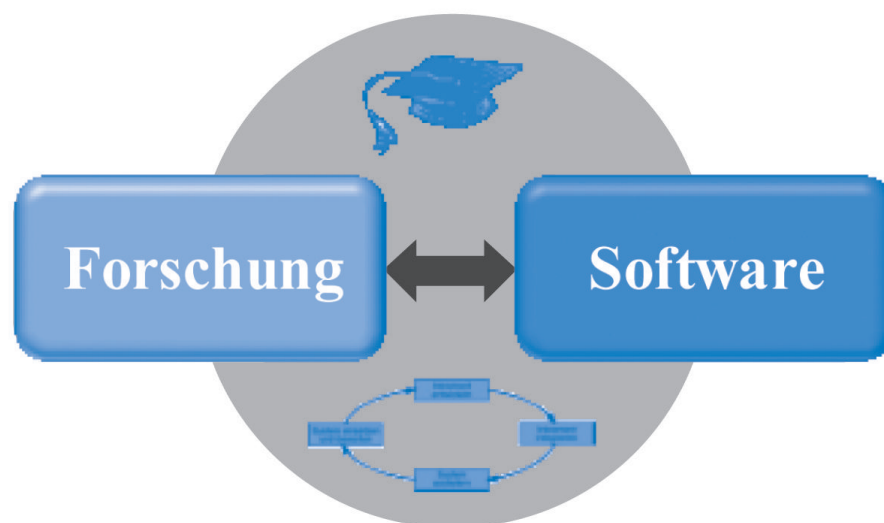


Entwicklung und Evolution von Forschungssoftware

Tagungsband

Rolduc, 10.-11. November 2011



Aachener Informatik-Berichte,
Software Engineering

Band 14

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe
Prof. Dr. rer. nat. Horst Lichter

Inhaltsverzeichnis

Einleitung und Vorwort <i>Horst Lichter</i>	1
--	---

Teil I: Projektbeschreibungen

Das HTR Code Package (HCP) <i>Sarah Scholthaus, Daniela Lambertz</i>	7
Development of libALF <i>Daniel Neider</i>	13
MeDIC - Eine Infrastruktur zum Verwalten, Dokumentieren und Visualisieren von Metriken <i>Matthias Vianden</i>	19
Score-P <i>Christian Rössel, Bernd Mohr, Felix Wolf</i>	23
OpenFlipper – An Open Source Geometry Processing and Rendering Framework <i>Jan Möbius, Leif Kobbelt</i>	31
Der Energie Navigator <i>Thomas Kurpick, Claas Pinkernell, Bernhard Rumpe</i>	37
Scalasca <i>David Böhme, Marc-André Hermanns und Felix Wolf</i>	43
AProVE – Automated Program Verification Environment <i>Carsten Otto</i>	49
SimWeld - 10 Jahre Schweißprozesssimulation am ISF <i>U. Reisgen, M. Schleser, O. Mokrov, A. Schmidt</i>	55
Die Entwicklung des Virtual Reality Toolkit ViSTA <i>Dominik Rausch, Bernd Hentschel und Torsten Kuhlen</i>	63
Entwicklung für das Softwarewerkzeug »NCProfiler« <i>Mario Pothen, Meysam Minoufekr, Lothar Glasmacher</i>	71

Teil II: Ablauf des Workshops – Ergebnisse und Ausblick

Ablauf des Workshop <i>Matthias Vianden</i>	79
Ergebnisse der Arbeitsgruppen und Ausblick <i>Andreas Ganser, Veit Hoffmann, Horst Lichter, Matthias Vianden</i>	81

Einleitung und Vorwort

Horst Lichter

Lehr- und Forschungsgebiet Softwarekonstruktion
RWTH Aachen University
lichter@swc.rwth-aachen.de

1 Motivation für diesen Workshop

In vielen Forschungsprojekten werden Softwaresysteme erstellt, die gewisse Aufgabenstellungen im Kontext der Forschungsziele bearbeiten. Die Entwicklung dieser Softwaresysteme ist geprägt durch verschiedene Faktoren. So arbeiten typisch viele verschiedene Personen teils eher kurz an der Entwicklung der Software, typisch fehlt auch eine stabile und nutzbare Dokumentation. Während manche Forschungssoftware nur kurzfristig benutzt wird und dann nicht mehr gebraucht wird, gibt es viele Beispiele für Forschungssoftware, die sich über Jahre zu einer wichtigen Forschungsplattform entwickelt haben, bzw. die sogar als Produkte vertrieben werden.

