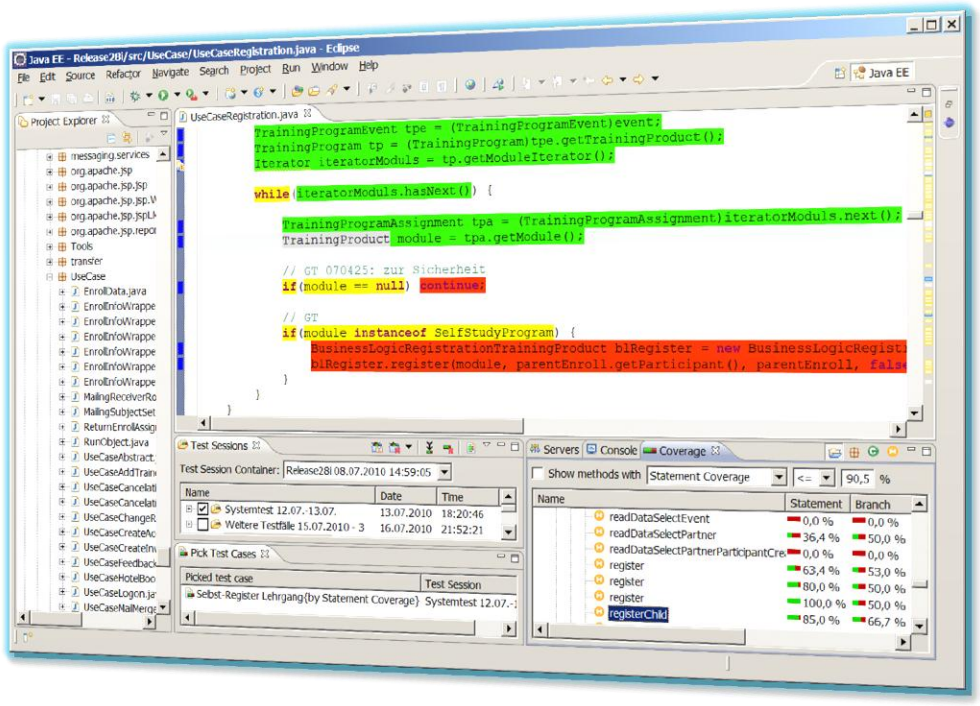
Testsuite-Optimierung mit CodeCover



Rainer Schmidberger

schmidrr@informatik.uni-stuttgart.de

Universität Stuttgart, Institut für Softwaretechnologie, SE

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Testsuite-Optimierung mit CodeCover

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Übersicht:

- Test und Glass-Box-Test (GBT)
- CodeCover „Eckdaten“
- Testfallherleitung mit CodeCover
- Spezielle Überdeckungskriterien
- Getting Started
- Weitere Release-Planung

Übersicht über die Prüfverfahren

„Testen ist die Ausführung eines Programms mit der Absicht, Fehler zu finden“

[Myers, 1979]

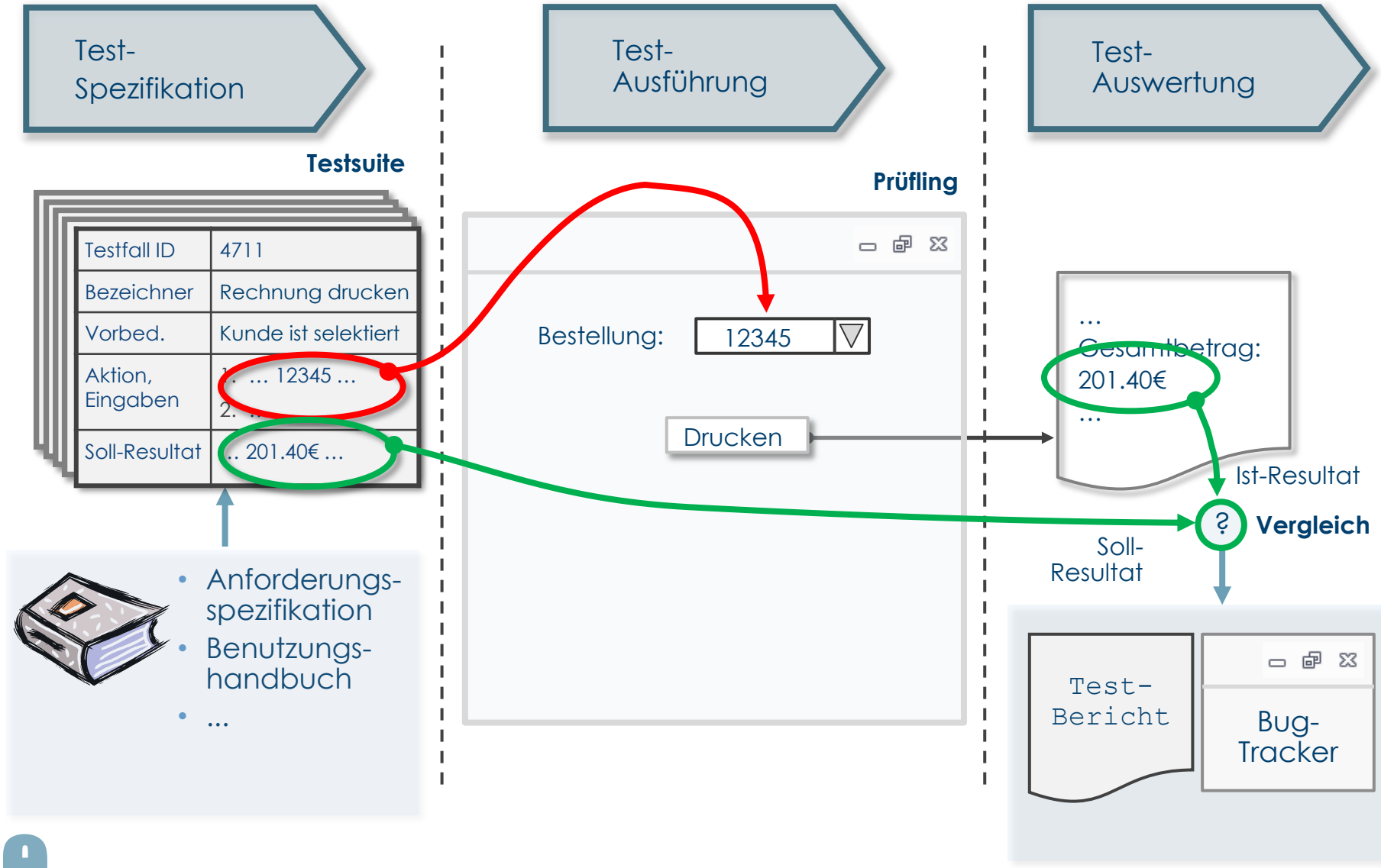
mit

Dokumentation der genauen Ausführungs-bedingungen wie z.B. Version, Testfälle und Resultate.

