Analyse und Konzept zur Verbesserung der statischen Fehlereingrenzung

Masterarbeit von Lars Gohlke

- Ziel der Arbeit
- Motivation
- State of the Art
- Methodik
- Zusammenfassung/Ausblick

leichtgewichtigen heuristischen Gegenentwurf zu aufwendigen statistischen Verfahren

Quelle: http://cache.gawkerassets.com/assets/images/4/2009/08/districtcarbon_black.jpg 2012.04.26

- Ziel der Arbeit
- **Motivation**
- State of the Art
- Methodik
- Zusammenfassung/Ausblick

Fehlereingrenzung

- engl.: bug isolation
- Fehlersuche
- sehr zeitintensiv

- **Szenario 1**

Langlaufende Integrationstests

z.B. 5h/Iteration

Quelle: http://www.elektronikpraxis.vogel.de/imgserver/bdb/225300/225322/original.jpg 2012.04.26

FACHHOCHSCHULE BRANDENBURG

Testergebnisse

Fehlschläge (+33)

Alle fehlgeschlagenen Tests

Testname
>>> org.apache.camel.management.JmxInstrumentationWithConnectorTest.testCounters
>>> org.apache.camel.management.ManagedCamelContextTracerTest.testCamelContextTracing
>>> org.apache.camel.management.ManagedCamelContextUpdateRoutesFromXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedCustomPolicyTest.testPolicy
>>> org.apache.camel.management.ManagedErrorHandlerOptionsTest.testManagedErrorHandlerOptions
>>> org.apache.camel.management.ManagedErrorHandlerRedeliveryTest.testManagedErrorHandlerRedelivery
>>> org.apache.camel.management.ManagedErrorHandlerTest.testManagedErrorHandler
>>> org.apache.camel.management.ManagedRegisterRouteTest.testRoutes
>>> org.apache.camel.management.ManagedRegisterTwoRoutesTest.testRoutes
>>> org.apache.camel.management.ManagedRouteDumpRouteAsXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedRouteNoAutoStartupTest.testRouteNoAutoStartup
>>> org.apache.camel.management.ManagedRoutePolicyTest.testRoutes
>>> org.apache.camel.management.ManagedRouteRemoveTest.testRemove
>>> org.apache.camel.management.ManagedRouteShutdownAndStartTest.testShutdownAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartCleanupTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopTest.testStopRoute
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutTrue
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutFalse

Quelle: https://builds.apache.org/job/Cassandra/1271/testReport/ 2012.04.26

- **Fragestellungen:**
  - Wie viele verschiedene Ursachen gab es?
  - Sind das vielleicht nur Folgefehler?

- **Szenario 2**

Neuer Mitarbeiter im Projekt

FACHHOCHSCHULE BRANDENBURG

Testergebnisse

Fehlschläge (+33)

Alle fehlgeschlagenen Tests

Testname
>>> org.apache.camel.management.JmxInstrumentationWithConnectorTest.testCounters
>>> org.apache.camel.management.ManagedCamelContextTracerTest.testCamelContextTracing
>>> org.apache.camel.management.ManagedCamelContextUpdateRoutesFromXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedCustomPolicyTest.testPolicy
>>> org.apache.camel.management.ManagedErrorHandlerOptionsTest.testManagedErrorHandlerOptions
>>> org.apache.camel.management.ManagedErrorHandlerRedeliveryTest.testManagedErrorHandlerRedelivery
>>> org.apache.camel.management.ManagedErrorHandlerTest.testManagedErrorHandler
>>> org.apache.camel.management.ManagedRegisterRouteTest.testRoutes
>>> org.apache.camel.management.ManagedRegisterTwoRoutesTest.testRoutes
>>> org.apache.camel.management.ManagedRouteDumpRouteAsXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedRouteNoAutoStartupTest.testRouteNoAutoStartup
>>> org.apache.camel.management.ManagedRoutePolicyTest.testRoutes
>>> org.apache.camel.management.ManagedRouteRemoveTest.testRemove
>>> org.apache.camel.management.ManagedRouteShutdownAndStartTest.testShutdownAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartCleanupTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopTest.testStopRoute
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutTrue
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutFalse

- **Fragenstellungen:**
  - Wo soll man beginnen?
  - Wie kann man zielgerichtet vorgehen?

Wie kann man effizient Fehler eingrenzen?

Quelle: http://customersrock.net/wp-content/uploads/2010/01/question-mark.jpg 2012.04.26

- Ziel der Arbeit
- Motivation
- **State of the Art**
- Methodik
- Zusammenfassung/Ausblick

Aktuelle Verfahrensweisen

- Konstruktive Qualitätssicherung
- Statistische Fehlerisolierung

Aktuelle Verfahrensweisen

- **Konstruktive Qualitätssicherung**
- Statistische Fehlerisolierung

Konstruktive Qualitätssicherung (statisch)

Fehler im Ansatz vermeiden

Quelle: http://icons.iconarchive.com/icons/deleket/sleek-xp-basic/256/Lamp-icon.png 2012.04.26

Bestandteile:

- Software-Richtlinien
- Typisierung
- Vertragsbasierte Programmierung
- Fehlertolerante Programmierung
- Portabilität
- Dokumentation

