

# Qualitätszentrierte Softwareentwicklung in wissenschaftlichen Arbeitsgruppen Auf dem Weg vom Prototypen zum Produkt

D. Maleike, J. Neuhaus, T. Heimann, M. Nolden, J. Poxleitner, M. Schöbinger,  
T. Schwarz, M. Seitel, I. Wegner, P. Wolber, H.-P. Meinzer, I. Wolf

Abteilung für Medizinische und Biologische Informatik, DKFZ Heidelberg  
`d.maleike@dkfz.de`

**Kurzfassung.** Neue Bildverarbeitungsmethoden werden häufig hauptsächlich für Publikationszwecke entwickelt. Die Weiterentwicklung bis zur Verwendbarkeit für klinische Anwender erfordert zusätzlichen Aufwand und wird daher oft nicht weiter verfolgt. In dieser Arbeit wurden mehrere Bildanalyseverfahren unserer Abteilung konsequent zu einem stabilen und benutzbaren Softwareprodukt weiter entwickelt, so dass es für den Einsatz bei klinischen Partnern und als Demonstrationssoftware auf Messen verwendbar ist. Wir beschreiben das Vorgehen bei der Weiterentwicklung, welche Erfahrungen gemacht wurden und welcher zusätzliche Aufwand damit verbunden ist.

## 1 Einleitung

In wissenschaftlichen Abteilungen entstehende Bildverarbeitungsmethoden sind in der Regel Machbarkeitsstudien. Sie sind mehr auf Publikationen ausgerichtet als auf die Benutzung durch Nicht-Informatiker (z.B. Mediziner). Stabilität, eine angemessene Benutzeroberfläche und Integration in den klinischen Arbeitsablauf sind häufig untergeordnete Ziele. In diesem Beitrag haben wir mehrere fortgeschrittene Segmentierungs- und Analyseverfahren unserer Abteilung zu einem Softwareprodukt weiter entwickelt, das Ansprüchen an die Softwarequalität so weit genügt, dass es für umfangreiche Evaluationen durch klinische Partner und für Demonstrationen auf Messen geeignet ist. Die entwickelte Anwendung ist in die CHILI<sup>®</sup>-PACS-Workstation integriert. Die folgenden Abschnitte beschreiben die notwendigen Schritte auf dem Weg zu diesem Softwareprodukt und die gewonnenen Erfahrungen.

## 2 Material und Methoden

Die folgenden Abschnitte beschreiben die Werkzeuge und die Vorgehensweise, die in der Weiterentwicklung der Anwendungsmodule zu einem hochqualitativen Softwareprodukt verwendet wurden.

## 2.1 Medical Imaging Interaction Toolkit

Die Softwareentwicklung in unserer Abteilung ist auf das Medical Imaging Interaction Toolkit (MITK) aufgebaut. MITK ist ein Softwareframework für das Rapid Application Development im Bereich medizinischer Bildverarbeitung.

**Versionsverwaltung** Alle Entwickler arbeiten an einer gemeinsamen in zahlreiche Module aufgeteilten Quellcodebasis. Damit Änderungen bei der Vielzahl an Entwicklern nachvollziehbar bleiben, wird das Versionierungssystem Subversion eingesetzt. Es erlaubt das Kommentieren von Änderungen, Archivierung alter Versionen, sowie bei Bedarf das Auftrennen der Quellcodeversion in verschiedene parallele Entwicklungszweige (Branching).

**Build Management** Mit dem Ziel, Anwendungen „in die Klinik“ zu bringen, ist es notwendig verschiedene Rechnerkonfigurationen zu unterstützen. Um dieses Ziel zu erreichen, wird das Continuous-Integration-Tool „Dart“ aus ITK verwendet. Client-Rechner mit verschiedenen Betriebssystemen, Compilern und Bibliotheksversionen übersetzen jede Nacht die aktuelle Quellcodeversion und führen automatisierte Modul- und Anwendungstests aus. Fehler beim Kompilieren und Testen werden den Entwicklern, die für die entsprechende Codeänderung verantwortlich sind, automatisiert per Email mitgeteilt. Zusätzlich werden die Ergebnisse auf einer Webseite veröffentlicht. Fehler, die nur in bestimmten Konfigurationen auftreten, werden so zeitnah bemerkt. Außerdem steht immer eine aktuelle Binärversion für jede unterstützte Rechnerkonfiguration zur Verfügung.

**Fehler-/Anforderungsverwaltung** Zur Erfassung und Verwaltung von Fehlermeldungen (Bug-Reports) und Anforderungen (Feature-Requests) wird das Fehlerverwaltungssystem Bugzilla verwendet. Anwender und Tester können damit Fehlerberichte sowie neue Anforderungen an Entwickler senden. Die Entwickler können Fehlerberichte diskutieren, einem Verantwortlichen zuweisen, Anforderungen zu geplanten Meilensteinen zuordnen und die Bearbeitung der Fehler dokumentieren.