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Black-Box-Test



© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Glass-Box-Test =White-Box-Test, Coverage Test

Test-Spezifikation

Test-Ausführung

Test-Auswertung

Testsuite

Testfall ID	4711
Bezeichner	Rechnung drucken
Vorbed.	Kunde ist selektiert
Aktion, Eingaben	1. … 2. …
Soll-Resultat	.. 201.40€ …

Prüfling

```
double endBetrag(Kunde kunde, double b) {

  // normaler Kunde
  double rabatt = 0.0;

  if(kunde.getUmsatz() > 2000) {
    // Alle Großkunden erhalten 10% Rabatt
    rabatt = 0.1;

    if(kunde.istGewerblich()) {
      // Gew. Kunden erhalten weitere 5% Rabatt
      rabatt += 0.05;
    }
  }
  double endBetrag = betrag * (1.0 - rabatt);
  return endBetrag;
}
```

…
Gesamtbetrag:
201.40€
…

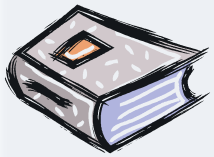
Ist-Resultat

? **Vergleich**

Soll-Resultat

Test-Bericht

Bug-Tracker



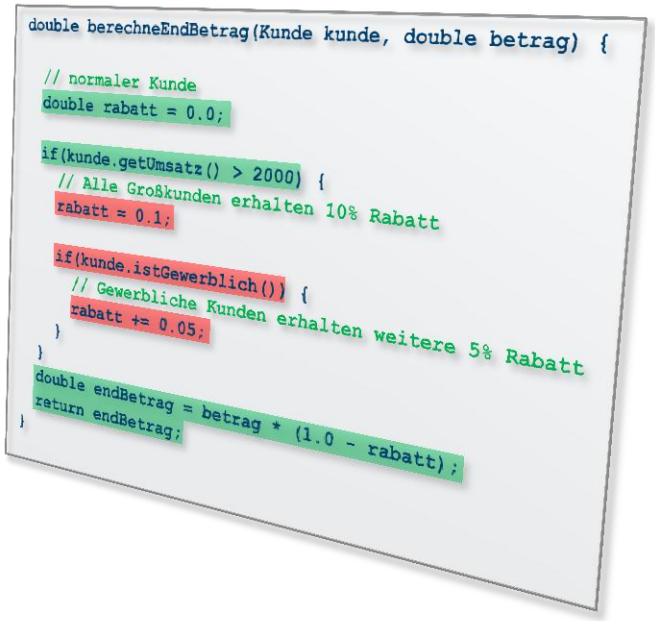
- Anforderungs-spezifikation
- Benutzungs-handbuch
- …

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Glass-Box-Test

- Der GBT ist ein Test, bei dem wir **zusätzlich beobachten**, wie weit der Prüfling ausgeführt wird.
- Den Anteil des ausgeführten Programms am gesamten Programm bezeichnet man dabei als **GBT-Überdeckung**.
- Der Tester ist bestrebt, die Überdeckung **in Richtung 100% zu erhöhen**.



```
double berechneEndBetrag(Kunde kunde, double betrag) {

  // normaler Kunde
  double rabatt = 0.0;

  if(kunde.getUmsatz() > 2000) {
    // Alle Großkunden erhalten 10% Rabatt
    rabatt = 0.1;

    if(kunde.istGewerblich()) {
      // Gewerbliche Kunden erhalten weitere 5% Rabatt
      rabatt += 0.05;
    }
  }
  double endBetrag = betrag * (1.0 - rabatt);
  return endBetrag;
}
```

Hintergrund:

- Auch ein gründlicher Funktionstest führt bei Programmen realer Größe **bei Weitem zu keiner vollständigen GBT-Überdeckung**.
- Der beim GBT neu überdeckte Programmcode hat eine **ähnliche Fehlerdichte** wie der restliche Programmcode.
- Mit vertretbarem Aufwand (d. h. bei hohem Nutzen!) lässt sich eine etwa **80%ige GBT-Überdeckung erreichen**.

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Überdeckungen

Überdeckung	Ausprägung	Beispiel
Anweisung/ Statement	ausgeführt / nicht ausgeführt	`System.out.println("Hallo CodeCover!");`
Block (Zweig)/ Branch	ausgeführt / nicht ausgeführt	`if(i == 99) { System.out.println("Hallo World!"); } else { System.out.println("Hallo Stuttgart!"); }`
Term	true- und false-Wirksamkeit je Einzelterm	`if(i > 5 || j > 1 || k == j) { System.out.println("Termüberdeckung"); }`
?-Operator	Ausdrücke wirksam: keiner / einer von beiden / beide	`k = i < 3 ? 2 : 1;`
Schleife	Schleifenkörper: nicht / einmal / mehrfach ausgeführt	`while(i < 5) { System.out.println("Schleifenüberdeckung"); i++; }`

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

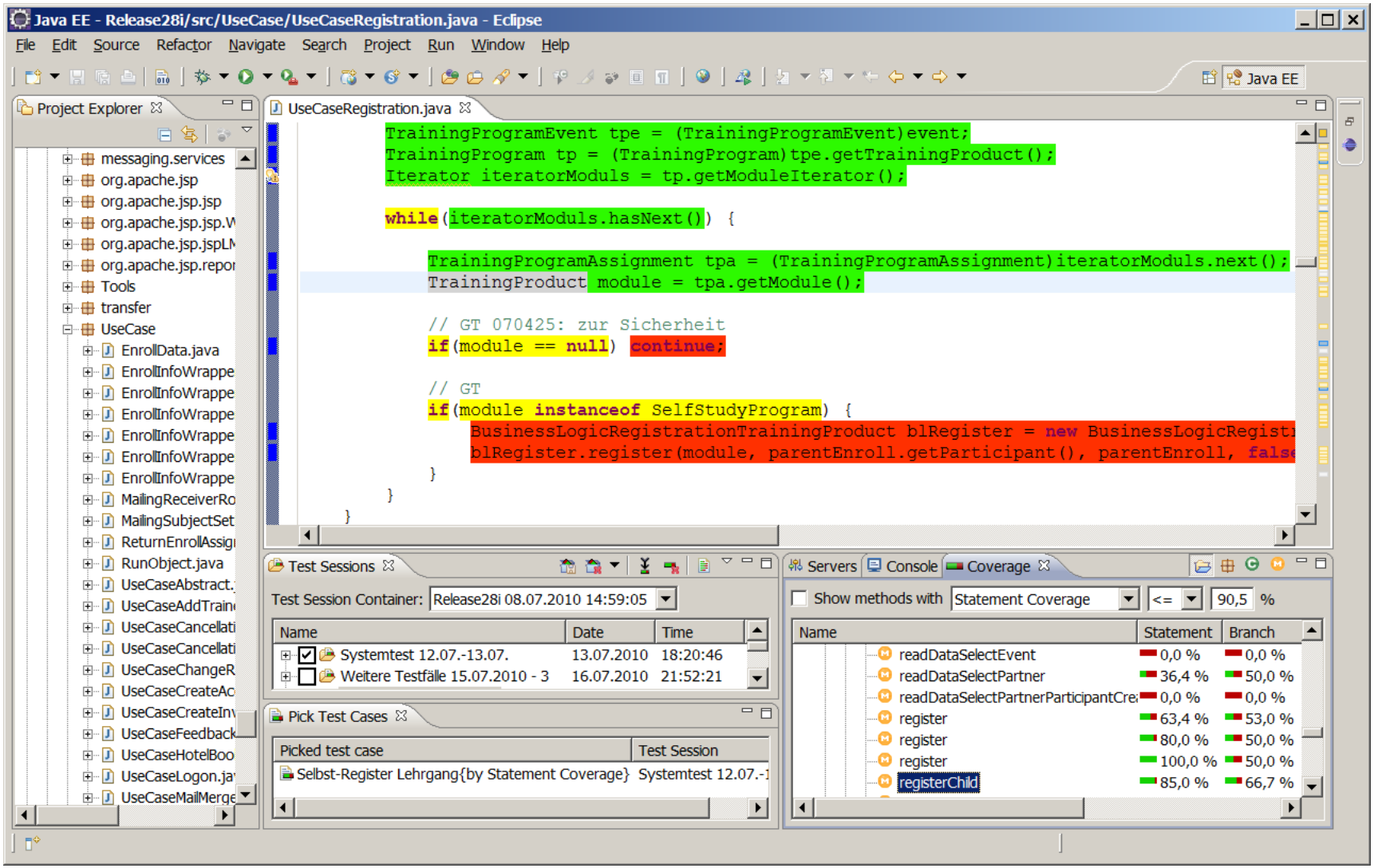
Testsuite-Optimierung mit CodeCover

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Übersicht:

- ☑ Test und Glass-Box-Test (GBT)
- ❑ **CodeCover „Eckdaten“**
- ❑ Testfallherleitung mit CodeCover
- ❑ Spezielle Überdeckungskriterien
- ❑ Getting Started
- ❑ Weitere Release-Planung

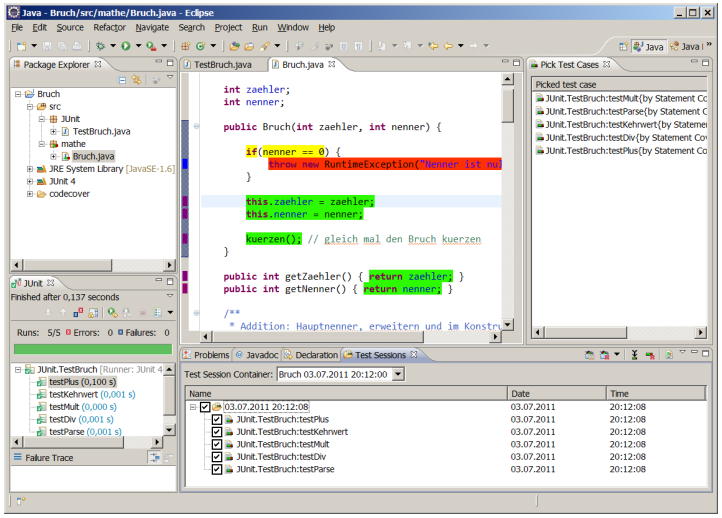
GBT-Werkzeug CodeCover



© 2011 Rainer Schmidberger Java Forum Stuttgart 2011

Siehe http://www.iste.uni-stuttgart.de/se/menschen/schmidberger/glassboxtools.html

GBT-Werkzeug CodeCover



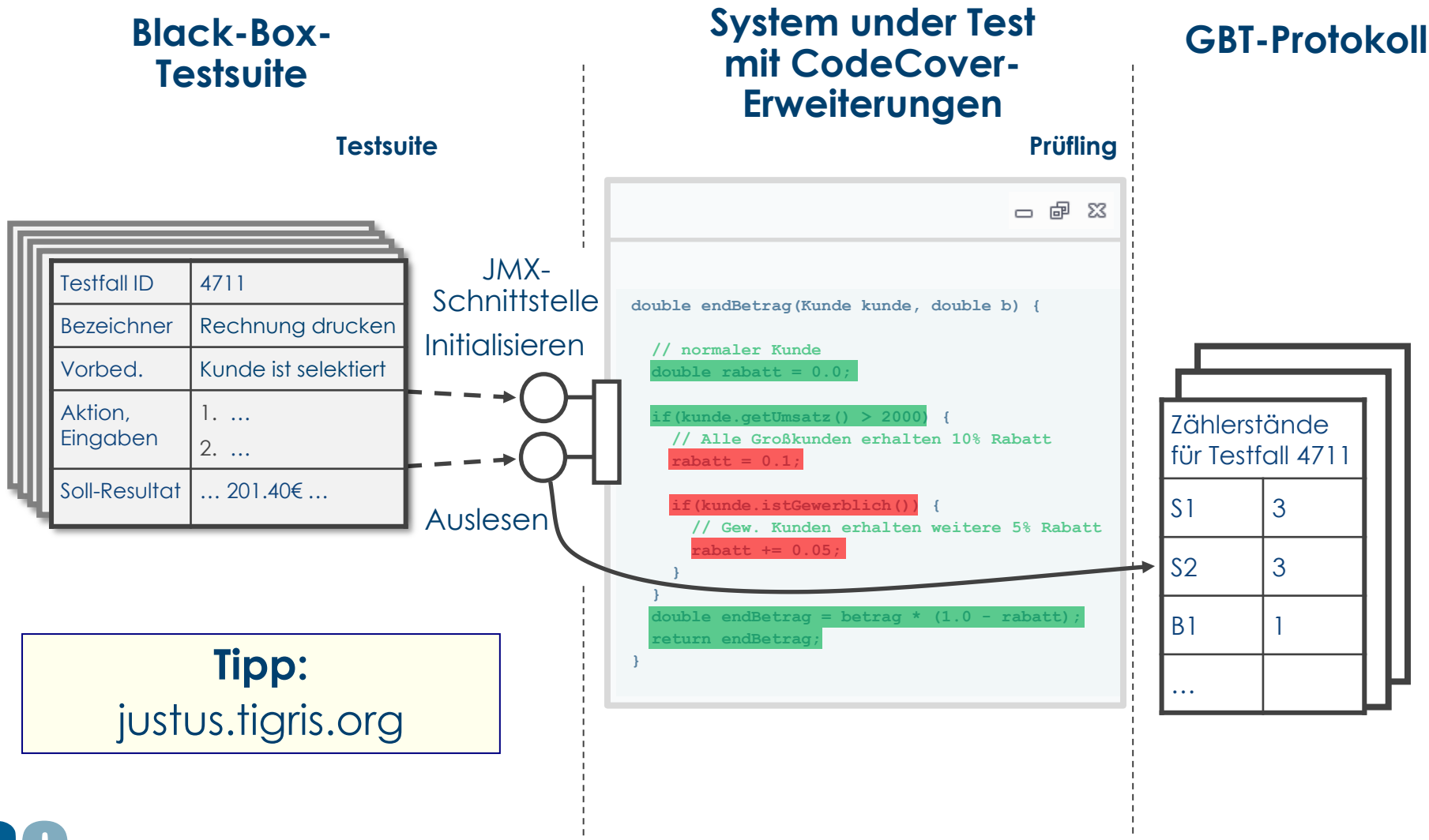
www.codecover.org

- Entwicklung 2007 an der Uni-Stuttgart, ISTE; dann Open-Source, EPL
- sourceforge.net/projects/codecover
- Überdeckungen
  - ⇨ Anweisung, Zweig, Schleifen
  - ⇨ Term
  - ⇨ ?-Operator
  - ⇨ Synchronisation
- JMX und JUnit-Schnittstellen
- Eclipse Plugin, Ant, Batch
- Programmiersprachen
  - ⇨ Java
  - ⇨ COBOL
  - ⇨ …

