



Hochschule Heilbronn
Fakultät Wirtschaft 1
Studiengang Electronic Business

Diplomarbeit

Modellgetriebene Softwareentwicklung
und deren Auswirkung auf die
Entwicklungsmethodologie
von Standardsoftware

Vorgelegt bei
Prof. Dr. Michael Gröschel

Von
Bastian Harzenetter
Matrikelnummer: 160825

Im
Wintersemester 2007/2008

Management Summary

Mit modellgetriebener Softwareentwicklung (MDSD) wird heute ein moderner Softwareentwicklungsansatz assoziiert. Große Aufmerksamkeit erfuhr MDSD durch den 2003 von der Object Management Group (OMG) vorgestellten Standard der Model Driven Architecture (MDA). Dieser Standard bietet ein Rahmenkonzept, bei welchem durch verschiedene Modelldefinitionen und deren schrittweise Transformation eine Anwendung erstellt wird. Die wesentlichen Potenziale dieses neuen Paradigmas sind unter Anderem eine Steigerung der Entwicklungsproduktivität und -effizienz, eine verbesserte Portabilität und Wartbarkeit der Anwendung und eine Wiederverwendbarkeit von erstellten Artefakten.

Zur Realisierung dieser Vorteile werden Modelle verwendet, die soweit formalisiert und mit Informationen angereichert sind, dass diese durch Transformationen verfeinert werden und unter Hinzunahme von Generatoren und Interpretern in lauffähige Software umgewandelt werden können. Auf den ersten Blick scheint dieser Gedanke der Anwendungsentwicklung nichts Neues darzustellen, da die angewandten Techniken hinlänglich bekannt und weit verbreitet sind, jedoch ist die Kombination dieser zusammen mit dem Entwickeln auf einer höheren Abstraktionsebene durch die Modellierung in diesem Umfang durchaus avantgardistisch. Folglich wird die letztendliche Anwendungsentwicklung bei MDSD nicht mehr auf Codeebene vorgenommen sondern auf Modellebene.

Neben den theoretischen Grundlagen modellgetriebener Entwicklung befasst sich die folgende Arbeit im Speziellen mit den zu erwartenden Auswirkungen von MDSD auf das Entwicklungsvorgehen von Standardsoftware im Credit Risk Projekt der SAP AG. In diesem Zusammenhang wird der im Projekt angewandte und SAP-weit vorgeschriebene Entwicklungsprozess des Product Innovation Lifecycles (PIL) analysiert. Im Laufe dieser Untersuchung stellte sich dabei heraus, dass dieser ähnliche Ziele verfolgt, wie sie auch in modellgetriebenen Ansätzen angestrebt werden. In beiden Fällen wird eine schrittweise Detaillierung der Entwicklungsartefakte vorgenommen und eine Trennung der Funktionalität von der dafür verwendeten Technologie forciert, weshalb der Entwicklungsprozess bei SAP und im Projekt Credit Risk für MDSD geeignet zu sein scheint. Durch eine Umstellung auf einen MDSD-Ansatz würden die derzeit verwendeten Dokumente ihre zentrale Rolle in der Entwicklung verlieren und Modelle diese Position einnehmen. Jedoch sind die durch eine Umstellung zu erwartenden positiven Folgen relativ gering, da viele Potenziale modellgetriebener Entwicklung bereits im derzeitigen Entwicklungsvorgehen kompensiert werden oder nicht relevant sind.