Aktuelle Verfahrensweisen

- Konstruktive Qualitätssicherung
- **Statistische Fehlerisolierung**

FACHHOCHSCHULE BRANDENBURG

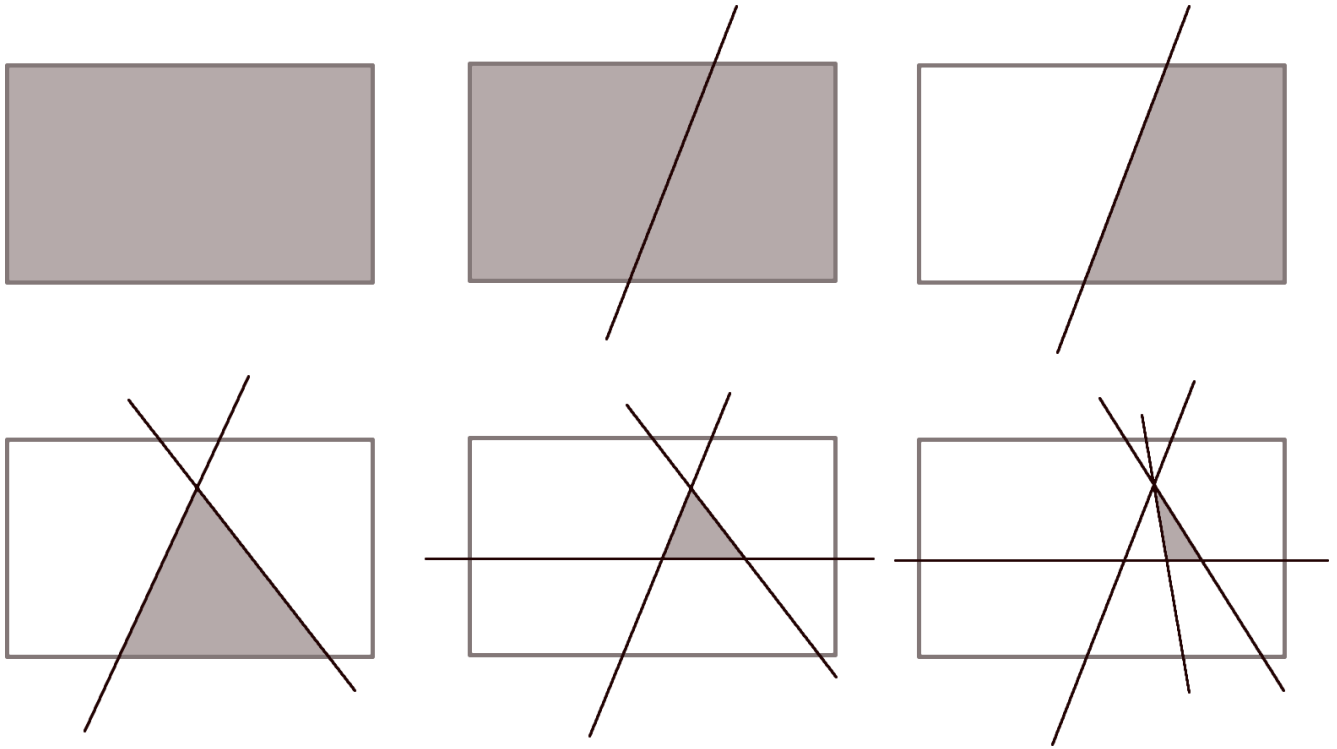
Statistische Fehlerisolation (dynamisch)

Quelle: http://www.theresilientearth.com/files/images/ibm_supercomputer.jpg 2012.04.26

- **delta debugging**
- **cooperative bug isolation**

- **delta debugging**

  - Analyse der Änderungshistorie



Quelle: http://www.st.cs.uni-saarland.de/dd/narrowing.gif 2012.04.26

- **cooperative bug isolation**
  - Laufzeitanalyse mit adaptivem Sampling

FACHHOCHSCHULE BRANDENBURG

- **Problem**

  - viele Durchläufe → viel Zeit

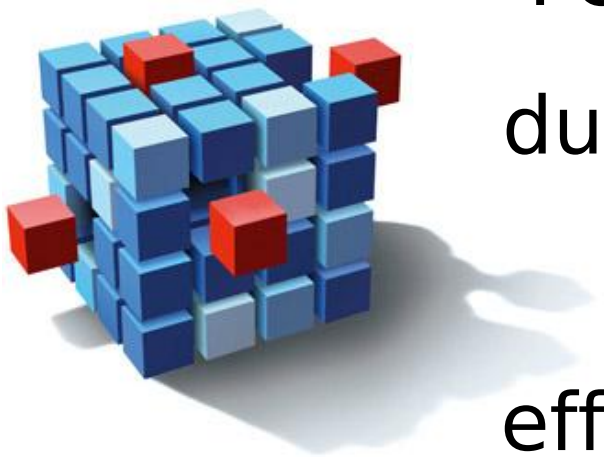
  - unbrauchbar für Integrationstests

- Ziel der Arbeit
- Motivation
- State of the Art
- **Methodik**
- Zusammenfassung/Ausblick

- **Idee**

Verbindung des statischen und dynamischen Ansatzes mit Hilfe einer Softwaremetrik

