

Testkonzept

Als fester Bestandteil des Integrationsprozesses sind folgende Komponenten- und Systemtests vorgesehen.

1. **Methodentests** Testen von einzelnen Methoden
2. **Backdoortesting** Testen von Projektsegmenten
3. **Frontendtests** Testen des gesamten Projektes

Zudem wird in jeder dieser Phasen ein Testplan nach der zu diesem Zeitpunkt vorliegenden Spezifikation erstellt.

1. Methodentests

Vorraussetzung für diese Testphase ist eine enge Zusammenarbeit mit den Entwicklern des Projektes. Hierfür wäre eine Umsetzung mit Hilfe von JUnit möglich. Allerdings ist dies aufgrund des engen Zeitrahmens, der langen Einarbeitungszeit und dem großen Umfang gegebener Möglichkeiten in JUnit nicht sinnvoll und nicht notwendig. Eine einfachere und von uns angestrebte Möglichkeit ist es, eigene Testmethoden zu schreiben. Diese rufen gegebene Projektmethoden mit Parametern auf und vergleichen Rückgabewerte mit vordefinierten Werten.

2. Backdoortesting

Bei dieser Testphase werden an verschiedenen Punkten Testklassen im Servlet implementiert. Diese Testklassen können ähnlich der eigentlichen Projektklassen von einem Client angesprochen werden. Nach dem Aufruf der Testklassen rufen diese mit gesetzten Parametern (ausgelesen aus einer Parameterdatei z.B. XML) andere Klassen und/oder Methoden auf und starten somit weiterführende Prozesse. Nach Beendigung der gestarteten Prozesse werden deren Rückgabewerte abgefangen, ausgelesen und mit vordefinierten Werten verglichen.

2.1 Lasttests

Auf diese Weise können auch Lasttest durchgeführt werden. Dazu müssen die Testklassen insofern erweitert werden, Daten zu empfangen (Parameter und Vergleichswerte). Nach genannter Modifikation könne diese von verschiedenen Clients und/oder Konsolen mit unterschiedlichen Parametern und Vergleichswerten zur selben Zeit initiiert werden.

Zum jetzigen Zeitpunkt ist es noch nicht ersichtlich, ob die Methode des Backdoortestings möglich ist, da die endgültige Designbeschreibung des Projektes noch nicht vorliegt, also das genaue Zusammenspiel der einzelnen Methoden und Funktionalitäten noch nicht ersichtlich ist.

3. Frontendtests

In dieser Testphase wird an bestehenden Benutzeroberflächen teilweise automatisiert getestet. Beim manuellen Testen von Benutzeroberflächen wird eine Prüfung anhand der bestehenden Spezifikation vorgenommen. Zur Realisierung automatisierter Frontendtests existieren mehrere Möglichkeiten, die z.B. durch Rational "Robot" oder Mercury Int. "WinRunner" gegeben sind. Wir haben uns hier, aufgrund gesammelter positiver Erfahrungen, für den Einsatz von WinRunner (TSL) entschieden. Bei der angestrebten Größe dieses Projektes ist es nicht sinnvoll, alle Benutzeroberflächen automatisiert zu testen (dazu sind manuelle Benutzertests

angedacht), sondern nur Tests von Eingabegrenzwerten (z.B. minimale und maximale Längen sowie valide Eingabewerte) der Benutzeroberflächenelemente durchzuführen. Zudem werden die, den jeweiligen Rollen zugewiesenen, Funktionalitäten durch Prüfung der Existenz durch Zugänglichkeit zu den einzelnen Funktionalitäten geprüft.

Die Automatisierung von Frontendtests wird nur dann berücksichtigt, wenn das Spezifizierte Zeitlimit der Testphase dies zulässt.