Project Software Engineering for Computational Science Software SECOSS II

Project Description

Supervisor: Prof. Dr. Barbara Paech
Co-Supervisor: Prof. Dr. Peter Bastian or Prof. Dr. Guido Kanschat (depending on the actual work of the PhD student)

Project Topic:
In the following Computational Science Software is also called Scientific Software.

Scientific Software Engineering has to handle the following challenges:

- The context of scientific software is usually very complex. Only scientists familiar with the scientific domain in question have the ability to entirely understand the software.
- Solving complex scientific problems with scientific software often requires special resources like high performance computing. At the same time, a special programming paradigm like the use of parallel computing is applied.
- Scientific software is often *multi-purpose*, that means it is provided as a framework of different algorithms for solving a class of problems (e.g. DEAL II, DUNE and HIFLOW3 in Heidelberg). This variety has to be handled efficiently during software engineering.
- At the beginning of a scientific software project, the known requirements are often the laws of nature, or stem from mathematics. In most cases, further requirements for the software have not been defined in advance, but emerge during software development.
- In scientific software development, the priority of non-functional requirements is high compared to functional requirements. Typical goals are correctness, performance, portability and maintainability.
- Scientific software is used for gaining research results or solving problems that cannot be solved by other means. The outcome is therefore often not known in advance. This is a problem for testing, since most testing techniques in software engineering assume accurate test oracles.
- The fact that the developers of scientific software mostly are domain scientists and not software engineers has to be considered in the development process. There is also a difference in the objective: a software engineer's goal is to produce high quality software, whereas the goal of a scientist is to produce high quality science. It is important to keep the process easy to learn and quick to adopt.
- The development is distributed and many developers are doctorate students or postdocs who only stay in the team for a few years. Thus, efficient communication and documentation of major decisions is important, but the development process needs to be lightweight.

In previous work in the project SECOSS we have developed a quality assurance process for scientific software [1] and in particular adapted product line testing for the scientific framework DUNE [2,3,4] (recently also with the tool FeatureIDE http://wwwiti.cs.uni-magdeburg.de/iti_db/research/featureide/).

In this project the *quality assurance process* developed for DUNE [5] should be extended for the software framework DEALII [6] and in addition *knowledge management methods* such as rationale management shall be introduced in the development of these frameworks to cope with the distributed development. In particular, the project will focus on supporting decision processes and decision documentation *during re-structuring of the framework*.

The following questions should be answered

- For knowledge management:
  - What are typical decisions made during the development of scientific frameworks? Which decisions need to be documented?
  - How to support distributed decision making?
  - How to support decision documentation as a lightweight addition to the development process? How to enhance issue tracking for that?
- For quality assurance
  - How does the quality assurance process (in particular the variability modeling used for system testing) for DUNE fit to DEALII? What needs to be adapted?
  - How can the developed variability model and system test cases be used to support re-structuring decisions and quality assurance during re-structuring?
  - How can regression testing (that means the selection of test cases) be efficiently adapted for specific changes involved in re-structuring?
  - How can integration/interface testing be supported for scientific frameworks and be used to support quality assurance during re-structuring?
- For both (as a particular form of documentation important for both re-structuring decisions and quality-assurance)
  - How can the variability model developed for testing be used to document the features of the scientific framework?
  - How can such a model be continuously extended for new features?
  - How can such a model support re-structuring?

The questions will be answered by a *technical action research* approach [7] where the development of the two scientific frameworks is studied empirically and the PhD student works with the teams to develop an appropriate software engineering method. This method is introduced to the development processes and evaluated. One major goal is also to understand and describe the differences between the two different frameworks and development processes and to develop a solution which can be adapted to both.

[1] Remmel H, Paech B, Engwer C, Bastian P (2013) Design and rationale of a quality assurance approach for a scientific framework, 5th international workshop on Software engineering for computational science and engineering (SECSE '13), ACM, pp. 58-67.
[2] Remmel H, Paech B, Engwer C, Bastian P (2014) A Case Study on a Quality Assurance Process for a Scientific Framework, Computing in Science and Engineering Vol. 16, No. 3, May 2014, pp. 58-66, IEEE 2014
[3] Remmel H, Paech B, Bastian P, Engwer C (2012) System Testing a Scientific Framework using a Regression-Test Environment, Computing in Science and Engineering Vol. 14, No. 2, March 2012, pp. 38-45, IEEE 2012
[4] Remmel H, Paech B, Engwer C, Bastian P (2011) Supporting the testing of scientific frameworks with software product line engineering: a proposed approach, 4th international workshop on Software engineering for computational science and engineering (SECSE '11), ACM, pp. 10-18.
[5] http://www.dune-project.org/
[6] http://www.dealii.org/
[7] Wieringa R, Morali A (2012) Technical action research as a validation method in information systems design science. Design Science Research in Information Systems, LNCS 7286, pp. 220–238.

