



Softwarequalität und Test von .NET-Anwendungen

Alexander Dais



Agenda

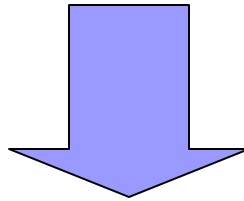
- Qualitätsbegriff
- Konstruktions- und Prüfverfahren zur Erhöhung der Softwarequalität
- XUnit
- Demo

Qualitätsbegriff (1)

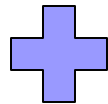
- Wenig funktionale Fehler?
- Rasche Programmausführung?
- Geringer Ressourcenverbrauch?
- Einfache Wartbarkeit?
- Einfache Migrierbarkeit?
- Einfache Erweiterbarkeit?
- Gute Ergonomie?
- Sicherheit?

Qualitätsbegriff (2)

Das Programmverhalten muss möglichst genau den Anforderungen entsprechen

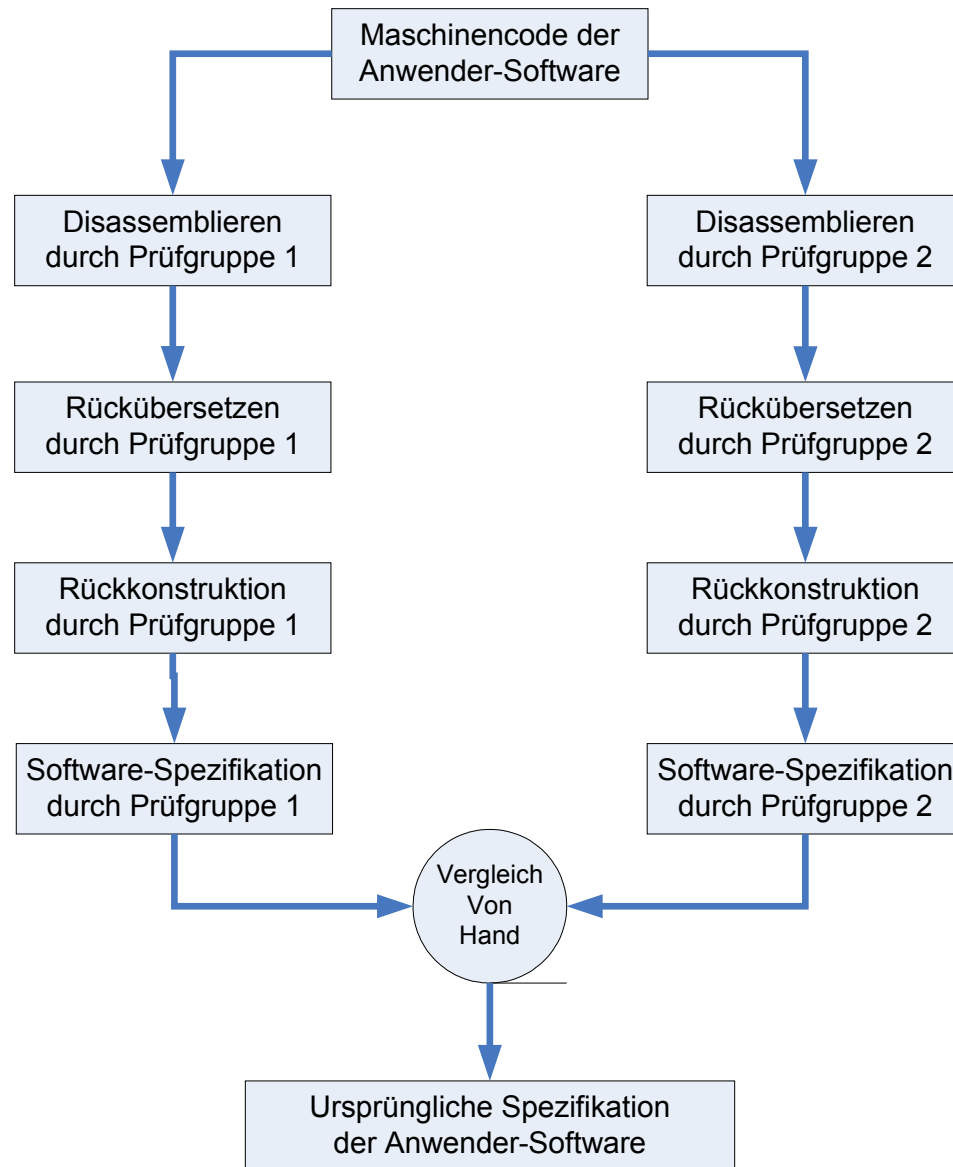


Anforderungen müssen genau beschrieben sein



Überprüfung ob das Programm die Anforderungen erfüllt

Beispiel: Diversitäre Rückwärtsanalyse(1)



Beispiel: Diversitäre Rückwärtsanalyse(2)

Vorteile:

- direkter problembezogener Ansatz
- geeignet für Leute mit unterschiedlichem Ausbildungshintergrund

Nachteile:

- 2 Mannmonate für 4KB Maschinencode
- Erst am Ende der Entwicklung überprüfbar ob das Programm mit den Anforderungen übereinstimmt

Traceability der Anforderungen (1)

Den Ergebnissen der einzelnen Entwicklungsschritte die Anforderungen gegenüberstellen

Beispiel:

Die Variable rechnungsBetrag wird als Integer implementiert, weil der Kunde seine Produkte nur zu vollen Euro-Preisen verkaufen will

Traceability der Anforderungen (2)

The screenshot illustrates the traceability of requirements in Microtools Objectif. It features several overlapping windows:

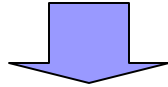
- UML Diagram:** A Use Case diagram showing an actor 'Safari-Inn' connected to a use case 'Unterkunft von externem Anbieter' with the relationship '<<kommuniziert>>'. A context menu is open over the use case, listing options like 'Beschreibung', 'Klassendiagramme', and 'Sequenzdiagramme'.
- Microsoft Word Document:** A document titled 'Dokument in Anwendungsfall Unterkunft von externem Anbieter - Microsoft Word'. It contains a table with the following structure:

Section	Description
