



Bachelorprojekt //TODO Buchstabe I

Verbesserung der Entwurfs- und Testmethodik in verkehrstechnischen, sicherheitskritischen Systemen

Idee

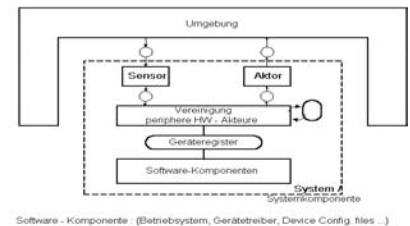
Für die Entwicklung von Kleinssoftware ist kein systematischer Entwurf notwendig. Allenfalls werden Skizzen gemacht, im Übrigen wird direkt programmiert. Größere Systeme dagegen brauchen einen systematischen, strukturierten Aufbau, der nur durch ein sorgfältig erstelltes Lösungskonzept sichergestellt werden kann.

Ferner legt ein gutes Lösungskonzept den Grundstein für ein leicht wartbares und testbares System. In der Konzipierung begangene Fehler sind häufig gar nicht oder mit nur großem Aufwand reparierbar. Sorgfältiges Konzipieren ist daher für jede Art von professionell entwickelter Software wirtschaftlich.

Die Softwareentwicklung ist mit der Einführung objektorientierter Methoden sowie der zunehmenden Verbreitung integrierter Softwareentwicklungsumgebungen mit der Möglichkeit zur Codegenerierung nachhaltig verändert worden. Besonders in sicherheitskritischen Systemen führen die objektorientierten Konzepte der Polymorphie und des dynamischen Bindens zu Problemen in der Qualitätssicherung. Diese Probleme können durch eine automaten- und aktivitätsbasierte Modellierung ausgeschlossen werden. Dieser Ansatz ist besonders für sicherheitskritische Systeme naheliegend, da diese Systeme über Sensoren und Aktoren gesteuert werden. Allerdings fehlen Werkzeuge um ein Lösungskonzept von der Analysephase bis zur Systemtestphase umzusetzen.

Aufgabe

Ziel des Projektes ist die Erweiterung eines bestehenden Werkzeugs für die Softwareentwicklung. Die einzelnen Teilkomponenten dieses Werkzeugs sollen die notwendigen analytischen und konstruktiven Qualitätssicherungsmaßnahmen für die jeweilige Entwicklungsphase ermöglichen. Die Leistungsfähigkeit des Werkzeuges soll dabei anhand eines sich gerade in der Entwicklung befindenden verkehrstechnischen Systems demonstriert werden.



**Betreuer bei der Siemens AG:**

Siemens AG, Corporate Technology
Jörg Gericke, CT PP 2

Betreuung:

Die Koordination wird von Seiten des Fachgebiets Softwaretechnik durch Herrn Prof. Dr. Peter Liggesmeyer erfolgen.
Die inhaltliche Betreuung wird von Dipl.-Inf. J. Gericke übernommen.

Vorgehensweise und Rahmenbedingungen:

Das Projektteam soll einen Eindruck gewinnen, wie ein Projekt unter industriellen Rahmenbedingungen abgewickelt wird. Daher wird jeweils wöchentlich ein Projektmitglied die Aufgabe eines Projektmanagers übernehmen und einen erstellten Projektplan weiterentwickeln.

Weiterhin sollen alle Phasen der Softwareentwicklung im Rahmen der Projektabwicklung durchgeführt werden. Dies erfordert einen strukturierten und zielgerichteten Entwicklungsprozess, welcher mit der Erstellung der Anforderungsspezifikation durch das Projektteam beginnt. Nach der Anforderungsspezifikation wird eine Architektur- und Entwurfsspezifikation entwickelt, welche anschließend objektorientiert implementiert wird. Für die analytische Qualitätssicherung sind darüber hinaus Modul-, Integrations- und Systemtests erforderlich.

Zu den Rahmenbedingungen des Projektes zählt zusätzlich die Verwendung von modernen und industriell verbreiteten Werkzeugen der Softwaretechnik. Die folgenden Werkzeuge sollen dabei zum Einsatz kommen::

Entwurfsspezifikation	Rational Rose und Rational Rose RT
Implementierung	Visual Studio 7.0, .Net Framework, C#

Durch die Einbindung des Projektes in eine industrielle Softwareentwicklung kann es erforderlich sein, dass ein Teil der Projektarbeit direkt vor Ort beim Industriepartner durchgeführt werden muss.

Ihre Vorteile:

- Klarer Funktionsumfang und eindeutige Zielvorgaben
- Nutzung moderner Softwareentwicklungstechniken
- Interessantes Aufgabengebiet, anspruchsvolle Aufgabe und interessanter Industriepartner
- Durchführung des Projektes unter industriellen Rahmenbedingungen
- Wissenserwerb in Spezialthemen der Softwaretechnik, wie z. B.:
 - Anforderungsverfolgung
 - Moderne Softwareentwicklungsumgebungen
 - Objektorientierte Methoden
 - Softwareerstellungsprozesse
 - Konstruktive und analytische Qualitätssicherung
 - UML 2.0
 - Sicherheitskritische Systementwicklung