- **Ziel**



Fehlereingrenzung

durch Vorsortierung

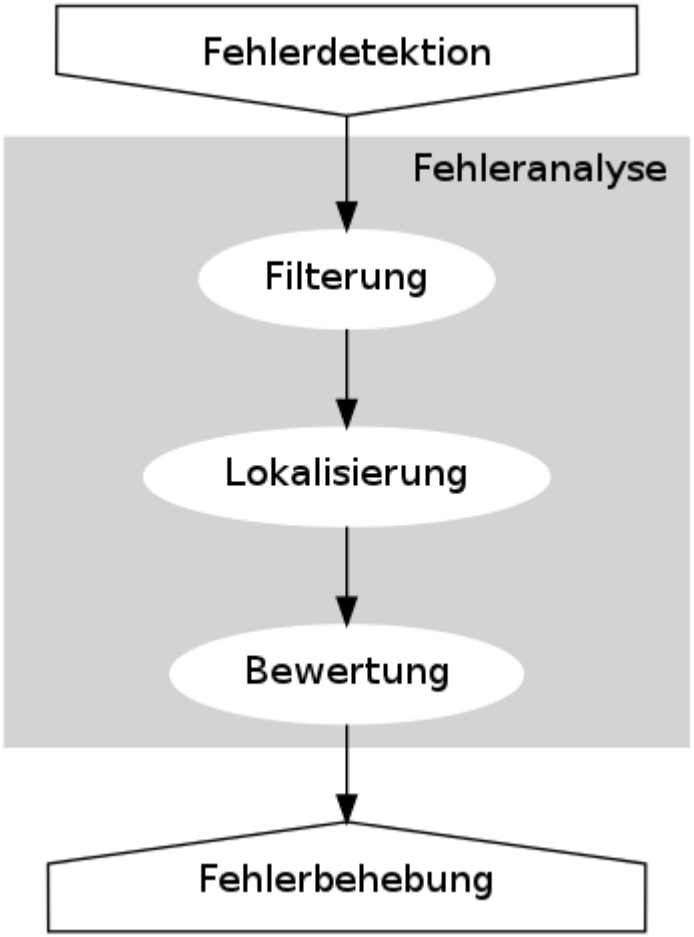
der Tests

effizienter gestalten

Quelle: http://www.wemo-tec.com/show.php?ID=4512_7 2012.04.27

- **Herangehensweise**
  - Fehlerbehandlungsprozeß
  - Schritt: Filterung
  - Typdistanz
  - allgemeine Verfahrensweisen
  - verbesserte Verfahrensweise

- **Herangehensweise**
  - **Fehlerbehandlungsprozeß**
  - Schritt: Filterung
  - Typdistanz
  - allgemeine Verfahrensweisen
  - verbesserte Verfahrensweise


Fehlerdetektion
Fehleranalyse
Filterung
Lokalisierung
Bewertung
Fehlerbehebung

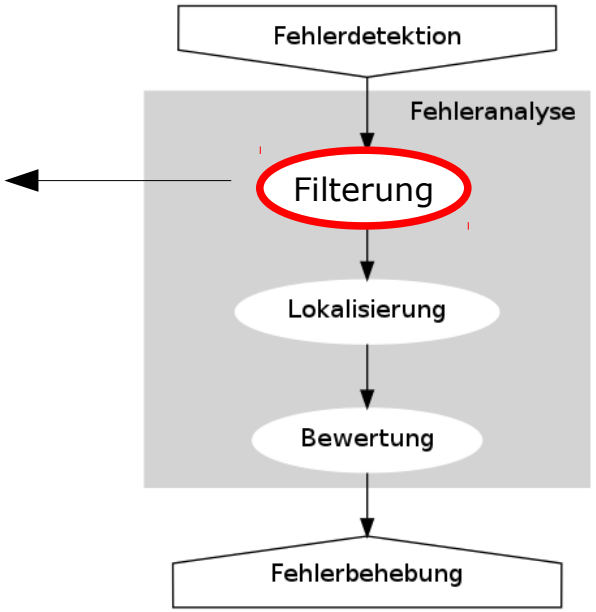
- **Herangehensweise**
  - Fehlerbehandlungsprozeß
  - **Schritt: Filterung**
  - Typdistanz
  - allgemeine Verfahrensweisen
  - verbesserte Verfahrensweise

Testergebnisse

Fehlschläge (+33)

Alle fehlgeschlagenen Tests

Testname
>>> org.apache.camel.management.JmxInstrumentationWithConnectorTest.testCounters
>>> org.apache.camel.management.ManagedCamelContextTracerTest.testCamelContextTracing
>>> org.apache.camel.management.ManagedCamelContextUpdateRoutesFromXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedCustomPolicyTest.testPolicy
>>> org.apache.camel.management.ManagedErrorHandlerOptionsTest.testManagedErrorHandlerOptions
>>> org.apache.camel.management.ManagedErrorHandlerRedeliveryTest.testManagedErrorHandlerRedelivery
>>> org.apache.camel.management.ManagedErrorHandlerTest.testManagedErrorHandler
>>> org.apache.camel.management.ManagedRegisterRouteTest.testRoutes
>>> org.apache.camel.management.ManagedRegisterTwoRoutesTest.testRoutes
>>> org.apache.camel.management.ManagedRouteDumpRouteAsXmlTest.testDumpAsXml
>>> org.apache.camel.management.ManagedRouteNoAutoStartupTest.testRouteNoAutoStartup
>>> org.apache.camel.management.ManagedRoutePolicyTest.testRoutes
>>> org.apache.camel.management.ManagedRouteRemoveTest.testRemove
>>> org.apache.camel.management.ManagedRouteShutdownAndStartTest.testShutdownAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartCleanupTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopAndStartTest.testStopAndStartRoute
>>> org.apache.camel.management.ManagedRouteStopTest.testStopRoute
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutTrue
>>> org.apache.camel.management.ManagedRouteStopWithAbortAfterTimeoutTest.testStopRouteWithAbortAfterTimeoutFalse