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

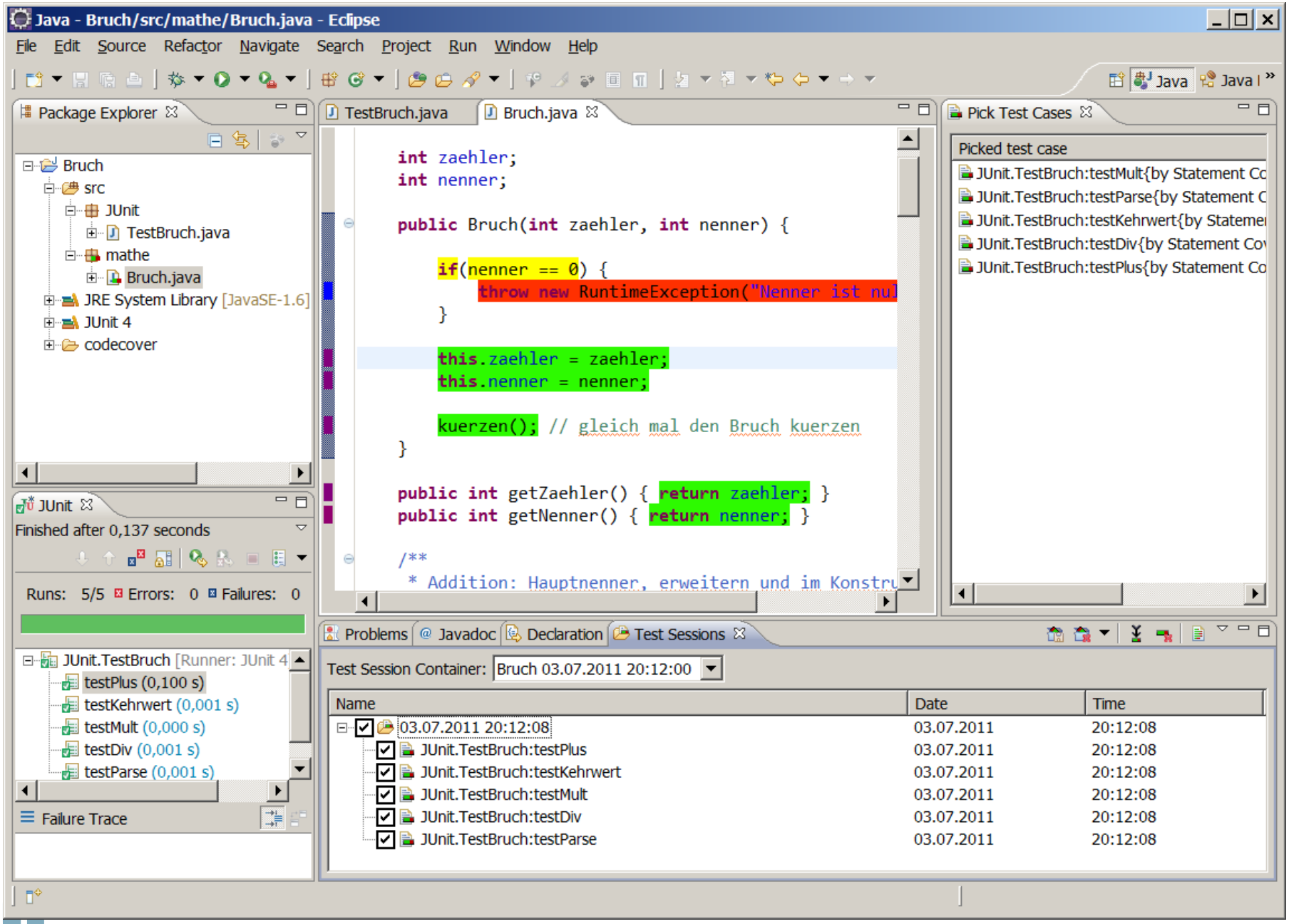
CodeCover JMX-Schnittstelle



Tipp:
justus.tigris.org

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de



© 2011 Rainer Schmidberger

Testsuite-Optimierung mit CodeCover

Übersicht:

- ☑ Test und Glass-Box-Test (GBT)
- ☑ CodeCover „Eckdaten“
- ❑ **Testfallherleitung mit CodeCover**
- ❑ Spezielle Überdeckungskriterien
- ❑ Getting Started
- ❑ Weitere Release-Planung

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Testsuite-Erweiterung

Protokoll beim „klassischen“ GBT:

Name	Statement	Branch	Loop	Term	?-Operator	Synchronized
ActionHandlerHoliday	90,9 %	38,9 %	–	38,9 %	50,0 %	–
ActionHandlerHotelBooking	100,0 %	100,0 %	–	100,0 %	–	–
ActionHandlerHotelBookingProcess	100,0 %	–	–	–	–	–
ActionHandlerHotelRoom	85,7 %	50,0 %	–	50,0 %	–	–
ActionHandlerInvoice	55,6 %	35,0 %	0,0 %	31,8 %	–	–
ActionHandlerMailMerge	55,6 %	68,2 %	0,0 %	60,7 %	–	–
ActionHandlerMailMergeView	45,2 %	30,0 %	0,0 %	23,5 %	–	–
ActionHandlerMergeEcadiaObjects	100,0 %	–	–	–	–	–
ActionHandlerMessageBroker	0,0 %	0,0 %	–	0,0 %	–	–
ActionHandlerMonitoringCompetencesM	0,0 %	0,0 %	0,0 %	0,0 %	–	–
ActionHandlerNews	44,9 %	29,2 %	0,0 %	26,9 %	50,0 %	–
ActionHandlerNewsletter	0,0 %	0,0 %	–	0,0 %	–	–
ActionHandlerNewsletterEdition	0,0 %	0,0 %	–	0,0 %	–	–
ActionHandlerNewsletterRegistration	0,0 %	0,0 %	0,0 %	0,0 %	–	–
ActionHandlerNewsletterSubscription	0,0 %	0,0 %	–	0,0 %	–	–
ActionHandlerOptions	100,0 %	100,0 %	–	100,0 %	–	–
ActionHandlerOrder	69,6 %	50,0 %	–	45,0 %	–	–

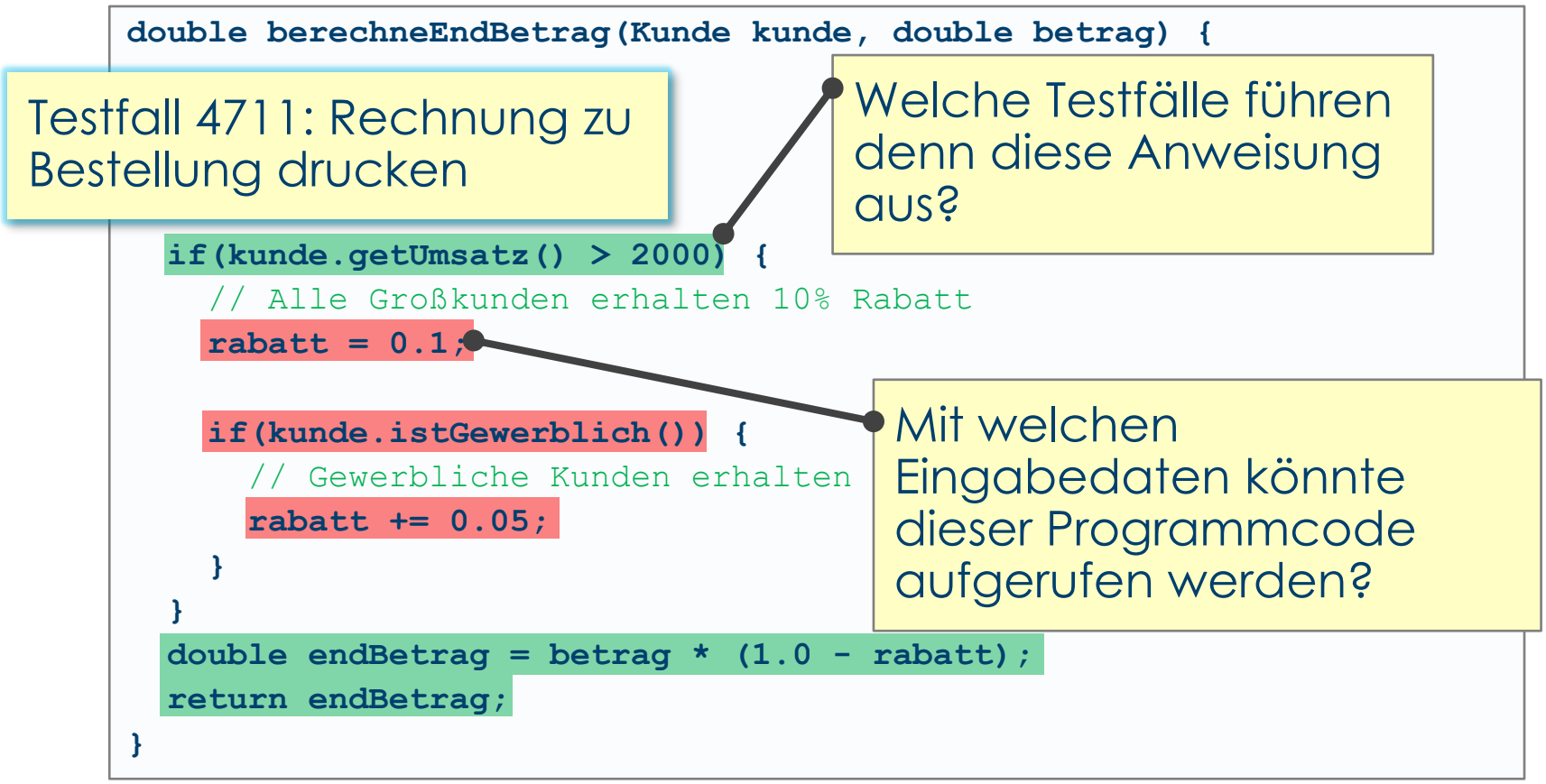
Aber welche Testfälle fehlen hier?

