SOFISMO
Ihr daten pilot

FORMALISIERTES SOFTWARE ENGINEERING

Die Formalisierung des Software Engineerings beginnt mit der Formalisierung des Requirements Engineerings; und zwar auf allen Abstraktionsebenen bis hin zum einzelnen Artefakt. Dies ist heute dank dem Einsatz von Domänen-spezifischer Modellierung (DSM) eine Realität. Die Grundidee ist, jedem Know-how-Träger, sei er fachlich oder technologisch gelagert, eine Modellierungs- oder Spezifikationssprache in die Hand zu drücken, mit der er selbständig die gewünschte, in seinen Kompetenzbereich fallende Funktionalität vollständig und lösungsneutral definiert. Damit wird die Basis für eine automatisierte Weiterverarbeitung dieser Spezifikationen zu einer Vielfalt von Artefakten wie Dokumentation, Testscripts, Konfigurationsparameter und selbstverständlich auch Code gelegt. Der gesamte Entstehungsprozess von Softwaresystemen erreicht eine bisher unerreichte Produktivität und Rückverfolgbarkeit. Die DSM-Methodik dürfte die Überlebensfähigkeit der Software-Entwicklung in Hochlohnländern wie der Schweiz oder Deutschland massgeblich beeinflussen.

Zusammenfassung des Vorgehens mit Domänen-spezifischer Modellierung

Domänenspezifische Modellierung (DSM) stammt ursprünglich aus dem „Product-Line Engineering“ und beruht auf eine perfekte Trennung von Spezifikation (aka: Requirements Engineering) und Implementierung.

Requirements Engineering

- Als Erstes wird das Gesamtproblem mittels einer Domänenanalyse in kleine, überschaubare Subdomänen unterteilt, die sowohl fachlich als auch technisch sein können. Diese Modularisierung der anstehenden Problematik ist von zentraler Bedeutung für die Beherrschung von Komplexität.
- Für jede Subdomäne wird eine eigene Modellierungs- oder Spezifikationssprache (= Metamodell) definiert, welche alle Konzepte (aka: Abstraktionen) repräsentiert, die in dieser Subdomäne für die Definition von Funktionalität benötigt werden. Besonders interessant dabei ist die Tatsache, dass die Abstraktionsebenen der Modellwelt beliebig den Bedürfnissen der Anwender angepasst werden können.
- Darauf aufbauend, wird eine textuelle (oder graphische) Notation (Symbolik) mit zugehörigem Editor (z.B. auf der Basis von Xtext) definiert, welche die komfortable Eingabe von Inhalten (aka: Instanziierung von Modellen) innerhalb des vereinbarten Kontexts ermöglicht. Solche Notationen können beliebig den Bedürfnissen der Benutzer angepasst werden und unterstützen sowohl die Mehrsprachigkeit als auch die Verwendung von verschiedenen Dialekten ein- und derselben Fachsprache. Damit erhalten die

(Fach)-Verantwortlichen ein optimal auf ihre Bedürfnisse abgestimmtes Spezifikationswerkzeug. Die Betroffenen müssen sich weder in "fremdartige" und komplexe Tools einarbeiten, noch müssen sie IT-"Sprachen" (z.B. UML) o.ä. beherrschen.
- Das Resultat ist ein Satz von Modellinstanzen, die eine vollständige, widerspruchsfreie und redundanzfreie Spezifikation darstellen. Jede Änderung wird nur einmal spezifiziert und automatisch in allen resultierenden Artefakten propagiert.

Implementierung
Im Prinzip ist eine Entwicklungsabteilung frei, wie sie die oben erwähnten Spezifikationen umsetzt. Wegen der formalen Form bietet sich aber an, die Umsetzung zu automatisieren:
- Die Realisierung erfolgt über eine sogenannte Referenzapplikation, die mindestens eine Instanz jedes Sprachkonstrukts aller Metamodelle in der jeweils aktuellen Technologie abbildet. Aus diesem selbst geschriebenen Code werden dann die sogenannten Code-Templates (ähnlich einem parametrisierbaren Makro) extrahiert, welche eine Musterabbildung der Fachwelt(en) auf der Codewelt darstellen. Besonders interessant ist hier, dass die Realisierung auf beliebigen Implementierungstechnologien automatisiert werden kann; also auch auf alten Legacysystemen, bestehenden Produkten und selbstgebauten Frameworks.
- Zum Schluss müssen die Applikationen resp. die Applikationsportfolios nur noch mit der obigen Modellwelt spezifiziert werden. Die Implementierung erfolgt automatisch durch das Zusammenfügen der zugehörigen parametrisierten Implementierungstemplates (Software-Fabrik). Die Agilität und Produktivität einer so arbeitenden Organisation steigt um Faktoren!

Vorteile der Domänen-spezifischen Modellierung (DSM)
(aus Requirements-Engineering Sicht)

Domänen-spezifische Modellierung ist in allen Gebieten einsetzbar, wo komplexe Überlegungen reproduzierbar dokumentiert werden müssen. Für eine IT-Abteilung mit viel Entwicklungsaufwand sind die folgenden Vorteile relevant:
- Die Spezifikationswerkzeuge können beliebig den Bedürfnissen der Benutzer angepasst werden (sowohl bezüglich Konzepten als auch bezüglich Aussehen).
- Die Gesamtproblematik kann in beliebig viele kleinere Problemkreise unterteilt werden, die sich genau an der Organisation ausrichten.
- Die häufig anzutreffende Vielfalt der Spezifikationsartefakte (Diagramme, Konzepte, Richtlinien, etc.) wird systematisch erfasst und auf reproduzierbare Weise zu einem Ganzen zusammengefügt.
- Die Einführung von DSM ist in kleinen Etappen möglich. Es müssen weder grosse "universelle" Toolentscheide gefällt werden, noch müssen grosse Ausbildungsanstrengungen unternommen werden, bevor erste produktive Resultate erzielt werden.
- Eine Organisation kann jederzeit "normal" weiterarbeiten, wobei die bis zu diesem Zeitpunkt erarbeiteten Resultate immer eine Verbesserung für die Zukunft sind.
- Die gesamte Spezifikationsphase bleibt bezüglich den späteren Implementierungstechnologien neutral.
- Die Toolpalette, die für DSM benötigt wird, ist Open Source verfügbar.
- Die Kosten für die Entwicklung und Wartung von Individualsoftware sinken massiv, weil ein grosser Teil der üblichen Fleissarbeit wegfällt.

- **Der gesamte Entstehungsprozess von Software wird 100% nachvollziehbar und die IT wird fast perfekt an den Bedürfnissen der Fachabteilung ausgerichtet.**

- **Mit Domänen-spezifischen Modellierungssprachen wird das Geschäftswissen einer Organisation explizit gemacht und technologieunabhängig konserviert.**