



G E B I T Solutions

Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

Acceptance-Testen mit Integrity

Dehla Sokenou



Motivation

- Automatisierte Testausführung wichtig, aber wie?
- Programmiert?
 - Beispiel: xUnit
 - Tests sind nicht für Kunden verständlich
- Natürliche Sprache?
 - Anbindung an zu testendes System?
 - Interpretation / Semantik
- Fazit: Irgend etwas dazwischen
 - Formalisierte Sprache, mehr oder weniger an natürlicher Sprache angelehnt
 - Beispiele: FitNesse, Behavior Driven Testing, Integrity

Integrity

- Acceptance-Test-Framework
- Intern bei GEBIT entwickelt
- Seit über 1½ Jahren im produktiven Einsatz
 - Anbindung an eigenes modellbasiertes Entwicklungsframework
- Inzwischen Open Source unter EPL
- Fokus auf
 - IDE-Support: Code Completion, Formatter, References...
 - Lesbarkeit von Testfällen und Ergebnissen
 - Flexibilität

Integrity

- Eclipse-Plugin
 - XText-basierte DSL
 - Definition von
 - Fixture-Klassen (Java)
 - Fixture-Bindings (in Integrity-Dateien)
 - Testsuiten (in Integrity-Dateien)
 - Integrity-Dateien können sowohl Fixture-Bindings als auch Testsuiten enthalten
 - Sinnvoll ist aber die Trennung in verschiedenen Dateien
 - Import von anderen Integrity-Packages möglich
-
- ▶ Ein einfaches Beispiel
 - ▶ Beispiel Parameter

Aufrufdefinitionen und Variablen

- Unterschied zwischen Call-Definition und Test-Definition
 - Tests vergleichen das Resultat mit dem erwarteten Wert
 - Calls führen eine Fixture ohne Vergleich aus und erlauben die Speicherung von Werten in Variablen
 - Variablen müssen deklariert werden
 - Speichern von Werten in Variablen global oder lokal möglich

▶ Beispiel CallDef & Variablen

Tabellen und Objekte

- Tabellen ermöglichen den Test in einem Rutsch, ohne jedes Mal die Fixture aufzurufen
 - Ähnlich FitNesse-Testtabellen
- Komplexe Objekte können definiert werden

▶ Beispiel Tabellen

Weiteres

- Testsuitenorganisation
 - Verlinkungen / Referenzen möglich
 - Abhängigkeiten definierbar
- Custom Proposal Fixtures
 - Um Vorschläge für Parameterwerte zu machen
- Custom Results Fixtures
 - Eigene Testauswertung

Anbindung an modellbasiertes Framework

- Analog zum FitNesse-Bindung mit dem modellbasierten Framework TREND
 - Führt Testbefehle anhand des Modells aus → generische Fixtures
 - Bisher fehlt: Testgenerierung aus dem Modell
- ▶ Beispiel Testsuite modellbasierte Anwendung

Testausführung

- Testausführung in der Console und zusätzlich per Debugger
- Debugger in Eclipse integriert
- Testausführung muss an den Debugger attached werden
- Unterstützt übliche Debugger-Funktionen
 - Sprung in den Testfall
 - Step-Modus
 - Breakpoints

▶ Testausführung im Debugger

Erfahrungen in Projekten

- Integrity seit ca. 1½ Jahren im Einsatz
- Ursprünglich als Ersatz für FitNesse
- Entwickler sind mit Integrity deutlich glücklicher
 - Vermeidung von Tippfehler
 - Vernünftiger Debugger
 - Bessere Lesbarkeit der Testergebnisse
 - Gewohnte Umgebung
- Erfahrungen mit Tests von Kundenseite fehlen noch

Mehr Informationen

- <http://www.integrity-tf.org>
 - Integrity-Frameworks
 - Jenkins-Plugin zur Ergebnisauswertung im automatischen Build-Prozess
 - Tutorials / Dokumentation (im Aufbau)
 - Fixtures für den Test von Swing-Anwendungen (im Aufbau)



G E B I T Solutions

Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

Fragen?