Es gibt keine Testfälle zu „Newsletter“

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

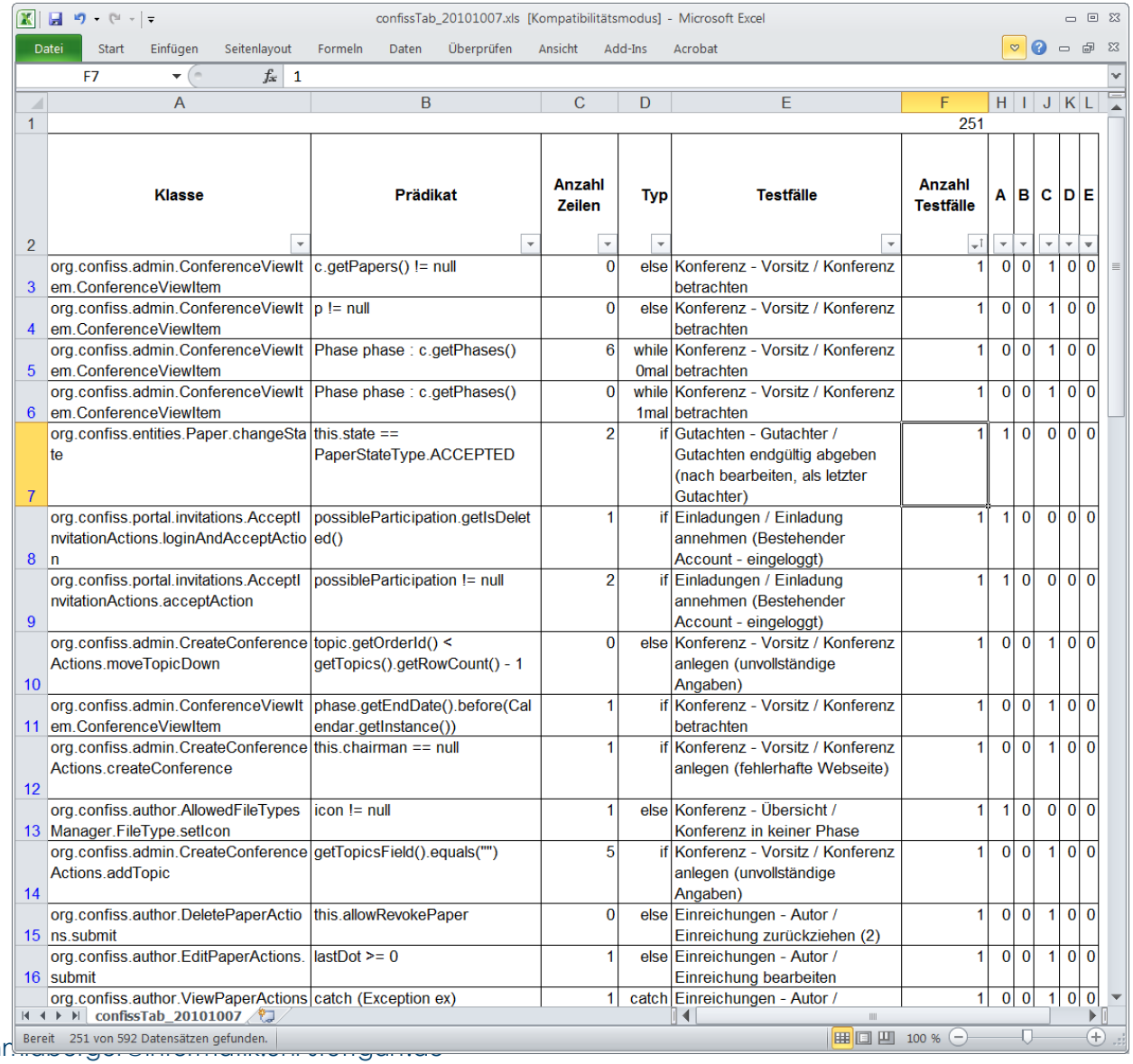
CodeCover: Testfallherleitung



© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Beispiel: Testfall-Hinweise zu „ConfISS“



confissTab_20101007.xls [Kompatibilitätsmodus] - Microsoft Excel

Datei | Start | Einfügen | Seitenlayout | Formeln | Daten | Überprüfen | Ansicht | Add-Ins | Acrobat