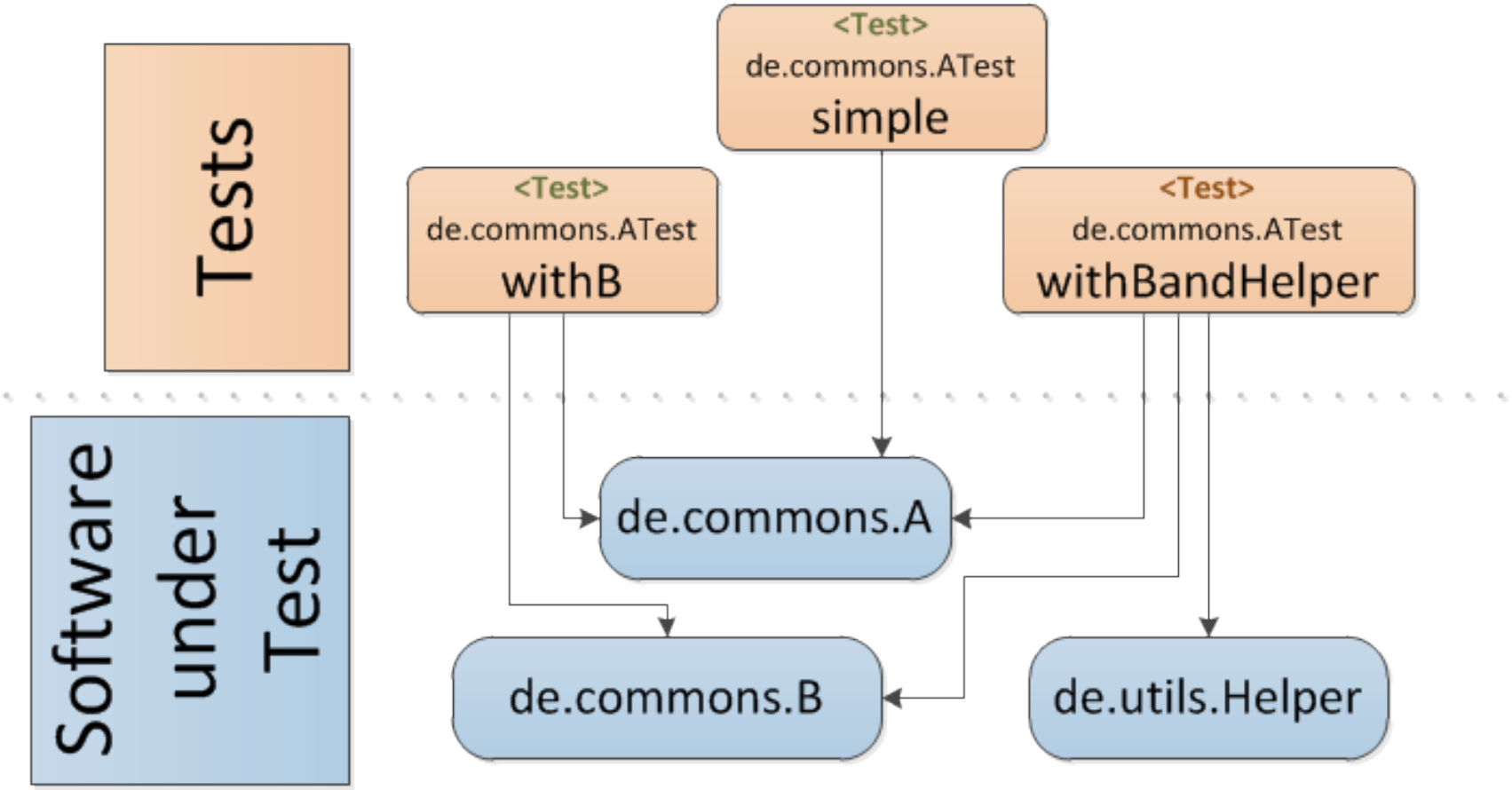


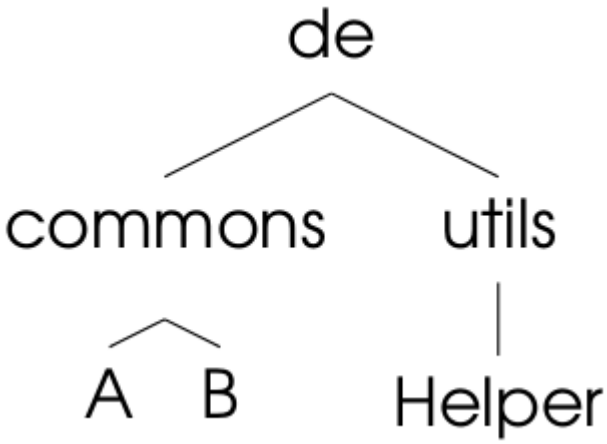
- **Herangehensweise**
  - Fehlerbehandlungsprozeß
  - Schritt: Filterung
  - **Typdistanz**
  - allgemeine Verfahrensweisen
  - verbesserte Verfahrensweise

