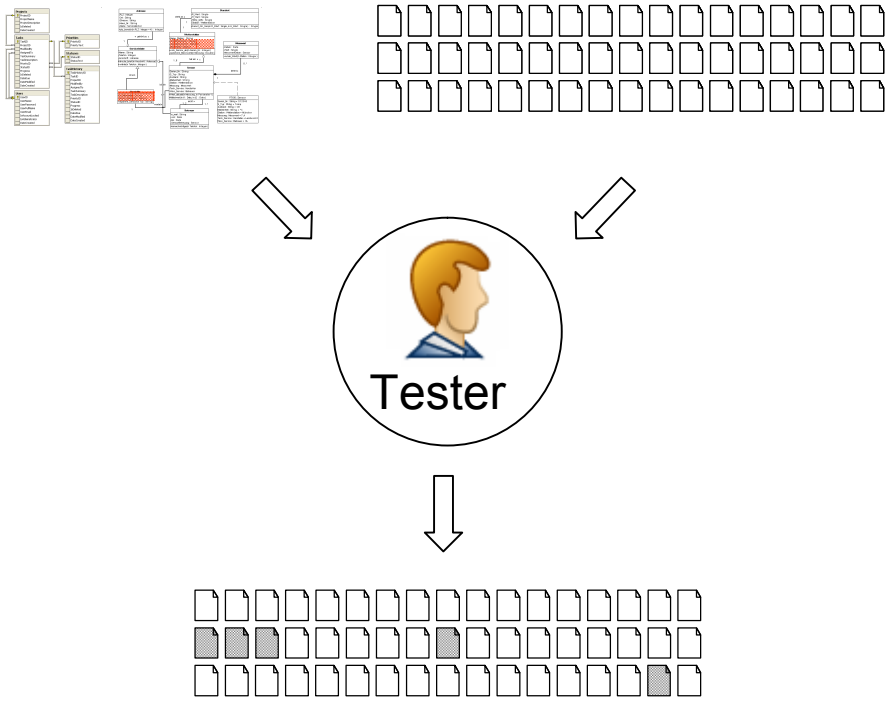
Trace-Guided Testfall Selektion

Ermittlung Testfall Kandidaten

Berechnung des Deltas
Ermittlung der Tests, die mit hoher Wahrscheinlichkeit Änderungen durchlaufen

Priorisierung von Kandidaten

Tester inspiziert Delta im Code
Ggf. Rückfragen an Entwickler
Wählt Testfälle aus Kandidaten



Ziel: Verbindung von SRT Werkzeug und Expertenwissen

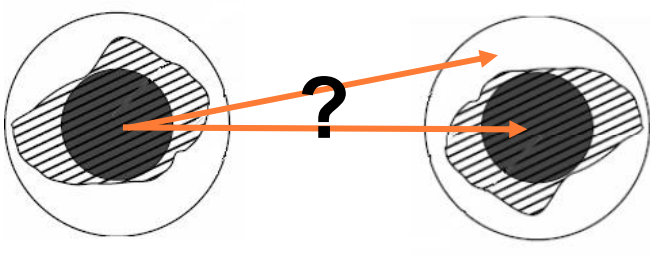
WINCOR NIXDORF

Trace Stabilität

RQ: **Wie stabil ist die Methodenüberdeckung über Systemversionen hinweg?**

Vorgehen

T_1, T_2, T_3 je 3x auf S_5 und S_{12} getraced
Jede Ausführung konsistent zu Spec.
Sind stabile Ms aus S_1 weiterhin stabil?



Ergebnis

Überdeckung stabil
Geeignet für Testfall Selektion

	$S_1 \rightarrow S_5$	$S_1 \rightarrow S_{12}$
T_1	99.1%	99.0%
T_2	99.6%	98.5%
T_3	100%	99.9%

WINCOR NIXDORF

Agenda

I Problemstellung und Projektziele

II Stand der Forschung: Selektiver Regressionstest (SRT)

III Evaluierung: Möglichkeiten und Grenzen von SRT

IV Werkzeugunterstützung

V Zusammenfassung und Diskussion

WINCOR NIXDORF

Aufzeichnung der Traces: Anforderungen

Programm muss nicht manuell modifiziert werden

Laufzeiten erhöhen sich nicht zu stark

Programmverhalten darf sich nicht ändern

Muss auch in Test-Umgebung funktionieren

WINCOR NIXDORF

Aufzeichnung der Traces: Mögliche Lösungen

Automatische und systematische Modifikation des Programms

- Instrumentierung zur Übersetzungszeit (im Quellcode)
- Byte-Code-Instrumentierung (nach Übersetzung)
- Instrumentierung zur Laufzeit

Aufzeichnung über eine kontrollierte Ausführungsumgebung (Simulation)

