Entwurf und praktische Erprobung einer Reengineering-Strategie für das Modellierungswerkzeug 3LGM^2-Baukasten

Sebastian Weber, Thomas Wendt, Gert Funkat
Institut für Medizinische Informatik, Statistik und Epidemiologie der Universität Leipzig

{sebastian.weber | thomas.wendt | gert.funkat}@imise.uni-leipzig.de

Zur Modellierung von Krankenhausinformationssystemen wird am Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE) der Universität Leipzig das Metamodell 3LGM^2 [1] entwickelt. Dieses soll durch eine graphische Darstellung des Informationssystems das Verständnis für die Funktion des Systems, sowie dessen Wartbarkeit und Erweiterbarkeit unterstützen. Um die Modellierung von Informationssystemen mit Hilfe des Metamodells 3LGM^2 zu vereinfachen, wurde vor nunmehr drei Jahren damit begonnen, eine Software zu entwickeln, welche den Informationsmanager bei dieser Aufgabe unterstützt. Diese, der so genannte 3LGM^2-Baukasten, ist nun zu einem sehr umfangreichen Werkzeug angewachsen, mit dem es möglich ist, komplexe Systeme zu modellieren. Zu den umfangreichen Funktionen gehören u.a. die Darstellung des modellierten Informationssystems über verschiedene Ansichten, das Anfertigen von Analysen und den Nutzer unterstützende Funktionen, wie das Importieren und Exportieren von Modellen und eine umfassende Online-Hilfe.

Bei der Entwicklung der Software wurden einige grundlegende Fehler begangen:

- Es wurde nicht nach einem gängigen Softwareentwicklungsprozeßmodell vorgegangen. Das Vorgehen kann am ehesten mit der Entwicklung eines Prototyps verglichen werden.
- Viele Funktionen wurden der Software im Zuge der evolutionären Weiterentwicklung hinzugefügt. Eine Spezifikation für diese Funktionen existiert nur unvollständig. Langfristige Ziele wurden nicht detailliert ausgearbeitet.
- Der Quellcode wurde nur lückenhaft dokumentiert. Eine Dokumentation auf abstrakterem Niveau wurde nicht entwickelt.

Die Software hat nun einen Umfang und eine Komplexität erreicht, bei der die Folgen der genannten Fehler nicht mehr handhabbar sind und eine weitere evolutionäre Entwicklung nur noch mit großem Aufwand durchführbar ist. Daher wird zur Zeit im Rahmen einer Diplomarbeit eine Reengineering Strategie für die Software entwickelt. Folgende Eigenschaften des Softwareprojekts können für das Reengineering von Vorteil sein:

- Auf die Erfahrung der Entwickler kann während des Reengineering zurückgegriffen werden, da diese noch im Projekt beschäftigt sind.
- Die Software wird in der Programmiersprache Java entwickelt. Eine Sprachkonversion von einer veralteten in eine moderne Programmiersprache, wie sie häufig bei Reengineering Projekten der Fall ist, ist somit nicht von Nöten.
- Bei der Software handelt es sich um eine Einzelanwendung. Die Betrachtung von Schnittstellen zu anderen Applikationen ist somit nicht nötig.

Da das Projektteam sehr klein ist, wird das Reengineering schrittweise stattfinden (Chicken-Little-Strategy [2]). Das inkrementelle Vorgehen muß mit der parallel dazu stattfindenden Weiterentwicklung abgestimmt werden.

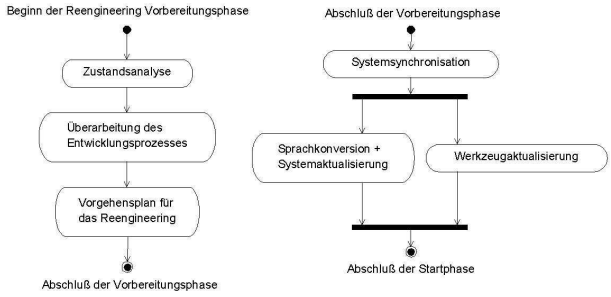

Abbildung 1: Vorbereitungs- und Startphase für das Reengineering

Das Reengineering wird in drei Phasen ablaufen. Abbildung 1 zeigt die ersten beiden Phasen. In der Vorbereitungsphase werden zuerst durch eine Zustandsanalyse die bisherigen Probleme analysiert, die zur Notwendigkeit eines Reengineering geführt haben, und die daraus resultierenden Ziele des Reengineering festgelegt. Anhand der gefundenen Probleme wird im Folgenden der bisherige Softwareentwicklungsprozeß überarbeitet. Daraufhin wird das Vorgehen beim Reengineering geplant. Dazu gehört die Vorgabe des Prozeßmodells für das inkrementelle Reengineering in der Reengineering Phase (dritte Phase).

In der zweiten Phase, der Startphase, werden Software und Softwareumfeld auf ein Reengineering vorbereitet. Die Software wird synchronisiert, das heißt aus den bestehenden Versionen wird ein

fehlerfreies Release der Software erzeugt, welches als Ausgangspunkt für das Reengineering und die dazu parallel laufende Weiterentwicklung der Software dient. Diese Version wird einer Sprachkonversion auf Java 1.5 unterzogen und eingebundene Frameworks werden aktualisiert. Eine weitere Aufgabe in der Startphase ist die Aktualisierung der für die Softwareentwicklung verwendeten Werkzeuge, gemäß den in der Überarbeitung des Softwareentwicklungsprozesses und der Planung des Reengineering gemachten Vorgaben. Die dritte Phase des Projekts beinhaltet das inkrementelle Reengineering der einzelnen Komponenten. Für jede Komponente wird dabei der komplette Reengineering Zyklus, also Reverse Engineering, Restructuring und Forward Engineering, durchlaufen. Wie dieser Prozeß aussehen soll, muß aber noch detailliert in der Reengineering Planungsphase festgelegt werden.

Das Projekt befindet sich zur Zeit noch in einer sehr frühen Phase. Schwierigkeiten bereitet vor allem, daß im Gegensatz zu Softwareentwicklungsprozessen im Bereich des Reengineering nur sehr wenige anpaßbare und praktisch verwendbare Prozeßmodelle existieren. Da die Weiterentwicklung der Software in Zukunft nach einem Rational Unified Process (RUP) [3] erfolgen soll, wird nun das inkrementelle Reengineering im RUP abgebildet. Anschließend wird ein erster Reengineering Zyklus durchgeführt. Die dabei gewonnenen Erkenntnisse werden zur Verbesserung des Reengineeringprozesses dienen.

Literatur

[1] Winter, Alfred; Brigl, Birgit; Wendt, Thomas: Modeling Hospital Information Systems (Part 1): The Revised Three-layer Graph-based Meta Model 3LGM. www.3lgm2.de/Publikationen/Dokumente/ModelingHIS1.pdf, 2003

[2] Brodie, Michael L.; Stonebraker, Michael: Darwin: On the Incremental Migration of Legacy Information Systems. http://db.cs.berkeley.edu/papers/S2K-93-25.pdf, 1993

[3] Kruchten, Phillipe: The Rational Unified Process 3E, An Introduction. Addison-Wesley, 2004