Die forschungsgetriebene Entwicklung von Software muss aufgrund der Randbedingungen und dieser speziellen Probleme offensichtlich grundsätzlich anders organisiert werden, als ein klassisches Auftragsentwicklungsprojekt.

Da die Investitionen in Forschungssoftware sehr erheblich sind (auch, wenn diese selten gemessen und bekannt sind) und Forschungssoftware einen wichtigen Beitrag zu unseren Forschungsergebnissen liefert, entstand die Idee, einen Workshop zum Thema „Entwicklung und Evolution von Forschungssoftware“ (EEFSW) zu organisieren und durchzuführen.

2 Merkmale von Forschungssoftware

Forschungssoftware selbst, aber auch die Projekte, in denen diese entstehen, sind sehr unterschiedlich. Sie unterscheiden sich in der Größe, der Projektdauer, der Anwendungsdomäne und der eingesetzten Technologie.

Die Entwicklung von Forschungssoftware ist mit spezifischen Problemen verbunden. Erste Betrachtungen zu den Problemen bei der Entwicklung von Forschungssoftware hat Segal gemacht [Seg05]. Demnach unterliegt der Software-Entwicklungsprozess in Forschung und Entwicklung besonderen Anforderungen und Beschränkungen. Den klassischen Prozessmodellen der Software-Entwicklung ist gemein, dass zu Beginn der Entwicklung das Ziel – zumindest den Anwendern – grob bekannt ist. Dies ist bei der Entwicklung von Forschungssoftware typischerweise nicht der Fall. Auch ist die Software nicht fertig, wenn sie das erste Mal betrieben wird. Vielmehr entstehen weitere Anforderungen, die durchaus widersprüchlich zu den ursprünglichen Anforderungen sein können.

Allen Entwicklungsprojekten von Forschungssoftware gemeinsam sind nach Hoffmann et al. [HLN10] die beiden folgenden Merkmale

- **Sie sind in eines oder mehrere Forschungsprojekte eingebettet.**
Dies hat Auswirkungen auf die zu entwickelnde Software, weil sich der Schwerpunkt und die Forschungsziele verschieben können. Es liegt in der Natur von Forschung, dass angestrebte Ziele nicht erreicht werden können, oder dass Ergebnisse, die erzielt wurden, einer Validierung nicht Stand halten. Dies führt dazu, dass in diesen Fällen die dazu entwickelte Forschungssoftware massiv umgearbeitet werden muss.
- **An der Entwicklung von Forschungssoftware sind Studierende massiv beteiligt.**
Viele Teilfunktionen von Forschungssoftware werden im Kontext von Abschlussarbeiten durch Studierende entwickelt. In vielen Fällen arbeiten weiterhin studentische Hilfskräfte an der Entwicklung mit. Dies führt zu zwei sehr charakteristischen Herausforderungen.
 - a) Die Studierenden kennen typischerweise die eingesetzte Technologie, die softwaretechnischen Methoden und Arbeitsweisen sowie die angewandte Vorgehensweise im Entwicklungsprojekt nicht.
 - b) Die Mitarbeit der Studierenden im Projekt ist zeitlich beschränkt und in der Regel recht kurz (typisch 6 bis 9 Monate). So haben die Studierenden häufig keine Chance, das Entwicklungsprojekt als Ganzes zu überblicken und sind in erster Linie darauf bedacht, Ergebnisse zu erarbeiten, die ihre Ziele unterstützen (d.h., die Abschlussarbeit erfolgreich abzuschließen). Globale Anforderungen seitens des Projektes können dabei zu kurz kommen.