- **Typdistanz**

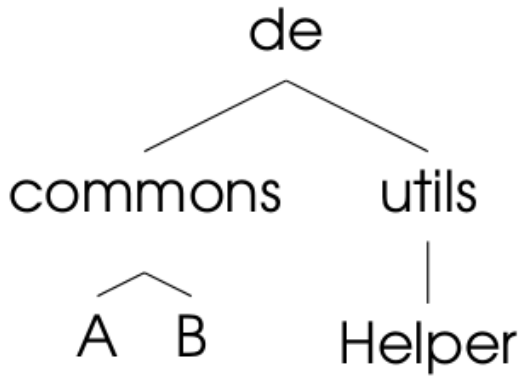
  - statische objektorientierte Softwaremetrik

  - Entfernung zweier Klassen (Typen)


Tests
<Test>
de.commons.ATest
simple
<Test>
de.commons.ATest
withB
<Test>
de.commons.ATest
withBandHelper
Software under Test
de.commons.A
de.commons.B
de.utils.Helper



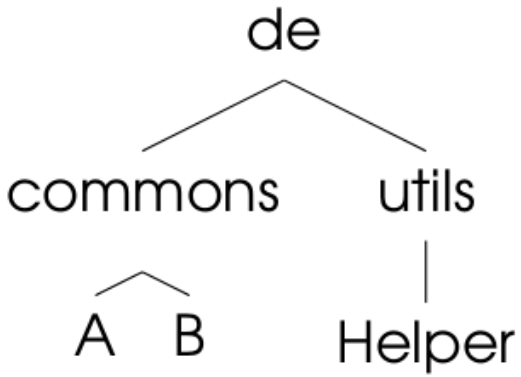
Namensraum als Baum



$$V = \{de, commons, A, B, utils, Helper\}$$

$$E = \{\{de, commons\}, \{commons, A\}, \{commons, B\}, \{de, utils\}, \{utils, Helper\}\}$$

$$M_{distance} = d(V_m, V_n)$$



$$d(V_0, V_1) = \begin{cases} 0 & \text{wenn } V_0 \text{ oder } V_1 \text{ Vertreter eines Typs des Standard-sprachumfangs sind} \\ 1 & \text{wenn } V_0 \text{ und } V_1 \text{ zwei Nachbarblätter sind} \\ x & \text{Anzahl der Kanten zwischen } V_0 \text{ und } V_1 + 1 \end{cases}$$

$$d(V_0, V_1) = \begin{cases} 0 & \text{wenn } V_0 \text{ oder } V_1 \text{ Vertreter eines Typs des Standard-sprachumfangs sind} \\ 1 & \text{wenn } V_0 \text{ und } V_1 \text{ zwei Nachbarblätter sind} \\ x & \text{Anzahl der Kanten zwischen } V_0 \text{ und } V_1 + 1 \end{cases}$$

V_0	V_1	$M_{distance}$
de.commons.A	de.commons.B	1
de.commons.A	de.utils.Helper	5 (4+1)
de.commons.A	java.lang.String	0

- **Einzeltypdistanz (genau eine Typdeklaration)**
- **Gesamttypdistanz (mehrere)**

Berechnung der Gesamttypdistanz

- 1. Maximale Einzeltypdistanz
- 2. Median der Einzeltypdistanzen
- 3. Anzahl der Anweisungen

$$Score_{FailedTests} = M_{distance_{max}} * 10^5 + \tilde{M}_{distance} * 10^3 + |S|$$

Bedeutung der Typdistanz

- Fehlerfortpflanzung
- Kleinster Test
- Kleinster Fehler

FACHHOCHSCHULE BRANDENBURG

Fehlerfortpflanzung

Name
net.sprd.qa.prototype.APIServiceTest#validateToken
net.sprd.qa.prototype.APIServiceTest#invalidateToken
net.sprd.qa.prototype.APIServiceTest#userNotExisting
net.sprd.qa.prototype.APIServiceTest#removeUserNotExisting
net.sprd.qa.prototype.Bus.BusConnectorTest#test
net.sprd.qa.prototype.Bus.BusServiceTest#smokeTest
net.sprd.qa.prototype.Bus.BusServiceTest#roundtrip

Kleinster Test

Name	Score
net.sprd.qa.prototype.APIServiceTest#validateToken	201013
net.sprd.qa.prototype.APIServiceTest#invalidateToken	201013
net.sprd.qa.prototype.APIServiceTest#userNotExisting	202010
net.sprd.qa.prototype.APIServiceTest#removeUserNotExisting	202010
net.sprd.qa.prototype.Bus.BusConnectorTest#test	301008
net.sprd.qa.prototype.Bus.BusServiceTest#smokeTest	904506
net.sprd.qa.prototype.Bus.BusServiceTest#roundtrip	1001018