## 2.2 Vorgehensweise

Um das Ziel eines bedienbaren und stabilen Softwareprodukts zu erreichen, haben wir einen komplettenentwicklungsprozess von der Definition von Zielen, daraus abgeleiteten funktionalen und qualitativen Anforderungen bis zu deren Umsetzung und einer intensiven Qualitätssicherung durchlaufen. Die funktionalen Anforderungen ergaben sich aus einer Reihe von medizinisch relevanten Fragestellungen, die mithilfe der in der Abteilung bereits verfügbaren Software kurzfristig gelöst werden konnten. Als Zielsysteme wurden SUSE Linux 10.2 und Windows XP mit festgelegten Versionen der verwendeten Bibliotheken definiert.

Aus dem Ziel, dass die Software bei klinischen Partnern eingesetzt werden soll, wurden die beiden Qualitätsanforderungen „Benutzbarkeit“ und „Zuverlässigkeit“ abgeleitet. Um das Ziel „Benutzbarkeit“ zu erreichen, wurde eine Menge einfacher Regeln für die Benutzeroberfläche definiert – nicht jedoch neue Bedienkonzepte eingeführt, weil dies in der Regel komplexe Änderungen bedeutet.

Durch Konzentration auf bereits existierende Softwaremodule war ein Großteil der funktionalen Anforderungen bereits erfüllt. Die noch fehlenden Teile wurden als erster Arbeitsschritt vervollständigt, so dass eine funktional vollständige Version der Software erstellt werden konnte. Parallel dazu wurde ein Demosystem auf einem Notebook eingerichtet, auf dem die Software dank eines lokalen PACS Servers mit anonymisierten Bilddaten auch ohne Netzwerkverbindung innerhalb der CHILI-PACS-Workstation laufen kann.

Nach Fertigstellung des ersten Prototypen wurde im Versionsverwaltungssystem ein neuer Entwicklungszeitweig eingerichtet, um die qualitative Weiterentwicklung von der parallel laufenden funktionalen Weiterentwicklung zu trennen. Der Prototyp wurde anschließend von einem Qualitätssicherungsteam auf die Qualitätsanforderungen hin getestet. Gefundene Fehler wurden mit dem Fehlerverwaltungssystem ausführlich dokumentiert und an die zuständigen Entwickler verteilt. Nach zwei Fehlerbehebungsrounds (je eine Woche) wurde eine Release-Kandidat-Version an ein auf 14 Personen erweitertes Testteam (davon zwei Mediziner) verteilt. Nach zwei weiteren Iterationen erfüllte die Software dann alle funktionalen und qualitativen Anforderungen und es konnte ein Release erstellt werden.

### 3 Ergebnisse

Diese Arbeit hat drei unterschiedliche, aber gleichermaßen wichtige Ergebnisse. Zum einen existiert nun ein stabiles Softwareprodukt, das an klinische Partner für Tests und Evaluationen herausgegeben werden kann. Zusätzlich wurde eine Vorgehensweise entwickelt, wie in einer wissenschaftlichen Abteilung der Übergang vom internen Prototyp zu einer extern verwendbaren Version machbar ist. Zusätzlich wurde durch die Arbeit auch der Aufwand quantifizierbar, der für die Weiterentwicklung eines Prototyps zu einem extern verwendbaren Softwareprodukt notwendig ist.

#### 3.1 Die Softwaremodule

Das entwickelte Softwareprodukt besteht aus folgenden Anwendungsmodulen:

- *Bronchialbaumsegmentierung auf CT-Daten* mit einem Bereichswachstumsverfahrens, dessen Auslaufen in die Lunge automatisch detektiert wird [1].
- Interaktive *Lymphknotenvolumetrie auf CT-Daten* durch eine deformierbare Oberfläche mit Formwissen zum Monitoring über den Therapieverlauf [2].
- *Lungensegmentierung auf Cine-MR-Daten (3D+t)* über statistische Formmodelle. Beweglichkeit und Volumenverlauf der Lunge über die Zeit sind interessant für die Diagnose.

- *Herzventrikelvolumetrie auf MR-Daten* durch statistische Formmodelle, woraus klinische Maße wie die Ejektionsfraktion bestimmt werden können [3].
- *Modellierung der Herzklappenauflösung (Annulus) auf zeitaufgelösten 3D-Ultraschall-Daten* durch interaktive Definition eines Splines. Dies kann die Grundlage für eine Prothesenauswahl bilden [4].
- *Leberresektionsplanung auf CT-Daten* aufgrund manueller Segmentierungen und Lebergefäß-Versorgungsgebietsberechnung [5].
- *Verkürzte Leber-OP-Planung auf CT-Daten* durch automatische Lebersegmentierung in ca. 2 Minuten [6] und manuelle Definition einer gekrümmten Schnittebene in volumenvisualisierten Lebergefäßen.
- *Gehirntumorvolumetrie auf MR-Daten* mittels interaktiver Graphcut-Segmentierung als Grundlage der Therapieplanung.