Diese Merkmale und Herausforderungen muss ein Projekt, in dem Forschungssoftware entwickelt werden soll, kennen und annehmen und es muss angemessen darauf reagieren. Klassische Vorgehensmodelle bieten hier wenige Antworten. Techniken und Organisationsformen der agilen Softwareentwicklung sind jedoch ein Fundus, um angemessene Vorgehensweisen für die Entwicklung von Forschungssoftware zu etablieren. Wie in vielen anderen Bereichen gilt aber auch hier, dass es nicht die eine Vorgehensweise geben kann, die für jede Projektsituation geeignet ist. Gesucht sind also Best-Practices und Vorgehensrahmen, die projektspezifisch konfiguriert werden können.

3 Ziele und zentrale Fragestellungen für den Workshop

Da es sich um den ersten Workshop zu diesem Thema handelt, sollte er weitestgehend zieloffen und interaktiv durch die Teilnehmer bestimmt sein. Auf Basis der formulierten spezifischen Merkmale von Forschungssoftware war jedoch ein denkbare Ziel des Workshops, eine erste Sammlung von Best-Practices-Techniken zu erarbeiten, die bei der Entwicklung von Forschungssoftware verwendet werden kann.

Folgende Fragestellungen sollten u.a. mit diesem Workshop adressiert werden:

- Welche verschiedenen Entwicklungsszenarien gibt es für Forschungssoftware?
- Welche positiven und negativen Erfahrungen mit Techniken etc. gibt es?
- Welche Techniken etc. haben sich unter welchen Randbedingungen bewährt?

- Wie können die Übergänge von Forschungsprototyp zu Forschungssoftware bzw. von Forschungssoftware zum Produkt gestaltet werden?

Als Input und Kommunikationsbasis wurden Projektbeschreibungen sowie Projektpräsentationen durch die Teilnehmer erstellt. Die folgenden Projekte wurden vorgestellt und diskutiert:

Projekt	
HTR Code Package (HCP)	Institut für Nukleare Entsorgung und Reaktorsicherheit, FZ Jülich
libAlf	Lehrstuhl Logik und Theorie diskreter Systeme
MeDIC	LuFG Softwarekonstruktion
SimWeld	Institut für Schweißtechnik und Fügetechnik
NCProfiler	Fraunhofer IPT
OpenFlipper	Lehrstuhl Computergraphik und Multimedia
ViSTA	Forschungsgruppe Virtuelle Realität
AProVE	LuFG Programmiersprachen und Verifikation
Scalasca	Lehrstuhl Parallele Programmierung
Energie Navigator	Lehrstuhl Software Engineering
Score-P	Jülich Super Computing Center (JSC)
SSELab	Lehrstuhl Software Engineering

4 Aufbau des Berichtes

Dieser Bericht dokumentiert den EEFSW-Workshop. Im ersten Teil sind kurze Projektpräsentationen der meisten oben aufgeführten Projekte enthalten. Im abschließenden zweiten Teil haben wir die im Workshop erarbeiteten Ergebnisse zusammengefasst.

Die Organisatoren des EEFSW-Workshops möchten sich ganz herzlich bei den Teilnehmern bedanken, die sehr intensiv und konstruktiv den Workshop mitgestaltet haben.

Außerdem möchten wir dem Forum Informatik der RWTH Aachen danken, das die Kosten für die Ausrichtung des EEFSW-Workshops übernommen hat.

5 Literatur

- [HLN10] Hoffmann, V., H. Lichter, A. Nyßen (2010): Processes and Practices for Quality Scientific Software Projects. In H. Kienle (Ed.) Proceedings of 3rd International Workshop on Academic Software Development Tools WASDeTT-3, Antwerp, pp 95-108.
- [Seg05] Segal, J.: When Software Engineers Met Research Scientists: A Case Study. In: Empirical Software Engineering, Band 10, Seiten 517–536, Hingham, MA, USA, 2005. Kluwer Academic Publishers.