Im konkreten Werkzeugbereich des Credit Risk Projekts wurden der Moduleditor, das Ableitungstool und das Formula Derivation Tool (FDT) sowie das Merkmals- und Kennzahlrepository betrachtet. Mit Hilfe des Moduleditors werden Verarbeitungsschritte definiert, das Ableitungstool und das FDT dienen zum Festlegen und Modellieren von Business Rules und im Merkmals- und Kennzahlrepository wird die Modellierung verwendeter Datenfelder vorgenommen und bestehende Datenfelder erweitert. Wenn man im Vergleich zu diesen heute verfügbare MDSD-Werkzeuge betrachtet, so stellt man fest, dass MDSD-Werkzeuge gegenüber den vorgestellten Entwicklungswerkzeugen eher als nachteilhaft zu betrachten sind. Dies liegt zum Einen daran, dass mit Hilfe der erwähnten Werkzeuge unterschiedliche Einsatzbereiche der Softwareentwicklung abgedeckt werden und somit eine Abstrahierung dieser Domänen vorgenommen wird. Zum Anderen ist der Funktionsumfang heute verfügbarer MDSD-Werkzeuge, der sich meist auf die Generierung von CRUD-Methoden (Create, Read, Update, Delete) beschränkt, nicht ausreichend, um die angewandten Werkzeuge adäquat zu ersetzen. Die vorgestellten Werkzeuge können vereinfacht als domänenspezifische Modellierungswerkzeuge betrachtet werden, die es wie MDSD-Werkzeuge ermöglichen, Anwendungslogik getrennt vom Anwendungscode zu erstellen und zu warten, was durch ausgedehnten Einsatz von Codegenerierung möglich ist. Da mit Hilfe der eingesetzten Werkzeuge eine Vielzahl der von MDSD propagierten Potenziale bereits substituiert werden, bietet eine Ablösung der bestehenden Werkzeuge keinen Vorteil.

Des Weiteren wurde im Laufe der Diplomarbeit eine Erforschung der vorhandenen Modellierungskennntnis im Projektteam durchgeführt. Mit Hilfe eines Fragebogens wurden alle an der Entwicklung beteiligten Personen mitunter bezüglich ihrer Kenntnisse in verschiedenen Modellierungsbereichen befragt. Das Resultat der Befragung ist, dass das Team positiv gegenüber der Modellierung eingestellt ist und diverse Vorteile in der Modellierung sieht, was als Grundvoraussetzung für die Einführung eines modellgetriebenen Ansatzes zu werten ist. Auch stellte sich heraus, dass das Projektteam bereits über ein breit gefächertes Wissen in der Unified Modeling Language (UML) verfügt, was im Falle der Einführung eines MDSD-Ansatzes auf Basis der UML komplementär wirken würde. Auch im Umgang mit den SAP-internen Modellen besitzen die Teammitglieder Erfahrung. Allerdings sind im Falle einer auf diesen basierend MDSD-Einführung Schulungen notwendig, um eine einheitliche Wissensbasis zu schaffen. Prinzipiell sind die Auswirkungen, die das Projektteam betreffen werden, als gering zu werten. Aufgrund der positiven Einstellung und des vorhandenen Modellierungs-

wissens würden sich die Auswirkungen auf Schulungen im Umgang mit der Modellierung und den dazugehörigen Werkzeugen beschränken.

Abschließend kann festgehalten werden, dass der Entwicklungsprozess bei SAP und das Vorgehen im Credit Risk Projekt für MDSD zwar geeignet sind und auch das Projektteam auf die Einführung eines modellgetriebenen Ansatzes vorbereitet ist, dennoch muss tendenziell von einem Einsatz von MDSD derzeit abgeraten werden. Dies liegt primär an den Stärken der derzeit angewendeten Werkzeuge und dem Mangel an adäquaten MDSD-Werkzeugen. Auch das Customizing der Software ist ein noch nicht betrachteter Grund von MDSD abzuraten.