F7: 1

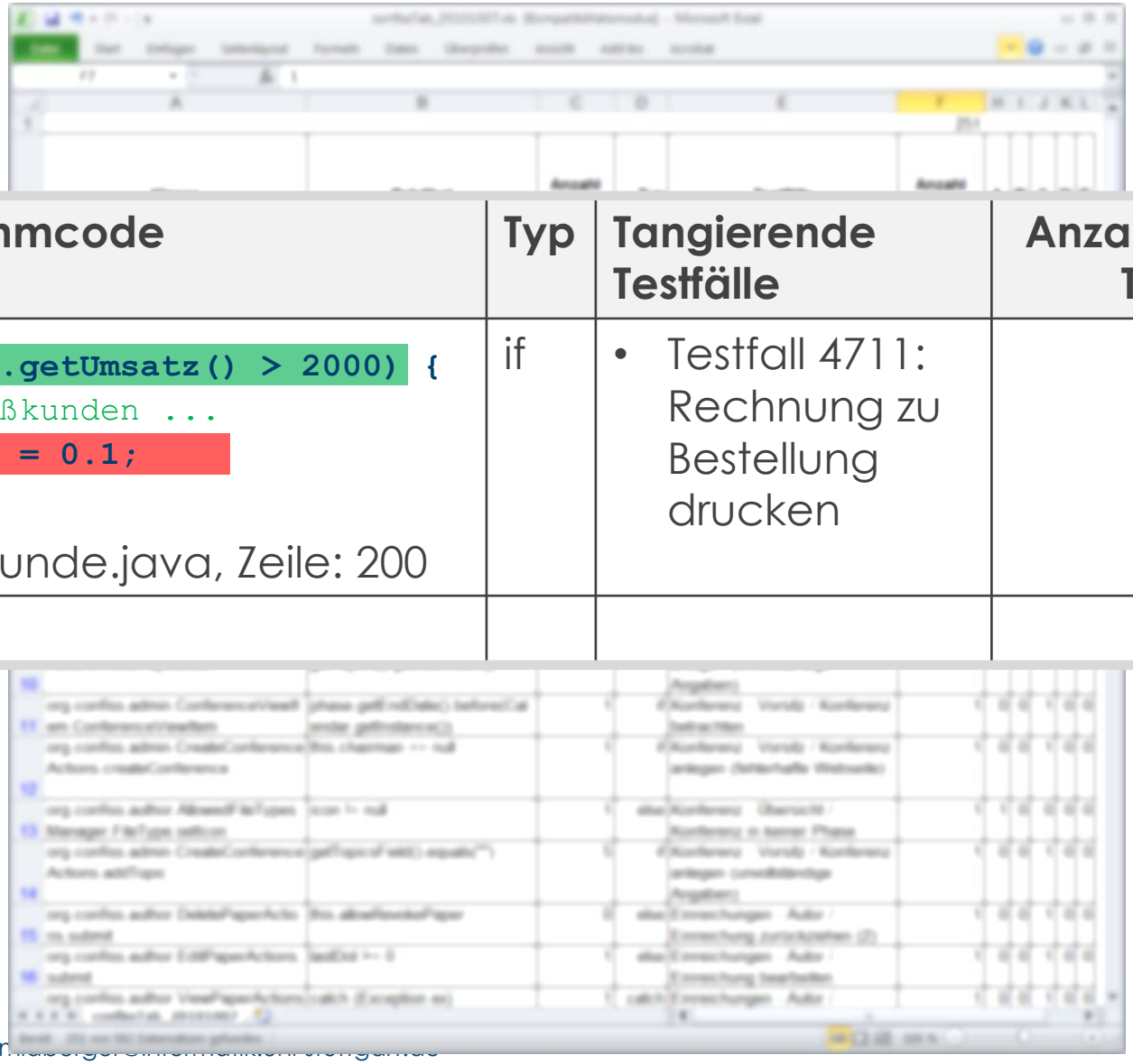
	A	B	C	D	E	F	H	I	J	K	L
1						251					
2	**Klasse**	**Prädikat**	**Anzahl Zeilen**	**Typ**	**Testfälle**	**Anzahl Testfälle**	**A**	**B**	**C**	**D**	**E**
3	org.confiss.admin.ConferenceViewItem.ConferenceViewItem	c.getPapers() != null	0	else	Konferenz - Vorsitz / Konferenz betrachten	1	0	0	1	0	0
4	org.confiss.admin.ConferenceViewItem.ConferenceViewItem	p != null	0	else	Konferenz - Vorsitz / Konferenz betrachten	1	0	0	1	0	0
5	org.confiss.admin.ConferenceViewItem.ConferenceViewItem	Phase phase : c.getPhases()	6	while 0mal	Konferenz - Vorsitz / Konferenz betrachten	1	0	0	1	0	0
6	org.confiss.admin.ConferenceViewItem.ConferenceViewItem	Phase phase : c.getPhases()	0	while 1mal	Konferenz - Vorsitz / Konferenz betrachten	1	0	0	1	0	0
7	org.confiss.entities.Paper.changeState	this.state == PaperStateType.ACCEPTED	2	if	Gutachten - Gutachter / Gutachten endgültig abgeben (nach bearbeiten, als letzter Gutachter)	1	1	0	0	0	0
8	org.confiss.portal.invitations.AcceptInvitationActions.loginAndAcceptAction	possibleParticipation.getIsDeleted()	1	if	Einladungen / Einladung annehmen (Bestehender Account - eingeloggt)	1	1	0	0	0	0
9	org.confiss.portal.invitations.AcceptInvitationActions.acceptAction	possibleParticipation != null	2	if	Einladungen / Einladung annehmen (Bestehender Account - eingeloggt)	1	1	0	0	0	0
10	org.confiss.admin.CreateConferenceActions.moveTopicDown	topic.getOrderId() < getTopics().getRowCount() - 1	0	else	Konferenz - Vorsitz / Konferenz anlegen (unvollständige Angaben)	1	0	0	1	0	0
11	org.confiss.admin.ConferenceViewItem.ConferenceViewItem	phase.getEndDate().before(Calendar.getInstance())	1	if	Konferenz - Vorsitz / Konferenz betrachten	1	0	0	1	0	0
12	org.confiss.admin.CreateConferenceActions.createConference	this.chairman == null	1	if	Konferenz - Vorsitz / Konferenz anlegen (fehlerhafte Webseite)	1	0	0	1	0	0
13	org.confiss.author.AllowedFileTypesManager.FileType.setIcon	icon != null	1	else	Konferenz - Übersicht / Konferenz in keiner Phase	1	1	0	0	0	0
14	org.confiss.admin.CreateConferenceActions.addTopic	getTopicsField().equals("")	5	if	Konferenz - Vorsitz / Konferenz anlegen (unvollständige Angaben)	1	0	0	1	0	0
15	org.confiss.author.DeletePaperActions.submit	this.allowRevokePaper	0	else	Einreichungen - Autor / Einreichung zurückziehen (2)	1	0	0	1	0	0
16	org.confiss.author.EditPaperActions.submit	lastDot >= 0	1	else	Einreichungen - Autor / Einreichung bearbeiten	1	0	0	1	0	0
	org.confiss.author.ViewPaperActions	catch (Exception ex)	1	catch	Einreichungen - Autor /	1	0	0	1	0	0

confissTab_20101007

Bereit 251 von 592 Datensätzen gefunden. 100 %

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Beispiel-Testfall-Hinweise zu „ConfISS“



Programmcode	Typ	Tangierende Testfälle	Anzahl tang. Testfälle	…

`if(kunde.getUmsatz() > 2000) {`
`  // Großkunden ...`
`  rabatt = 0.1;`
`}`
Datei: Kunde.java, Zeile: 200
	if	• Testfall 4711: Rechnung zu Bestellung drucken	1	…

…				

© 2011 Rainer Schmidberger

Priorisierungen der Testfall-Hinweise

In der Praxis ist mit **sehr vielen Testfall-Hinweisen** zu rechnen. Um diese auf ein **praktikables Maß zu reduzieren**, sind aber Priorisierungen möglich:

- Bezogen auf die tangierenden Testfälle
  - ⇨ Priorität der tangierenden Testfälle
  - ⇨ Anzahl der tangierenden Testfälle
  - ⇨ …
- Bezogen auf die Programmcodestelle
  - ⇨ Typ des dominierenden GBT-Elements (if, else, …)
  - ⇨ Programmcode Fehlerprognosen (FindBugs)
  - ⇨ Versionshistorie
  - ⇨ …

Hierzu gleich noch mehr unter Release-Planung

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Fallstudien

Produkt	Anzahl Anweisungen	Anzahl Testfälle	Anw.-überd.	Tangierte if-Zweige	Mit Priorität	< 3 tang. Testfälle
A	6.329	154	76%	108	84	7
B	3.371	181	78%	10	4	4
C	160.607	211	51%	1.012	343	42*

*Aus den 42 Testfall-Hinweisen werden 14 neue Testfälle entwickelt, die etwa 700 Anweisungen zusätzlich ausführen (das entspricht etwa einer um 0,5% höheren Anweisungsüberdeckung).

Von den 14 neuen Testfällen wird ein bislang unbekannter Fehler angezeigt.