Requirements for the PhD student: The student should have very good knowledge of Computer Science, in particular Software Engineering. Experience with developing complex software in a distributed setting would be very beneficial. The student should also be interested in working at the interface between Computer Science and Applied Mathematics. Knowledge in numerical methods for the solution of partial differential equations is of advantage.

Begründung

Innovation: Die Innovation dieses Projektes liegt in der Anwendung von Software Engineering Methoden auf komplexe Software für wissenschaftliches Rechnen. Sie baut auf dem Projekt SECOSS auf, das von Hanna Remmel (auch Doktorandin an der HGS) durchgeführt wurde. http://hgs.iwr.uni-heidelberg.de/ipp/index.php?id=60 und http://se.ifi.uni-heidelberg.de/research/projects/secoss.html
Frau Remmel hat sich bisher auf Qualitätssicherung für Scientific Frameworks konzentrierte und dabei insbesondere Produktlinientest-Methoden und –Werkzeug für DUNE entwickelt und erfolgreich auf dem einschlägig veröffentlicht.
Diese Skizze beschreibt innovative Folgearbeiten im Bereich Qualitätssicherung und Wissensmanagement/Verteiltes Arbeiten. Das konkrete Thema sollte je nach KandidatIn gewählt werden. *Das IWR hat im Bereich Scientific Frameworks mit DEAL II, DUNE und HIFLOW3 ein Alleinstellungsmerkmal. Dies sollte ausgebaut und nachhaltig werden durch Entwicklung und Verwendung darauf angepasster Software Engineering Methoden. Die Innovation liegt also in der Verwendung nachhaltiger Methoden.*

Interdisziplinarität: In dem Projekt SSEDM arbeiten Informatik und Mathematik zusammen. Auch wenn Mathematik und Informatik viel gemeinsam haben, so ist das ingenieursmäßige Vorgehen im Software Engineering sehr unterschiedlich von der Entwicklung effizienter mathematischer Algorithmen. DoktorandInnen im Bereich wissenschaftliches Rechnen konzentrieren sich auf die Mathematik und erstellen Software nur soweit nötig. Fortgeschrittene Methoden der Qualitätssicherung und des Wissensmanagement sind kaum bekannt und werden nicht verwendet. Umgekehrt fehlen den meisten DoktorandInnen im Bereich Software Engineering die Mathematikkenntnisse, um den Anwendungsbereich der Scientific Software zu verstehen. Frau Remmel war in dieser Hinsicht ein Glücksfall, weil sie sowohl Mathematik (auf Lehramt) als auch Informatik studiert hat und mehrere Jahre in der Industrie Software entwickelt hat. *Da es so schwierig ist, geeignete KandidatInnen mit diesen Kenntnissen zu finden, kann das Projekt besonders profitieren von einer internationalen Ausschreibung im Rahmen der HGS.*

Internationalität: In dem Projekt arbeiten 3 Gruppen des IWR zusammen. Es bietet sich an, zu dem Thema am IWR einen Workshop zu organisieren. Frau Remmel hat durch ihre Teilnahme an SECSE gute internationale Kontakte entwickelt, so dass entsprechende ForscherInnen für Lehraufträge und Vorträge gewonnen werden könnten. Entsprechende Vorschläge (siehe unten) wurden früher an die HGS gemacht, aber noch nicht umgesetzt. Eine vertiefte Kooperation oder Ko-Betreuung scheint nicht realistisch, da schon die Interessen mehrere IWR-Gruppen vereint werden sollen.
Dr. Roscoe Barlett arbeitet als Software Engineering Leiter in einem Forschungslabor (Oak Ridge National Laboratories). Er hat langjährige Erfahrung in der Entwicklung eines großen wissenschaftlichen Softwareprojekt "Trilinos". http://www.ornl.gov/~8vt/index.html Er ist nicht regelmäßig in Lehre involviert. Ein Kurs von ihm wäre eher ein Fortgeschrittenen-Kurs zur Programmierung.
Prof. Jeffrey Carver ist Associate Professor an der Universität von Alabama in den USA. (http://carver.cs.ua.edu/). Er organisiert seit Jahren den internationalen Workshop für Software Engineering für wissenschaftliches Rechnen (SECSE). In seiner Forschung hat er mehrere große Fallstudien im Bereich Software Engineering für wissenschaftliche Software durchgeführt. Ein Vortrag von ihm über seine Erfahrungen in unterschiedlichen wissenschaftlichen Softwareprojekten würde einen breiten Überblick zu den aktuellen Themen in dem Forschungsbereich geben.
Dr. Diane Kelly ist Associate Professor an der Royal Military College of Canada. http://www.rmc.ca/aca/mcs-mi/per/kelly-d-eng.asp Unter "Experience and Research Interests" sind ihre Schwerpunkte beschrieben. Die Ausrichtung auf Testen passt besonders gut zur aktuellen Forschung in SECOSS. Sie hat viel Erfahrung in der Lehre von Scientific Software Engineering. Ein Kurs von ihr wäre besonders als Einstiegskurs in das Thema geeignet.