Kleinster Fehler

Name	Score
net.sprd.qa.prototype.APIServiceTest#validateToken	201013
net.sprd.qa.prototype.APIServiceTest#invalidateToken	201013
net.sprd.qa.prototype.APIServiceTest#userNotExisting	202010
net.sprd.qa.prototype.APIServiceTest#removeUserNotExisting	202010
net.sprd.qa.prototype.Bus.BusConnectorTest#test	301008
net.sprd.qa.prototype.Bus.BusServiceTest#smokeTest	904506
net.sprd.qa.prototype.Bus.BusServiceTest#roundtrip	1001018

- **Herangehensweise**
  - Fehlerbehandlungsprozeß
  - Schritt: Filterung
  - Typdistanz
  - **allgemeine Verfahrensweisen**
  - verbesserte Verfahrensweise

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell			

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste		

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert			

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel		

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	

Bekannte Verfahrensweisen

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	$\ominus$
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	$neutral$

- **Herangehensweise**
  - Fehlerbehandlungsprozeß
  - Schritt: Filterung
  - Typdistanz
  - allgemeine Verfahrensweisen
  - **verbesserte Verfahrensweise**

verbesserte Verfahrensweise

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	*neutral*
bottom-up			

verbesserte Verfahrensweise

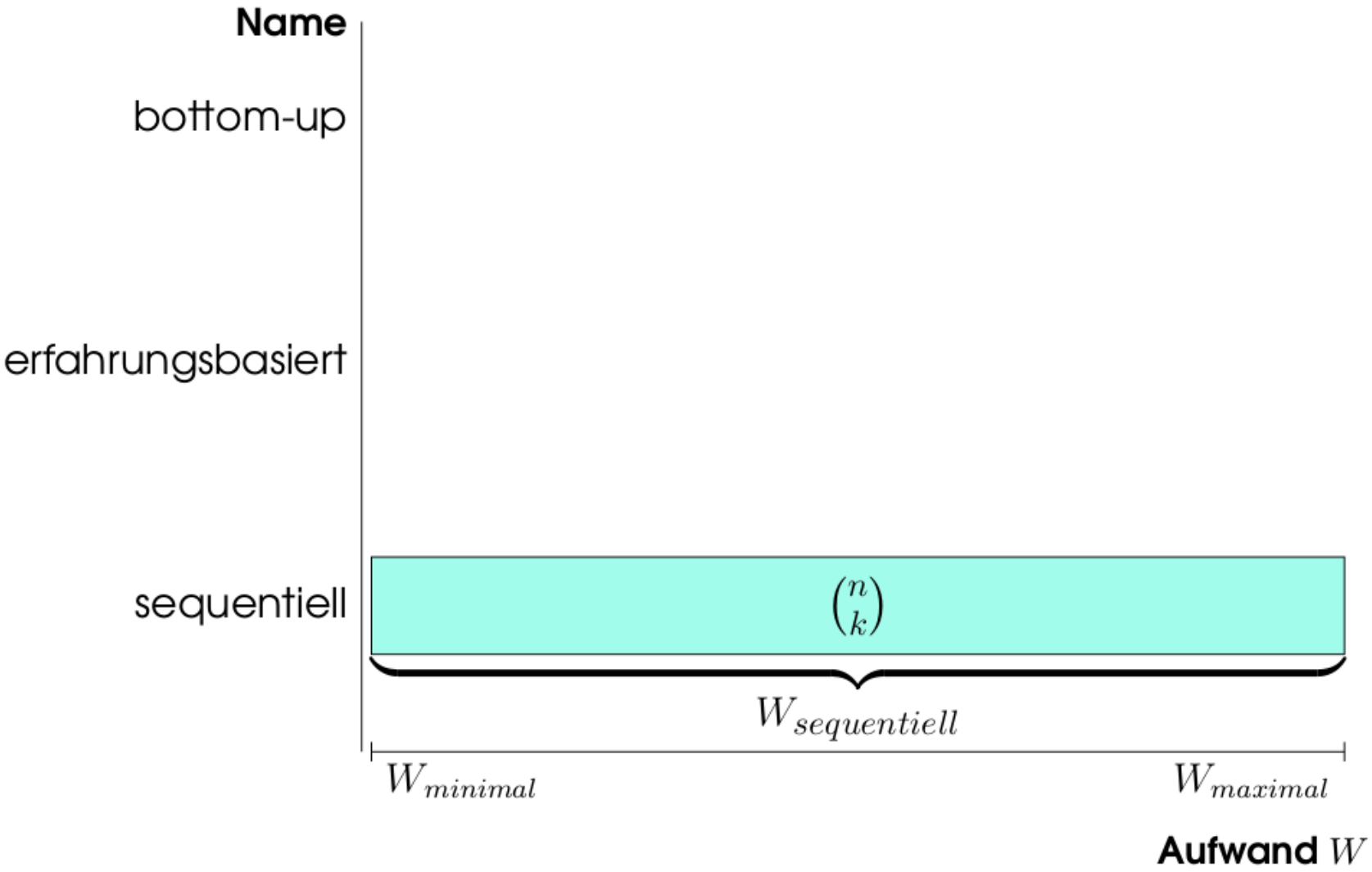
Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	*neutral*
bottom-up	Kriterien für Sortierung, Werkzeugunterstützung notwendig		