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Testsuite-Optimierung mit CodeCover

Übersicht:

- ☑ Test und Glass-Box-Test (GBT)
- ☑ CodeCover „Eckdaten“
- ☑ Testfallherleitung mit CodeCover
- ❑ **Spezielle Überdeckungskriterien**
- ❑ Getting Started
- ❑ Weitere Release-Planung

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Synchronized-Überdeckung

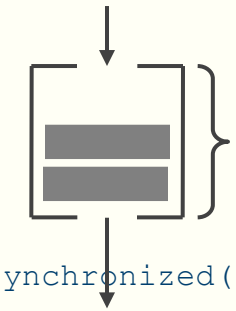
```
static void m() {
  // ...

  Object lock = ...

  synchronized(lock) {

    // ... der kritische Abschnitt ...


  }
}
```



y(lock) : Anzahl der Threads, die auf lock warten

- Mit `synchronized` wird für ein Lock-Objekt sichergestellt, dass ein Code-Block nur von einem Thread zeitgleich betreten wird
- Mögliche Fehler bei Verwendung von `synchronized` sind Flaschenhälse und Deadlocks
- Funktionale Tests prüfen auf diese Fehler nicht gezielt – ein Vollständigkeitskriterium „Synchronized-Coverage“ (race-Coverage) kann auf fehlende Testfälle hinweisen

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Synchronized Überdeckung

SyncTest.java

```
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    static void m(String threadName, int waitTimeInMethodeM, Object mutex) {
        try {

            java.text.DateFormat format = new java.text.SimpleDateFormat("HH:mm:

            System.out.println(threadName + " - waiting at " + format.format(Cal

            synchronized(mutex) {

                System.out.println(threadName + " - starts m at " + format.forma
                Thread.sleep(waitTimeInMethodeM);
                System.out.println(threadName + " - ends m at " + format.format(
            }

        } catch(Exception e) {
```

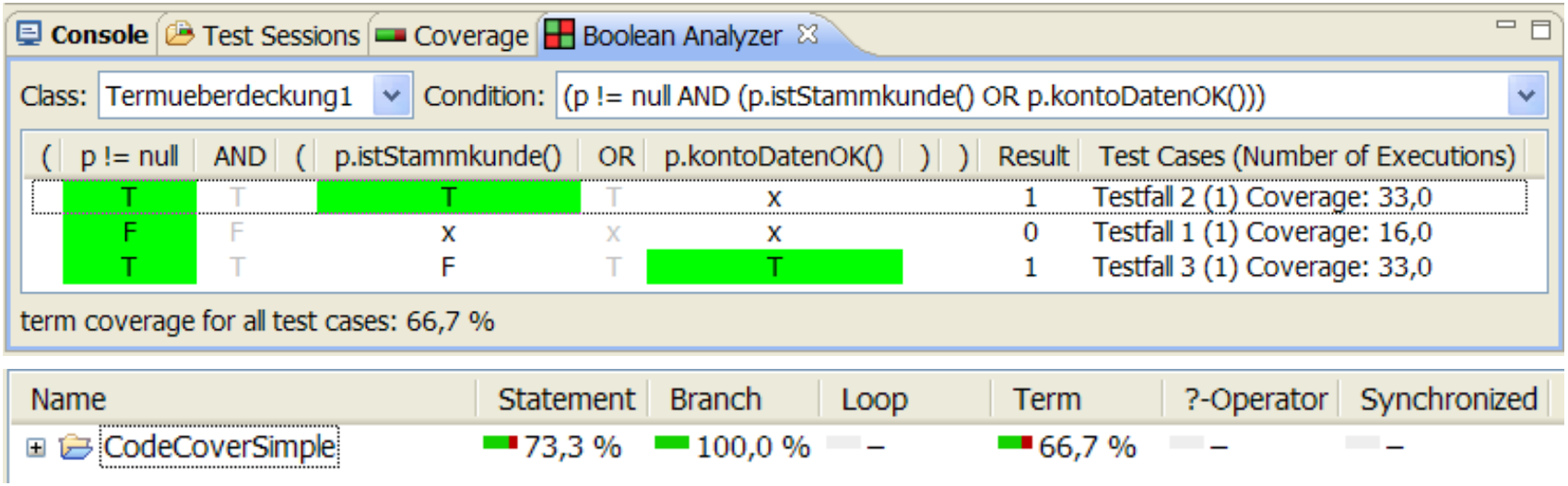
Name	Statement	Branch	Loop	Term	?-Operator	Synchronized
SyncTest	74,4 %	25,0 %	33,3 %	–	–	66,7 %

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Termüberdeckung

```
static void test(Person p) {
    if(p != null && (p.istStammkunde() || p.kontoDatenOK())) {
        System.out.print("Bestellung ist OK");
    }
}
```



© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

Testsuite-Optimierung mit CodeCover

© 2011 Rainer Schmidberger

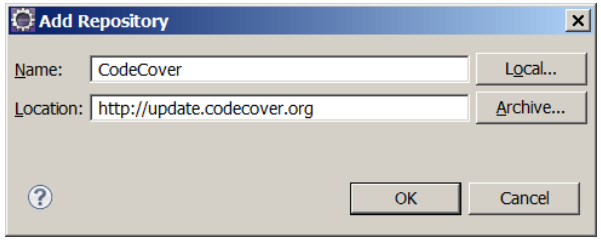
Übersicht:

- ☑ Test und Glass-Box-Test (GBT)
- ☑ CodeCover „Eckdaten“
- ☑ Testfallherleitung mit CodeCover
- ☑ Spezielle Überdeckungskriterien
- ❑ **Getting Started**
- ❑ Weitere Release-Planung

