

## Vorgehensweise JUNIT-Test SWE7.1 LosC3 VRZ3 – RDS/TMC

### Überblick:

Die JUnit-Tests für die SWE RDS/TMC laufen folgendermaßen ab:

1. Der Testgenerator wird gestartet (als Thread innerhalb JUnit). Dieser speist folgende Testdaten ein
  - Belegung der Störfallindikatoren
  - Parameteränderungen
  - Erzeugen dynamischer Stau und- Baustellenobjekte,gemäss den Vorgaben – auch zeitlich – der Prüfspezifikation.
2. Danach wird die Applikation RDS/TMC als Tochterprozess von JUNIT automatisch gestartet
3. JUnit meldet sich auf neu erzeugte und vorhandene dynamische Objekte vom Typ RDSMeldung an und wertet die erzeugten Daten gemäss Prüfspezifikation mit Assertions aus.
4. Optional kann ein Protokollierer manuell gestartet werden
5. Am Ende wird auch die Applikation RDS/TMC automatisch beendet

Für jeden der beiden Prüffälle 1 und 2 der Prüfspezifikation gibt es je eine JUnit-Testsuite – *Prueffall1\_Suite* und *Prueffall2\_Suite*, mit insgesamt 20 Unterfällen.

Dadurch werden die Prüfungen der Prüfspezifikation vollständig in automatisierte Tests abgebildet.

Die in der Prüfspezifikation spezifizierte Prüfstrecke wird in der (im mitgelieferten Kernsystem enthaltenen) Dambach-Teststrecke exakt auf den ersten 5 Straßensegmenten abgebildet (verkehrsModellNetz.testBW, außerdem wird die LandesMeldeStelle bwLandesMeldeStelle verwendet).

### Ausführen der JUnit-Tests:

Voraussetzungen

1. Das mitgelieferte Test-Kernsystem ist lokal auf dem Rechner zu installieren, auf dem die Tests stattfinden sollen
2. Die DaV-Verbindung der Testsoftware erfolgt mit den Standardeinstellungen 127.0.0.1:8083, die DaV-seitig nicht geändert werden dürfen
3. Für die Authentifizierung erwartet die Testsoftware eine Datei passwd in ihrem Arbeitsverzeichnis (als gültige Authentifizierung-Datei) . Hierfür eignet sich die Datei passwd, die Bestandteil des skript...-Verzeichnisses des Kernsystems ist. Erfolgt der Aufruf der Testsoftware nicht aus dem Skriptverzeichnis, so muss die passwd-Datei entsprechend kopiert werden, dies gilt insbesondere, wenn die Tests aus der Eclipse-Entwicklungsumgebung gestartet werden: passwd wird in diesem Falle ins Eclipse-Projektverzeichnis kopiert.
4. Im entsprechenden Konfigurationsobjekt vom Typ Parametrierung (AOE) müssen in der Attributgruppe Parametrierung folgende Attributgruppen als parametrierend angegeben sein: RDSModell, RDSMeldung, RDSMeldungGeneriert, RDSMeldungSenden, RDSMeldungBearbeitet.
5. Zu Beginn der Tests müssen die Parameter der ATG *RDSModell* exakt so versorgt werden, wie in der Prüfspezifikation vorgegeben. Außerdem dürfen noch keine dynamischen Objekte vom Typ *RDSMeldung* existieren, und es darf keine Instanz der SWE RDS/TMC im System laufen.

Diese Voraussetzungen gelten zum Ausführen von *Prueffall1\_Suite*.

*Prueffall2\_Suite* kann nur ausgeführt werden, wenn unmittelbar vorher *Prueffall1\_Suite* erfolgreich ausgeführt wurde.

Die Tests können innerhalb der Eclipse-Entwicklungsumgebung ablaufen, dazu sind die gelieferten Test-Quellen als Eclipse-Projekt einzurichten.

Um die Tests außerhalb der Eclipse-Umgebung ablaufen zu lassen, ist die Startklasse org.junit.runner.JUnitCore mit der Testklasse (Testsuite) *de.bsvrz.ste.rdstmc.junit.Prueffall1\_Suite* oder *de.bsvrz.ste.rdstmc.junit.Prueffall2\_Suite* anzugeben. Neben den Standard jre-Java-Bibliotheken und den Standard-Distributionspaketen müssen sich in jedem Fall im classpath befinden:

- de.bsvrz.ste.rdstmc-test.jar
- de.bsvrz.ste.rdstmc.jar

- de.bsvrz.sys.funclib.dambach.jar
- junit-4.4.jar

#### Beispiel für Aufruf-Skript (Windows) für die JUnit-Tests – Prüffall1\_Suite

```
@echo off
if x%JAVA_HOME%x == xx ( set java=java) else set java=%JAVA_HOME%\bin\java

title RDSTMC JUnit Test Prüffall1

%java% ^
-classpath ^
..\..\de.bsvrz.ste.rdstmc\de.bsvrz.ste.rdstmc-runtime.jar;^
..\..\de.bsvrz.ste.rdstmc\test\de.bsvrz.ste.rdstmc-test.jar;^
..\..\de.bsvrz.ste.rdstmc\test\junit-4.4.jar;^
-Xmx300m ^
org.junit.runner.JUnitCore ^
de.bsvrz.ste.rdstmc.junit.Prueffall1_Suite

echo errorlevel %errorlevel%

rem Fenster nicht sofort wieder schließen, damit eventuelle Fehler noch lesbar sind.
pause
```