Inhaltsverzeichnis

Management Summary.....	V
Inhaltsverzeichnis	IX
Abbildungsverzeichnis	XII
Tabellenverzeichnis.....	XIII
Abkürzungsverzeichnis	XIV
1 Einleitung	1
1.1 Motivation.....	2
1.2 Zielsetzung.....	2
1.3 Vorgehensweise	3
2 Grundlagen modellgetriebener Softwareentwicklung.....	4
2.1 Definition und Begriffsabgrenzung	4
2.2 Geschichte der Softwareentwicklung	5
2.3 Grundelemente modellgetriebener Softwareentwicklung	6
2.3.1 Abstraktion	7
2.3.2 Die Rolle von Modellen und Metamodellen	8
2.3.3 Domänenspezifische Sprachen	11
2.3.4 Transformationen, Generatoren und Interpreter	12
2.3.5 Softwareproduktfamilien	13
2.4 Allgemeine Vorteile modellgetriebener Entwicklung	14
2.5 Modellierungsarten und Werkzeuge	15
2.5.1 Modellierungsarten.....	15
2.5.2 Modellierungswerkzeuge.....	18
2.5.3 Das Zachman Framework.....	19
3 Modellgetriebene Ansätze und ihre Unterschiede	22
3.1 Model Driven Architecture	22
3.1.1 MDA und die OMG.....	22
3.1.2 MOF, XMI und weitere Standards	23
3.1.3 Kritik und Schwachpunkte	24
3.2 Software Factories und Microsoft.....	25

3.2.1	Vorstellungen und Auswirkungen von Software Factories	27
3.2.2	Abgrenzung zur MDA und MDSD	27
3.3	Weitere Ansätze	28
3.3.1	Domänenspezifische Modellierung	28
3.3.2	Generative Programming	28
3.3.3	Agile Model-Driven Development	29
3.4	Zusammenfassung	29
4	Modellgetriebene Technologien bei SAP	31
4.1	Modellierungsinfrastruktur: MOIN	31
4.2	Geschäftsprozessmodellierung: Galaxy	33
4.3	Oberflächenmodellierung: Visual Composer	34
5	Entwicklungsmethodologie von Standardsoftware bei SAP	36
5.1	Standardsoftware	36
5.2	Der Product Innovation Lifecycle	37
5.2.1	Invent-Phase	38
5.2.2	Define-Phase	38
5.2.3	Develop-Phase	40
5.2.4	Deploy-Phase und Optimize-Phase	42
6	Entwicklung im Projekt Credit Risk	43
6.1	Projektbeschreibung	43
6.2	Der Bank Analyzer	44
6.3	Werkzeuge	46
6.3.1	Modul Editor	46
6.3.2	Ableitungstool und FDT	48
6.3.3	Merkmals- und Kennzahlrepository	50
6.4	Herausforderungen bei der Entwicklung	52
7	Einsatzmöglichkeiten und Grenzen modellgetriebener Entwicklung	53
7.1	Einsatzmöglichkeiten im Product Innovation Lifecycle	53
7.1.1	Projektmanagement	53
7.1.2	Generelles Vorgehen	54
7.1.3	Customizing	56

7.2	Einsatzmöglichkeiten im Werkzeugbereich	57
7.2.1	Eigenschaften der eingesetzten Werkzeuge	57
7.2.2	Bewertung im Vergleich zu MDSD-Werkzeugen.....	59
7.3	Einsatzmöglichkeiten im Team	61
7.3.1	Aufbau des Fragebogens	61
7.3.2	Ergebnisse der Befragung.....	62
7.3.3	Bewertung.....	64
7.4	Zusammenfassende Betrachtung	65
8	Auswirkung auf die Entwicklung im Projekt Credit Risk.....	66
8.1	Auswirkung auf das Entwicklungsvorgehen	66
8.1.1	Allgemeine Auswirkungen	66
8.1.2	PIL-spezifische Auswirkungen.....	69
8.2	Auswirkungen auf den Werkzeugeinsatz	70
8.3	Auswirkungen auf das Team	73
9	Fazit.....	74
	Anhang I: Fragebogen	XVII
	Anhang II: Ergebnisse der Befragung	XXV
	Anhang III: Modellierungsidee	XXIX
	Literaturverzeichnis	XXXI
	Eidesstattliche Erklärung	XLIX