CodeCover in eclipse – Getting started

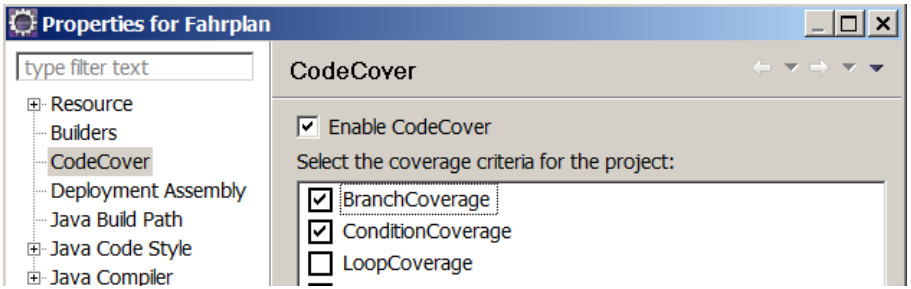
Installation

Install new Software …
http://update.codecover.org



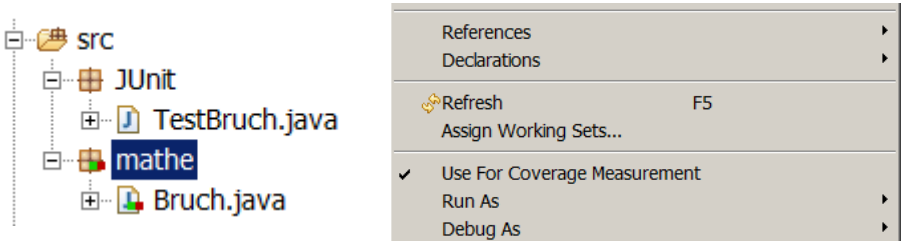
Aktivieren

Projekt-Properties

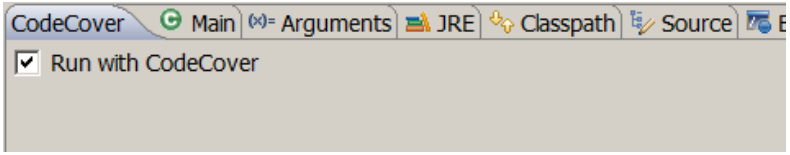


Pakete auswählen

Kontextmenü im Projekt-Explorer



Run-Konfiguration



© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de

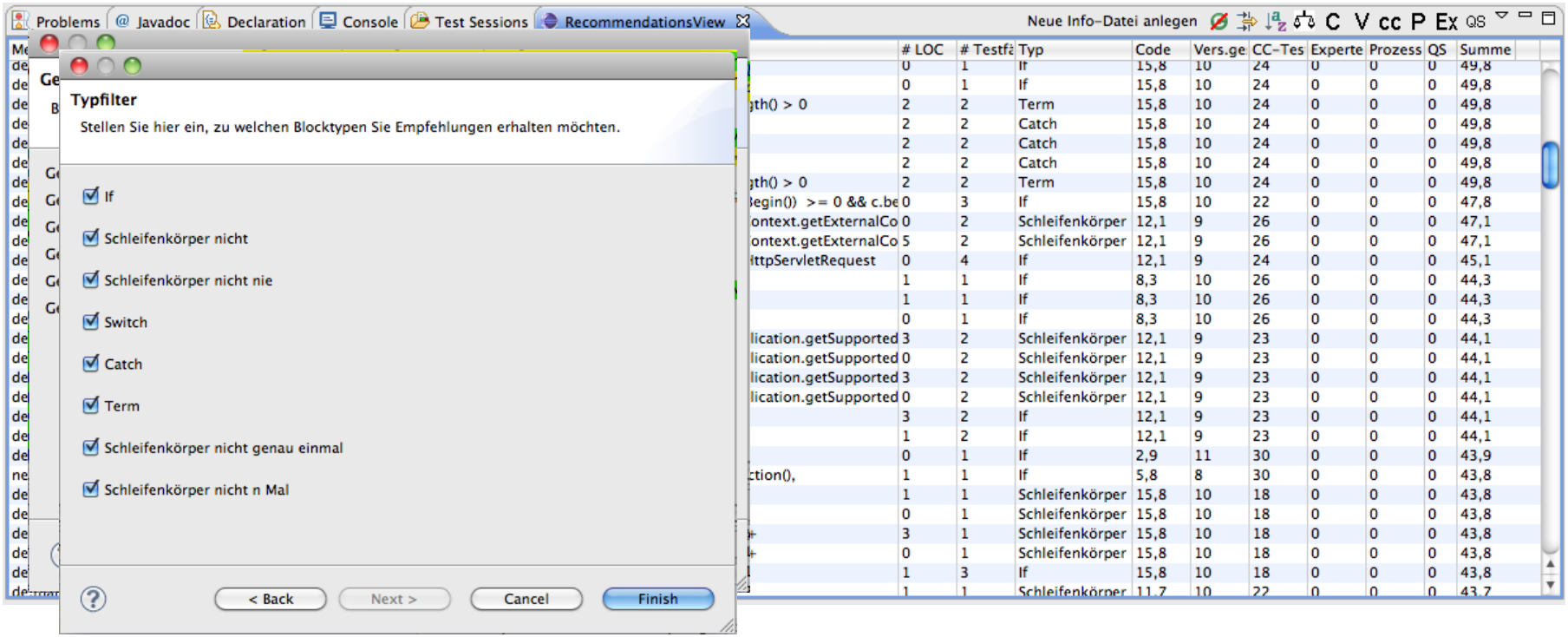
Testsuite-Optimierung mit CodeCover

Übersicht:

- ☑ Test und Glass-Box-Test (GBT)
- ☑ CodeCover „Eckdaten“
- ☑ Testfallherleitung mit CodeCover
- ☑ Spezielle Überdeckungskriterien
- ☑ Getting Started
- ☐ **Weitere Release-Planung**

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Aussichten - Release 1.0.2.0



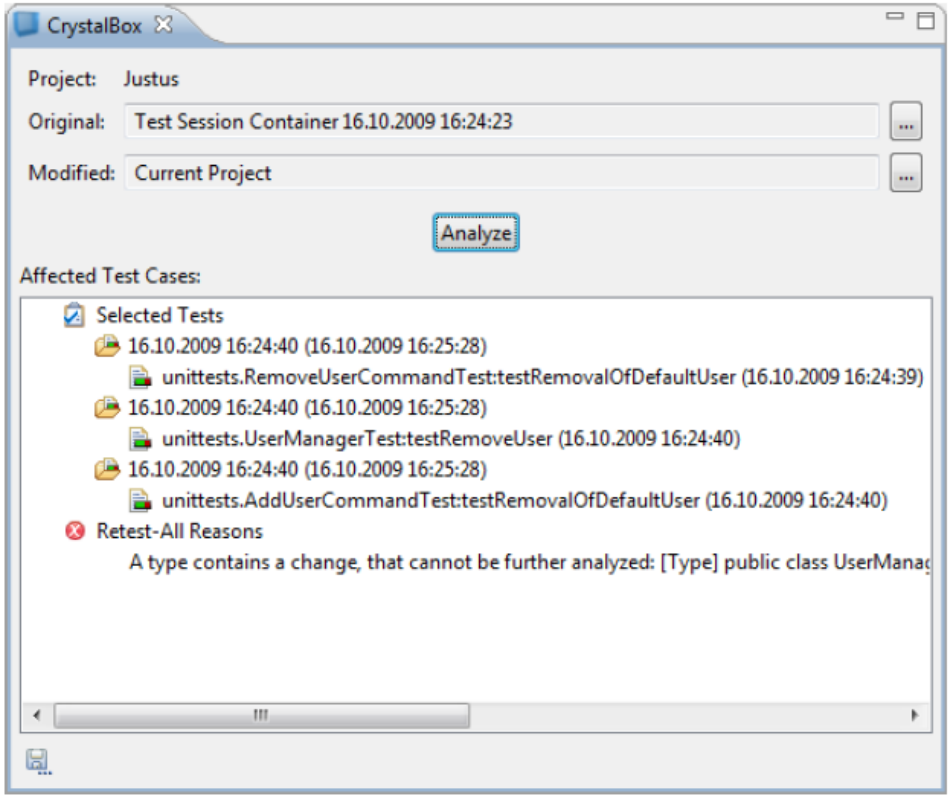
- Testfall-Hinweise nicht mehr als „Excel“-Liste, sondern in CodeCover integriert
- Zur Priorisierung zusätzliche Auswertung von FindBugs und svn (Versionshistorie)

→ Beta ab Mitte August 2011

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Aussichten - Release 1.0.2.1

- Vergleich und „Merge“ des Test Session Containers mit einer veränderten Programmversion
- Unterstützung beim selektiven Regressions-test: Die von der Programmänderung betroffenen Testfälle werden angezeigt



→ Ende 2011

Java Forum Stuttgart 2011 © 2011 Rainer Schmidberger

Zusammenfassung

- Der Glass-Box-Test bietet sich als eine attraktive Ergänzung zu einem gründlichen Funktionstest an, um mit geringem Aufwand neue und gute Testfälle zu entwickeln.
- Mit CodeCover steht hierfür eine entsprechende Werkzeugunterstützung zur Verfügung.

- Nutzen Sie CodeCover!

Vielen Dank für Ihre Aufmerksamkeit

© 2011 Rainer Schmidberger

Rainer.Schmidberger@informatik.uni-stuttgart.de