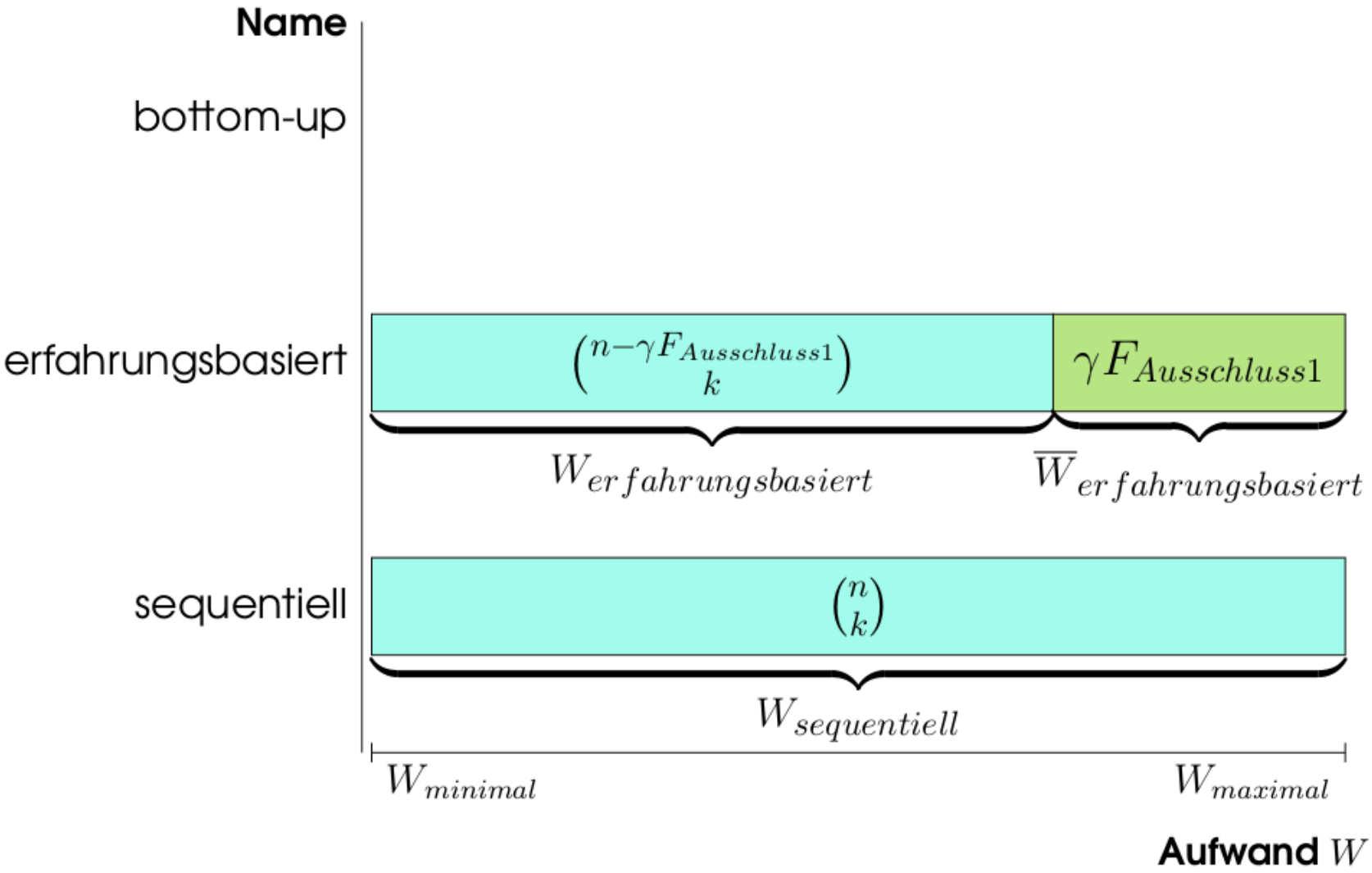
verbesserte Verfahrensweise

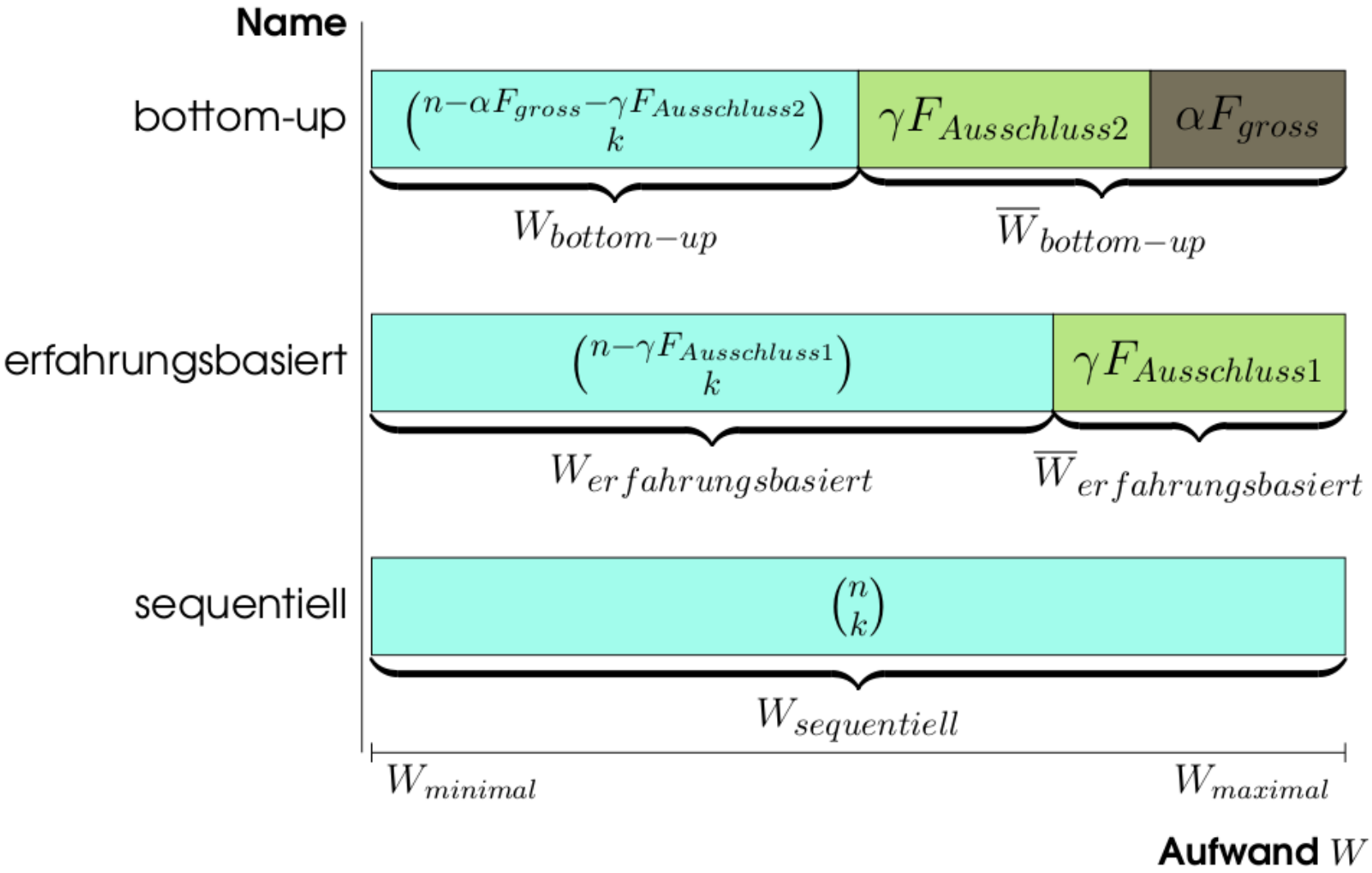
Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	*neutral*
bottom-up	Kriterien für Sortierung, Werkzeugunterstützung notwendig	geringster Aufwand	

verbesserte Verfahrensweise

Strategie	Schwächen (Risiken)	Stärken (Vorteile)	aufwandsstabil
sequentiell	Reihenfolge in der Fehlerliste	einfach	⊖
erfahrungsbasiert	fehlerhafte Filterung, Erfahrungsmangel	nutzt Erfahrung	*neutral*
bottom-up	Kriterien für Sortierung, Werkzeugunterstützung notwendig	geringster Aufwand	⊕


Name
bottom-up
erfahrungsbasiert
sequentiell
$\binom{n}{k}$
$W_{sequentiell}$
$W_{minimal}$
$W_{maximal}$
Aufwand W


Name
bottom-up
erfahrungsbasiert
$\binom{n-\gamma F_{Ausschluss1}}{k}$
$\gamma F_{Ausschluss1}$
$W_{erfahrungsbasiert}$
$\overline{W}_{erfahrungsbasiert}$
sequentiell
$\binom{n}{k}$
$W_{sequentiell}$
$W_{minimal}$
$W_{maximal}$
Aufwand W


Name
bottom-up
$\binom{n-\alpha F_{gross}-\gamma F_{Ausschluss2}}{k}$
$\gamma F_{Ausschluss2}$
αF_{gross}
$W_{bottom-up}$
$\overline{W}_{bottom-up}$
erfahrungsbasiert
$\binom{n-\gamma F_{Ausschluss1}}{k}$
$\gamma F_{Ausschluss1}$
$W_{erfahrungsbasiert}$
$\overline{W}_{erfahrungsbasiert}$
sequentiell
$\binom{n}{k}$
$W_{sequentiell}$
$W_{minimal}$
$W_{maximal}$
Aufwand W

- Ziel der Arbeit
- Motivation
- State of the Art
- Methodik
- **Zusammenfassung/Ausblick**

Zusammenfassung

- Implementierung in Java
- Integration in Junit/TestNG/Maven
- Berechnung der Metrik mit Sonar
- Untersuchung durchgeführt

Ausblick

- empirische Bestätigung fehlt
- Projekt öffentlich unter

  http://code.google.com/p/metricanalyzer/

Danke für Ihre Aufmerksamkeit!