Ziel	Welches fachliche Ziel wird mit seiner Durchführung verfolgt? Auf diesen Eintrag kann verzichtet werden, wenn der Name des Anwendungsfalls das Ziel bereits unmissverständlich ausdrückt. Es soll eine Unterkunft aus dem Pool des neuen Partners „Safari-Inn“ für die Etappe reserviert werden. Der Server des Partners erlaubt das Abrufen einer Liste aller angebotenen Unterkünfte. Für jedes Safari-Inn-Camp werden alle buchungsrelevanten Informationen übertragen.
Kontext der Verwendung	In welchem fachlichen Zusammenhang tritt der Anwendungsfall auf? Reservierung einer Reise
Vorbedingungen	In welchem Zustand befindet sich die fachliche Welt vor Beginn des Anwendungsfalls? Es wird davon ausgegangen, dass der hier beschriebene Zustand für den Anwendungsfall immer gegeben ist und nicht innerhalb des Anwendungsfalls abgeprüft wird. Zielort und Datum der Etappe
Nachbedingung im Positivfall	In welchem Zustand befindet sich die fachliche Welt nach erfolgreicher Ausführung des Anwendungsfalls? Vervollständigen Sie die Beschreibung des Zustands. Unterkunft ist beim Partner
Nachbedingung im Negativfall	In welchem Zustand befindet sich die fachliche Welt nach erfolgloser Ausführung des Anwendungsfalls? Vervollständigen Sie die Beschreibung des Zustands. Auch im Fehlerfall, wenn keine Unterkunft reserviert werden kann, muss sichergestellt werden, dass sich das System in einem definierten Zustand befindet. Verwenden Sie dazu auch die Begriffe <<Fehler>>, <<Warnung>>, <<Information>>, <<Fehler>>, dann <<sicher>>. Es ist keine Unterkunft reserviert
- Klasse anlegen (Create Class) Dialog:** A dialog box for creating a new class. The class name is 'Safari-Inn-Camp', the stereotype is 'Class', and the 'Öffentlich' (Public) checkbox is checked.
- objectif - Design View:** A design view window showing a project tree with the following structure:
 - Gui
 - Prototype
 - Tests
 - TourManagement
 - TourManagement
 - Calculate Tour Price
 - Klassen und Typen
 - AdventureTour
 - Camel
 - Camp
 - ITransport
 - Safari-Inn-Camp
 - Stage
 - Importierte Typen
 - TourReservation
 - Architecture
 - Prototype