### 3.2 Erfahrungen aus dem Projekt

Die wesentlichen Erkenntnisse aus dem Projekt sind:

- Die Aufstellung medizinisch relevanter Fragestellungen zu Beginn hat sich als nützliche Grundlage für alle zu treffenden Entscheidungen im Projektverlauf erwiesen.
- Das Abspalten der qualitativen von der funktionalen Weiterentwicklung durch einen Branch im Versionsverwaltungssystem hat sich als nützlich erwiesen. Da Fehlerbehebungen in der Regel in beiden Entwicklungszweigen durchgeführt werden sollen, ergibt sich ein zusätzlicher Schulungsbedarf bzgl. des Versionierungstool.
- Bei der Kombination mehrerer Anwendungsmodule kann es zu Seiteneffekten kommen, z.B. dass ein Modul fälschlicherweise Daten eines anderen Moduls ändert oder löscht. Um diese Fehler zu finden, sollten die Module möglichst schnell zu einem Gesamtsystem integriert und als solches getestet werden. Hieraus lässt sich die Designregel ableiten, dass Module keine impliziten Annahmen über den Zustand ihrer Daten treffen dürfen, sondern diese bei jedem Zugriff auf Validität testen müssen.
- Mit Tests, bei denen mehrere Datensätze hintereinander bearbeitet werden, können Speicherlecks gefunden werden. Speicherlecks können unter anderem dadurch entstehen, dass ein Anwendungsmodul Daten eines anderen Moduls referenziert und so die Freigabe verhindert. Dies sollte ebenfalls durch eine Designregel verboten werden.
- Das manuelle Testen durch ein Testteam mit unterschiedlichen Hintergründen hat sich sehr bewährt. Damit sollte begonnen werden, sobald die Software eine Grundstabilität erlangt hat.

### 3.3 Aufwandsabschätzung

Tabelle 1 zeigt den geschätzten Aufwand für das Projekt. Hierbei ist hervorzuheben, dass der Aufwand für einen ersten Prototypen ebenso hoch ist wie die Weiterentwicklung zur Erfüllung der Qualitätsanforderungen.

**Tabelle 1.** Geschätzter Aufwand für die Erstellung des Softwareprodukts. Die Spalte „Entwicklungsstand“ zeigt den geschätzten Zustand der Softwaremodule bezogen auf die funktionalen und qualitativen Anforderungen (vor Projektbeginn). Die Spalte „Entwicklungsaufwand“ enthält die Arbeitszeit in Personentagen (8h-Tage) aufgeteilt in Aufwand zur Erfüllung der funktionalen und der Qualitätsanforderungen und in den Test- und Verwaltungsaufwand. Der Gesamtaufwand beträgt etwa ein Personenjahr

	Entwicklungsstand	Entwicklungsaufwand
Funktionale Anforderungen	65 %	126 Personentage
Qualitative Anforderungen	45 %	105 Personentage
Testen	—	10 Personentage
Verwaltung	—	13 Personentage
Gesamt		254 Personentage

## 4 Diskussion

Wir haben in dieser Arbeit ein stabiles Softwareprodukt entwickelt, das eine Auswahl der Forschungsarbeiten unserer Abteilung beinhaltet. Es wurde so weit entwickelt, dass es an klinische Partner herausgegeben sowie auf Messen als Demonstrationssystem gezeigt werden kann. Auf dem Weg zu einem einsetzbaren Produkt fehlt hier noch die Zertifizierung nach dem Medizinproduktegesetz. Hierfür ist beträchtlich mehr Aufwand notwendig, da bei einem solchen Vorhaben zusätzlich eine systematische Planung und Dokumentation nötig ist.

Das Projekt hat neben dem Softwareprodukt an sich vor allem Erfahrungen über das optimale Vorgehen und den benötigten Aufwand gebracht. Dies gilt es nun in Schulungsunterlagen den Entwicklern zugänglich zu machen und bei zukünftiger Projektplanung zu berücksichtigen.

## Literaturverzeichnis

1. Wolber P, Wegner I, Heimann T, et al. Tracking und Segmentierung baumförmiger tubulärer Strukturen mit einem hybriden Verfahren. Proc BVM. 2008.
2. Maleike D, Fabel M, Tetzlaff R, et al. Lymph node segmentation on CT images by a shape model guided deformable surface method. Proc SPIE. 2008;Accepted.
3. Schwarz T, Heimann T, Wolf I, et al. 3D heart segmentation and volumetry using deformable shape models. Proc Comp Cardiol. 2007;In print.
4. Seitel M, Frank C, Simone RD, et al. Unterstützung der Planung von Mitralklappen-Annuloplastien mittels 4D-Echokardiographieaufnahmen. Proc CURAC. 2007; p. 283–6.
5. Meinzer HP, Thorn M, Cardenas CE. Computerized planning of liver surgery: An overview. Comput Graph. 2002;26(4):569–76.
6. Heimann T, Münzing S, Meinzer HP, et al. A shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation. Inf Process Med Imaging. 2007;20:1–12.