- Wird z.B. von Valgrind genutzt

WINCOR NIXDORF

Aufzeichnung der Traces: Eingesetzte Technik

Instrumentierung des Java-Byte-Codes zur Laufzeit

Instrumentierungsschnittstelle: Java Agents

- Unterstützt ab Java 1.5

Vorteile

- Keine Änderungen an Source- und Byte-Code
- Funktioniert auch mit Byte-Code der erst zur Laufzeit erzeugt wird (z.B. AOP)
- Robust in komplexen Umgebungen (mehrere Threads, mehrere Class Loader)
- Aktivierung/Deaktivierung durch „einfachen“ Neustart des Programms

WINCOR NIXDORF

Aufzeichnung der Traces: Evaluierung (1)

System	#Tests	Zeit ohne Tracing (Sek.)	Zeit mit Tracing (Sek.)	Slow Down
CCSM Commons	**417**	**7,7**	**13,2**	**71%**
ConQAT Driver	**108**	**1,1**	**1,2**	**9%**
ConQAT CloneDetective	**95**	**2,0**	**2,9**	**45%**
ConQAT System Tests	**27**	**260**	**267**	**<3%**

Aufzeichnung der Traces: Evaluierung (2)

Wincor-Unit-Tests liefern gleiches Ergebnis mit und ohne Tracer

Zeitmessung der Startphase der PC/E-Prozesses (inkl. WebSphere-Overhead)

System	Normal	Tracer ohne Aufzeichnung	Tracer mit Aufzeichnung
real	54 - 82	57 - 71	56 – 57
user	3,3 - 3,9	3,3 – 4,0	3,0 – 3,7
system	2,2 – 2,5	2,1 – 2,6	2,2 – 2,8

WINCOR NIXDORF

Ermittlung der Systemänderungen: Anforderungen

Änderungen am System müssen „robust“ erkannt werden

- Tolerant gegenüber Änderungen der Formatierung/Einrückung
- Tolerant gegenüber Änderungen außerhalb des ausgeführten Codes
  - Kommentare
  - Import-Statements

➔Einfacher Diff reicht nicht

WINCOR NIXDORF

Ermittlung der Systemänderungen: AST-Basiert

Nutzung eines Parsers um Struktur des Java-Codes zu erkennen
Bestimmung der Unterschiede auf dem AST

Kann mit Formatierung und Kommentaren umgehen
Problematisch: Import-Statements, etc.

Guter Parser für Java schwer zu erstellen/zu finden
Benötigt Typ-Resolution um zuverlässig zu funktionieren

WINCOR NIXDORF

Ermittlung der Systemänderungen: Byte-Code

Alternativer Ansatz: Erkennen der Änderungen im Byte-Code

Vorteil: Compiler überführt Code in Normalform

- Tolerant gegenüber Formatierung und Kommentaren
- Import-Statements sind normalisiert
- Tolerant gegenüber Umbenennung von lokalen Variablen und Parametern
- Genau der kompilierte Code wird betrachtet (unabhängig von MKS)

Änderungen werden nur für Methoden betrachtet

- Änderungen an Feldern wirken sich auf (compilierte) Methoden aus die diese Nutzen
- Konstanten werden bei Java inline genutzt (oder in *static initializer* belegt)

➔Wichtig: Version und Einstellungen des Compilers können Auswirkungen haben

WINCOR NIXDORF

Agenda

I Problemstellung und Projektziele

II Stand der Forschung: Selektiver Regressionstest (SRT)

III Evaluierung: Möglichkeiten und Grenzen von SRT

IV Werkzeugunterstützung

V Zusammenfassung und Diskussion

WINCOR NIXDORF

Fazit: Werkzeugunterstützung SRT

Ausrichtung

Unterstützung von Testern bei Auswahl von Regressionstests
(Kein Ersetzen der Tester durch Tool)
Heuristisches Verfahren, da sichere Verfahren nicht anwendbar für Systemtests
Liefert keine obere Schranke für Testdurchführung

Stand Prototyp

Tracer evaluiert auf PC/E Testumgebung
Differ unabhängig von Source Konfiguration, erfolgreich eingesetzt
Vollständiger Funktionsumfang prototypisch realisiert

Literatur

Elmar Juergens, Benjamin Hummel, Florian Deissenboeck, Martin Feilkas, Christian Schlögel, Andreas Wübbeke:
Regression Test Selection of Manual System Tests in Practice
To appear in Proc. of 15th European Conference on Software Maintenance and Reengineering (CSMR'11), 2011.

Ausblick: Mögliche Prozess-Integration

Aufzeichnen der Traces

Integration des SRT Werkzeugs in das WN Testmanagementwerkzeug zur Trace Erzeugung?
Ziel: Minimierung der notwendigen Arbeitsschritte für Tester

Aufzeichnen der Snapshots

Import in SRT Tool, wenn neuer Snapshot zum ersten Mal getestet wird
Integration in Prozess zur Installation eines neuen Snapshots im Test?

Anwendung des Werkzeugs

Auswahl von Regressionstestfällen für Hotfixes
Überprüfung der Testabdeckung von neuem Code
Überprüfung der Testabdeckung von Änderungen
(Fokus nicht auf *vollständiger*, sondern auf *bewusster* Überdeckung)

WINCOR NIXDORF

Diskussion