Testplanung fuer die Implementierung des vis2000–Projektes:

Teststrategie

Folgende Regeln muessen von allen am Test beteiligten eingehalten werden:

Allgemeine Regeln:
1. Jeder, der eine Klasse implementiert, ist fuer den Klassentest zustaendig.
2. Ein Test wird in einem Testprotokoll festgehalten. Wird ein Fehler gefunden, wird zusätzlich ein Fehlerprotokoll ausgefüllt. Dieses wird über einen Verantwortlichen an den jeweiligen Implementator weitergeleitet. Dieser behebt den Fehler und beendet das Fehlerprotokoll.
3. Es duerfen nur getestete Klassen in CVS eingechecked werden, eventuell zusammen mit einem Testprotokoll an noch zu spezifizierender Stelle.
4. Jedes Testprotokoll umfaßt jeweils nur eine Schicht.

Inkrementeller Test:

Das zu testende System ist in 3 Schichten oder Ebenen eingeteilt:
Die oberste Schicht ist die Praesentationsschicht mit Servlets und Java Server Pages (JSPs).
Darunter liegt die Logikschicht mit den Cache–Klassen und den Operationsklassen.
Auf der untersten Schicht liegen die Enterprise JavaBeans, die auf die darunterliegende Datenbank zugreifen.

Es erfolgt ein inkrementeller Test, der in 3 Schritten ablaeuft.
Erst wenn alle Tests eines Schrittes erfolgreich verlaufen sind, duerfen die Test des naechsten Schrittes ausgefuehrt werden.

1. Komponententest:
Im ersten Schritt werden die kleinsten Einheiten, die Klassen, unabhaengig voneinander getestet. Dabei werden alle Methoden getestet.
Es muessen geeignete Datensaetze erschaffen werden, die an die Methoden der zu testenden Klassen uebergeben werden.

1.1 Es werden klassenweise alle Methoden getestet.
Zuerst werden die Klassen der Datenbankschicht (EJBs) getestet, da nur Stellvertreter oberhalb der Datenbankschicht implementiert werden muessen. Diese rufen die komplexen Methoden in den EJBs auf und fangen die von den EJBs gelieferten Rueckgabewerte auf, die sonst an die Klassen der Logikschicht geliefert werden wuerden.Hierbei wird gleichzeitig die Datenbankschicht getestet, indem Daten über die EJBs in die Datenbank geschrieben und dann wieder ausgelesen werden.

1.2. Es folgt der Test der Logikebene. Es werden Stellvertreter implementiert, die die Methodenaufrufe aus der Praesentationsschicht simulieren.
Es werden Stellvertreter in der Datenbankschicht implementiert die anstelle der EJBs die Remote–Aufrufe verarbeiten und Dummy–Werte zurueckgeben anstatt auf die Datenbank zuzugreifen. Die gelieferten Werte werden von den Logik–Klassen verarbeitet und die Ergebnisse getestet.

1.3. Schliesslich folgt der Test der Praesentationsschicht.

Diese Tests sind am schwierigsten, da hier die Benutzerinteraktion simuliert werden muss. Statt Eingaben in HTML–Seiten muessen Stellvertreterklassen geschrieben werden, die die Methoden der Servlets bzw. die Methoden der JSPs aufrufen.
Dazu muessen die richtigen Einstiegspunkte fuer die Klassen gefunden werden, da ein Test, beginnend mit dem Frontservlet einen unnoetigen Zusatzaufwand darstellt.
Es muessen in der Logikschicht Stellvertreter implementiert werden, deren Methoden anstelle der Methoden der Logikklassen aufgerufen werden und die Dummy–Werte zurueckliefern. Diese Werte werden von den Servlets bzw. den JSPs verarbeitet. Die dann erzeugten Werte werden getestet.

2. Integrationstest:
Das Zusammenspiel von Klassen, die sich in verschiedenen Schichten befinden, wird getestet. Imlizit werden hiermit die Schnittstellen der 3 Schichten getestet.
Die einzelnen Aktionen der Anforderungsdefinition werden fuer die Tests verwendet.

3. Systemtest:
Beim Sytemtest wird das das Gesamtsystem, aufbauend auf den Benutzeranforderungen, getestet. Es werden jegliche erdenklichen Tests durchgefuehrt, die auch nicht in der Anforderungsdefinition stehen, um das System auf "Herz und Nieren" zu testen. D.h. vor allem die Fehlbedienung muss getestet werden.

Dokumentation der Tests:
Zu den durchgefuehrten Tests gehoert die genaue Dokumentation folgender Punkte in Anlehnung an die von Natalie vorgestellten Inspektionsergebnisse und Fehlerbehebung:

1.
Tester
Implementator der Klasse
Datum
Schicht
Zu testende Klasse
zu testende Methoden
Testdatensatz
erwartete Ergebnisse
eingetretene Ergebnisse
Soll/Ist – Vergleich

2.
Wenn Fehler aufgetreten sind, muss ein Fehlerprotokoll geschrieben werden:
Ref.–Nr auf Test

Tester
Implementator
Datum
Fehlerklasse:
Unvollstaendigkeit (U), falsche Information (F),
Ueberspezifikation und irrelevante Informationen (R),
Inkonsistenzen und Widersprueche (I),

sonstige Fehler (S)
Fehlerbeschreibung
Vorschlag zur Fehlerbehebung

3.
Wenn der Implementator informiert wurde, versucht er den Fehler zu beheben und vervollständigt das Fehlerprotokoll mit folgenden Daten:

Hilfsmittel und Werkzeuge:
Es wird mit JUnit, einem Testwerkzeug, dass fuer Xtreme Programming entwickelt wurde, eingesetzt. Es ist flexibel genug, um Tests feingranular durchfuehren zu koennen bis zu Attributwerten.
Die graphische Version JUnitX wird in TogetherJ integriert.