Konstruktive Arten der Qualitätssicherung

Diversität:

- Verschiedene Implementierungen
- Verschiedene Hardware

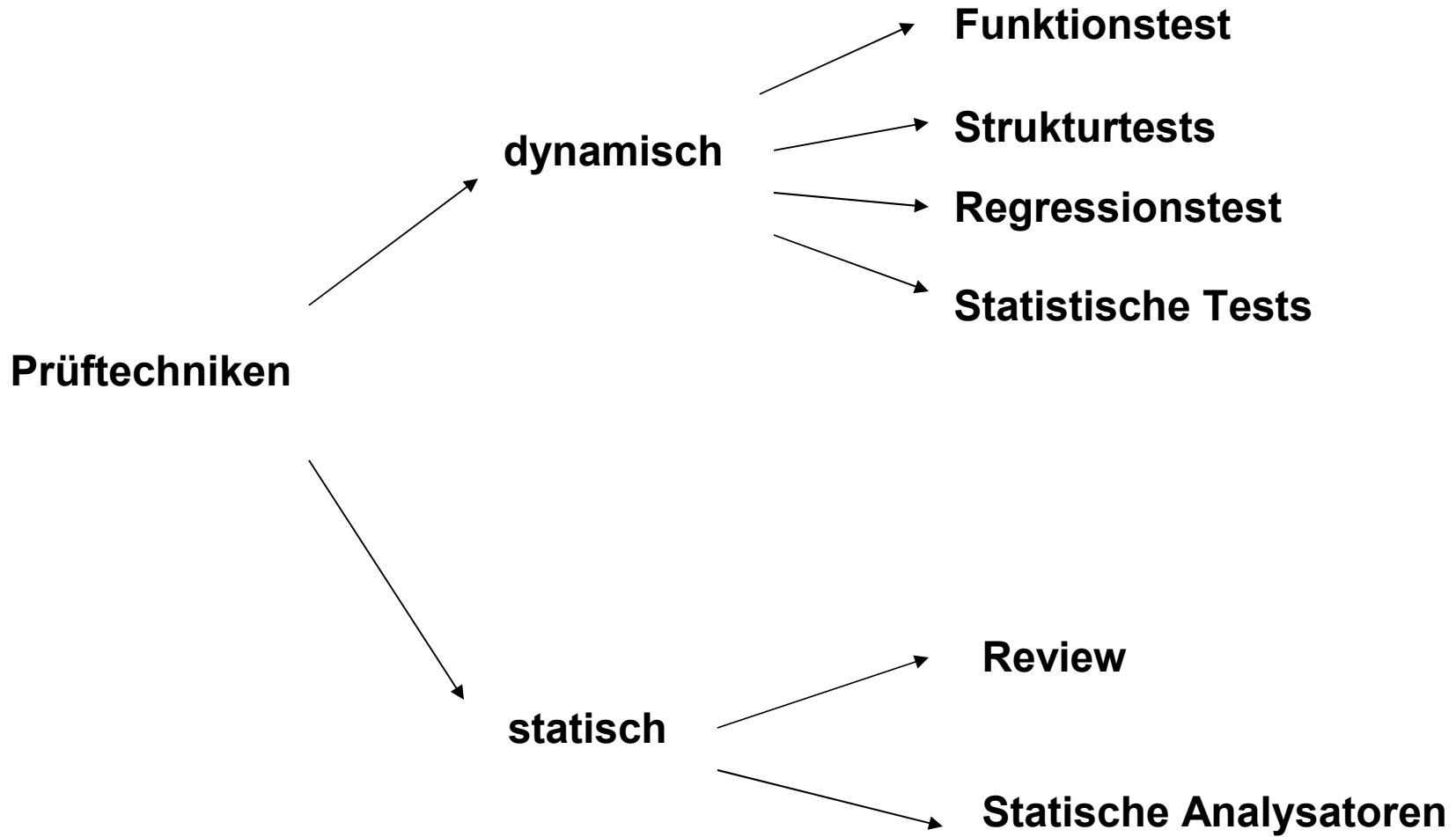


Vergleich der Ergebnisse

Selbstüberwachung:

An bestimmten Stellen im Programm
Werte auf Plausibilität prüfen

Prüftechniken



Funktionstest

- „Black Box Test“
- Untersuchung auf funktionaler Ebene
- Kein Wissen über Implementierung erforderlich

Voraussetzungen:

- Spezifikation
- ausführbarer Code

Strukturtest

- „White Box Test“
- Untersuchung des Kontrollflusses
- Entwicklertest
- Debugger ist ein mögliches Werkzeug

Voraussetzung:

- Quellcode

Regressionstest

- Test werden in einer wiederholbaren Form angefertigt -> Testautomatisierung
- Jede Version der Software durchläuft die einmal gefertigten Tests

Voraussetzungen:

Tests müssen aktuell gehalten werden

z.B.

- an veränderte Schnittstellen angepasst werden
- für neue Softwarefunktionen müssen die bestehenden Tests erweitert werden

Statistische Tests

statistische Versagenswahrscheinlichkeit

Voraussetzungen:

- Testläufe dürfen sich nicht beeinflussen
- Eingangsdatenkombinationen müssen mit der gleichen Wahrscheinlichkeit wie im späteren Betrieb kommen
- Jeder Ausfall muss bemerkt werden

Statistische Tests werden kaum in der Praxis eingesetzt

Review

- Mehraugenprinzip
- Passend für jedes Entwicklungsergebnis:
Anforderungen, Design, Code, Test usw.

Vorgehensweise:

- Planung
- Einführung/Vorbesprechung
- Vorbereitung
- Reviewsitzung
- Nachbereitung

Statische Analysatoren(1)

Statische Prüfung des Quellcodes z.B.:

- Kontrolle der Codierrichtlinien
- Ermittlung von Metriken

Allgemein bekannte Statische Analysefunktionen:

- Compiler
- Syntaxhighlighting

Vorteil:

- Funktioniert „auf Knopfdruck“

Statische Analysatoren(2)

Kostenloses Analysewerkzeug für .NET:

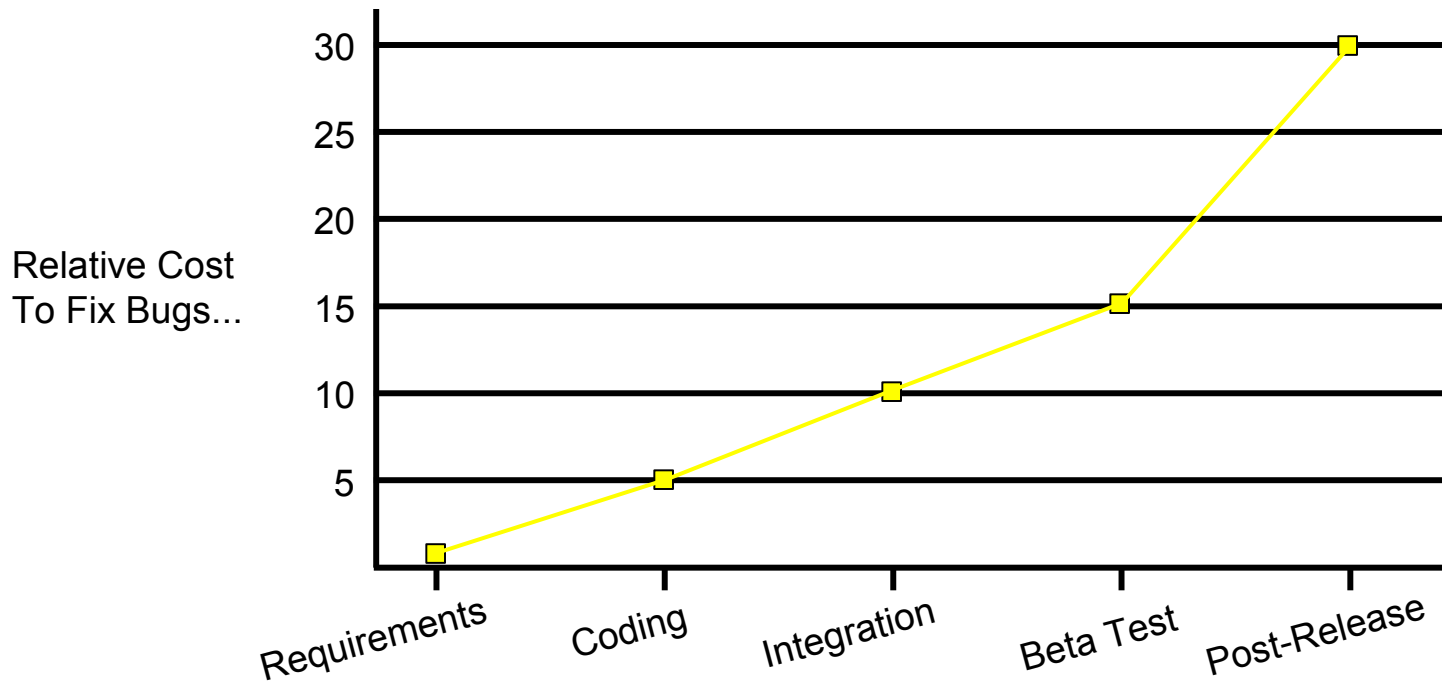
FxCOP:

Überprüft die Einhaltung der
Microsoft Naming Convention

XUnit(1)

- Funktionstest mit Regressionfähigkeit
- „Erfinder“ Kent Beck (SUnit)
- Inzwischen gibt es JUnit, NUnit usw.

XUnit(2)



Grafik aus: www.nist.gov/director/prog-ofc/report02-3.pdf

-> **Frühzeitiges Fehlerfinden reduziert Kosten**

XUnit(3)

- Code a little, test a little
- XUnit ist Teil von XP
 - > Nach jeder Integration alle XUnittests laufen lassen

NUnit

- Framework zum XUnit-Testen mit .NET.
- Nutzt Reflection
- Grafische Oberfläche oder Console
- Opensource

Aufbau NUnittest

- [TestFixture]
- [SetUp]
- [Test]
- [TearDown]

Möglichkeiten im NUnittest

- `Assert.IsTrue`
- `Assert.IsFalse`
- `Assert.AreEqual`
- `Assert.IsNotNull`
- `Assert.IsNull`

Testfälle finden

Zahlen:

- 0
- Grenzbereiche

Strings:

- Leerer String

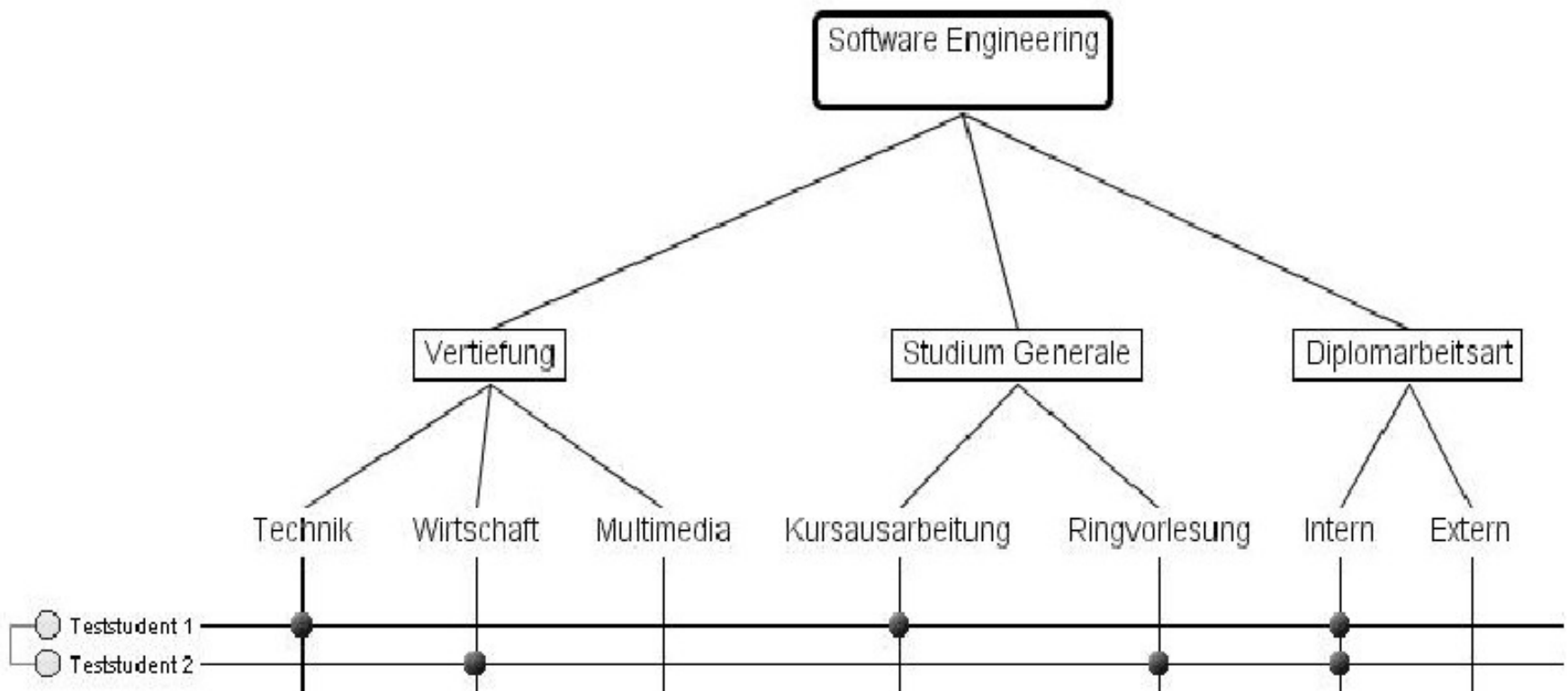
Objekte:

- Null

Collections:

- Leer
- Ein Element
- Maximal

Klassifikationsbaummethode



Grafik: Anwendung der Klassifikationsbaummethode mit dem CTE XL

NUnitForms

Erweiterung von NUnit für GUI Tests

Capture und Replay Tool:

- Nimmt Benutzeraktionen auf
- Speichert diese als NUnit Tests ab



NCover

Ermittelt die Codeabdeckung/Coverage
von NUnit Testfällen

