



Testplan

Hochschule Luzern
Technik & Architektur

Software Komponenten – FS13

Gruppe 03 | Horw, 16.04.2013

Bontekoe Christian | Estermann Michael | Moor Simon | Rohrer Felix

Autoren

Bontekoe Christian	
Studiengang	Informatiker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	

Estermann Michael	
Studiengang	Informatiker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	

Moor Simon	
Studiengang	Informatiker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	

Rohrer Felix	
Studiengang	Informatiker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	

Änderungskontrolle

Version	Datum	Autor	Beschreibung
1.0	15.03.2013	Felix Rohrer	1. Version
1.1	21.03.2013	Felix Rohrer	Testfälle definieren
1.1	16.04.2013	Felix Rohrer	Version 1.1 freigegeben

Inhalt

1	Einführung.....	1
1.1	Kontext	1
1.2	Testphilosophie	1
2	Begriffe, Abkürzungen und Referenzen	1
2.1	Begriffe & Abkürzungen	1
2.2	Referenzen	1
3	Testplanung.....	2
3.1	Unit-Tests	2
3.2	Test First Methode	2
3.3	Integrationstest	2
3.4	Systemtest	2
4	Testspezifikation	3
5	Testfälle.....	3

1 Einführung

1.1 Kontext

Ziel ist es ein „Logger-Programm“ zu erstellen, bei welchem sich der Kunde nicht um Fehler resp. Unregelmässigkeiten herumschlagen muss. Daher ist es wichtig die Software zu testen, um mögliche Fehler bereits frühzeitig zu erkennen.

1.2 Testphilosophie

Die Testphilosophie besteht aus den folgenden Punkten:

- Testen der Komponenten (Unit-Tests)
- Integrationstest
- Systemtests

2 Begriffe, Abkürzungen und Referenzen

2.1 Begriffe & Abkürzungen

Abkürzung	Beschreibung
-----------	--------------

2.2 Referenzen

- keine

3 Testplanung

Unit-Tests werden innerhalb der Entwicklungsumgebung erstellt. Unit-Tests werden jeweils nach Bearbeitung resp. Änderungen im Code getestet. Unit Tests sollen auf Stufe Klasse unabhängig von den anderen Komponenten durchgeführt werden.

Die Entwickler testen das Zusammenspiel der Komponenten zusammen in System- und Integrationstest. Durch Peer Reviews mit dem Team 4 kann eine externe Meinung eingeholt werden, welche Dokumente und Code überprüfen.

Ein Protokoll soll auf Programmier- oder Überlegungsfehler aufmerksam machen.

3.1 Unit-Tests

Zu möglichst jeder Klasse soll ein Unit-Test erstellt werden. Die Tests werden parallel zur Entwicklung erstellt und werden laufend ergänzt und angepasst. Zu einem späteren Zeitpunkt werden die erfolgreich getesteten Softwareteile zusammengefügt und mit einem Integrations- und Systemtest auf Fehler überprüft.

3.2 Test First Methode

Die geplanten Tests sollen im Rahmen dieses Projektes nach der Test First Methode durchgeführt werden. Diese beschreibt, dass die Testfälle vor der Implementation geschrieben werden sollen. Indem man sich zuerst die Testfälle wie z.B. Spezialfälle oder Extremwerte überlegt, kann die Implementation des Programmcodes bereits bei der Produktion optimiert werden. Der Programmcode reift durch dieses Verfahren förmlich heran.

3.3 Integrationstest

Beim Integrationstest ist es wichtig, dass die einzelnen Funktionen und Komponenten bereits im Voraus mit Unit Tests geprüft wurden. Denn beim Integrationstest werden die einzelnen Komponenten zusammengeführt und auf das korrekte Zusammenspiel geprüft. Die Integrationstests prüfen somit die von Beginn weg definierten Schnittstellen und deren Kompatibilität.

3.4 Systemtest

Wenn das Testen des Zusammenspiels der Teilkomponenten erfolgreich war, folgt schlussendlich der Systemtest. Beim Systemtest wird das System als Ganzes zusammengeführt und getestet. Beim Systemtest wird nach der Spezifikation getestet, somit kann die Korrektheit der Funktionen sichergestellt werden.

4 Testspezifikation

Folgende Komponenten werden getestet:

- Demo Applikation (inkl. Logger Komponente)
- Log Viewer
- Log Server

5 Testfälle

# 1 Demo Applikation starten	
Beschreibung	Die Demo Applikation kann erfolgreich gestartet werden
Akteure	Anwender
Vorbedingungen	Die Applikation muss zur Verfügung stehen
Ergebnis	Die Applikation wird erfolgreich gestartet.
Ergebnis im Fehlerfall	Fehlermeldung
Ablauf	1. Programm starten

# 2 Logger Komponente kann konfiguriert werden	
Beschreibung	Demo Applikation kann Logger Komponente konfigurieren
Akteure	Anwender
Vorbedingungen	Test-Case #1
Ergebnis	Die Konfigurationen können gesetzt werden und werden von der Logger Komponenten entsprechend angewendet.
Ergebnis im Fehlerfall	Konfigurationen werden nicht übernommen
Ablauf	1. Demo App starten 2. Konfiguration anpassen 3. Überprüfen, ob neuen Konfigurationen angewendet werden

# 3 Message an Server übermitteln	
Beschreibung	Demo Applikation kann Log Messages an den Server übermitteln
Akteure	Anwender
Vorbedingungen	Test-Case #2
Ergebnis	Die Log Message können auf dem Logger Server empfangen und eingesehen werden
Ergebnis im Fehlerfall	Die Log Message wird nicht und/oder falsch übermittelt
Ablauf	1. Demo App starten 2. Message generieren 3. Log Eingang auf dem Server überprüfen

# 4 Logger Server speichert Message in Datei	
Beschreibung	Logger Server speichert die empfangene Message in einer Datei
Akteure	Anwender
Vorbedingungen	Test-Case #3
Ergebnis	Die empfangene Message werden vom Logger Server in einer definierten Datei gespeichert. Die Datei kann betrachtet werden.
Ergebnis im Fehlerfall	Message wird nicht / falsch in der (falschen) Datei gespeichert.
Ablauf	<ol style="list-style-type: none">1. Demo App starten2. Log Message generieren3. Meldungseingang auf Server sicherstellen4. Speicherung der Message in der Datei überprüfen

# 5 Server unterstützt mehrere Clients gleichzeitig	
Beschreibung	Logger Server kann mehrere Clients, Messages gleichzeitig verarbeiten
Akteure	Anwender
Vorbedingungen	Test-Case #4
Ergebnis	Mehrere Clients können gleichzeitig verbinden und der Logger Server speichert alle Messages in der Datei
Ergebnis im Fehlerfall	Verbindungsfehler, nicht alle Messages werden gespeichert
Ablauf	<ol style="list-style-type: none">1. Demo App mehrfach starten / auf unterschiedlichen Clients starten2. Log Messages generieren3. Meldungseingang auf Server überprüfen4. Speicherung in Datei überprüfen

# 6 Viewer kann Messages von Server anzeigen	
Beschreibung	Logger Viewer kann Messages von Server darstellen
Akteure	Anwender
Vorbedingungen	Logger Server muss funktionieren, Log Einträge müssen vorhanden sein
Ergebnis	Generierte Messages können im Logger Viewer angezeigt werden
Ergebnis im Fehlerfall	Log Messages werden im Viewer nicht dargestellt.
Ablauf	<ol style="list-style-type: none">1. Log Viewer starten2. Empfang der Log Messages überprüfen