Report in XML:

- Welche Codezeilen ausgeführt wurden
- Wie oft eine Codezeile ausgeführt wurde

Nester

"Why just think your tests are good when you can know for sure? Sometimes Jester tells me my tests are airtight, but sometimes the changes it finds come as a bolt out of the blue. Highly recommended." Kent Beck

Zitat von der Seite des verwandten Tools Jester: <http://jester.sourceforge.net/>

Errorseeding Tool:

Der Sourcecode wird verändert

-> Die Qualität der Tests wird geprüft

Demo

NUnit

NUnitForms

Einsatz des Capture und Replay Tool von NUnitForms:

- Console: `NUnitForms.Recorder.exe`
- Auswahl des zu testenden Assembly
- Benutzerinteraktionen mit der Oberfläche
- Aufgenommener NUnitForms-Test

NCover

Zur Durchführung einer Coveragemessung von NUnit Tests:

- Console: `NCover.Console.exe /c "nunit-console.exe" "[Test].dll"`
- Feststellung der Code-Coverage in `Coverage.Xml`

Links

NUnit:

<http://www.nunit.org/>

NUnitForms:

<http://nunitforms.sourceforge.net/>

NCover

<http://www.ncover.org/>

Nester:

<http://nester.sourceforge.net/>

CTE XL für die Klassifikationsbaummethode

<http://www.systematic-testing.de/>

FxCop:

<http://www.gotdotnet.com/team/fxcop/>



Fragen

?