



Solid Innovator



MASTER THESIS "TOOLS FOR TOOLS"

Mark Jans

Scriptienummer: 577
Supervisor: Rinus Plasmeijer



Radboud Universiteit Nijmegen



MASTER THESIS "TOOLS FOR TOOLS"

Scriptienummer: 577

Student:

Mark Jans

Studentnummer s0438596

Supervisor:

Rinus Plasmeijer

Radboud Universiteit Nijmegen

Opdrachtgever:

Peter Hendriks

Info Support B.V.

Titel	Master Thesis
Project/Onderwerp	Tools for Tools
Versie	1.0
Status	Definitief
Datum	20-2-2008
Bestand	Master Thesis
Bedrijf	Info Support B.V.

Kruisboog 42, 3905 TG Veenendaal, Tel. +31 (0) 318 55 20 20 Fax +31 (0) 318 55 23 55
De Smalle Zijde 39, 3903 LM Veenendaal, Tel. +31 (0) 318 50 11 19 Fax +31 (0) 318 51 83 59
Rabobank Bergh 30.59.52.889, Postbank 47.38.593

Samenvatting

Projectspecifieke tools zijn tools die het ontwikkelproces van een bepaald software project ondersteunen, maar geen onderdeel van het project zelf zijn. Ze worden ontwikkeld door een teamlid van het project als er geen geschikte 3rd party tool beschikbaar is. Bij dit soort tools moet gedacht worden aan bijvoorbeeld code generators, data converters en tools om systeemadministratieve zaken te (semi)automatiseren.

Het ontwikkelen van projectspecifieke tools gebeurt over het algemeen zeer ad-hoc, omdat in eerste instantie verwacht wordt dat de tool maar op beperkte schaal ingezet gaat worden. Echter, later blijkt vaak dat de tool nuttig blijft tijdens de lifecycle van het project, bijvoorbeeld tijdens het testen of beheren. Omdat het aantal gebruikers groeit, er meer functionaliteit gewenst is en/of de gebruiksomgeving verandert moet de tool steeds aangepast worden. Er wordt niet goed over het ontwerp nagedacht, waardoor het doorvoeren van wijzigingen vaak lastig of zelfs helemaal niet mogelijk is.

Info Support B.V. is een IT-dienstverlener en ontwikkelt en beheert kantoorautomatiserende software projecten voor haar klanten. Hierbij streven ze ernaar om dit op een zo professioneel mogelijke manier te doen. Het competence center Java is een werkgroep binnen Info Support waar kennis en standaarden worden ontwikkeld en gedeeld voor de Java projecten.

De manier waarop projectspecifieke tools ontwikkeld worden sluit niet aan bij de professionele manier van werken. Deze thesis geeft de resultaten van een onderzoek naar het ontwikkelen van projectspecifieke tools weer, dat in opdracht van het Competence Center Java van Info Support is uitgevoerd. De focus van het oplossingsgebied ligt dan ook op Java en Eclipse, de ontwikkelomgeving die binnen dit Competence center gebruikt wordt.

Door een inventarisatie onder verschillende werknemers van Info Support uit projecten voor verschillende klanten zijn een zestal knelpunten geïdentificeerd, die vaak optreden tijdens het ontwikkelen van projectspecifieke tools: "Niet geschikt voor evolutie", "Niet geschikt voor build proces", "Weinig tot geen documentatie", "Te weinig overdracht aan PDC", "Project wordt geremd door afhankelijkheid tool" en "Geen simpele richtlijnen".

Omdat het probleem zeer algemeen is, zullen er waarschijnlijk talloze oplossingen voor de knelpunten te vinden zijn, maar door het tijdsbestek waarin het onderzoek gedaan is, is er gefocust op model driven engineering. Er is een veelzijdigheid aan projectspecifieke tools en om de knelpunten af te dekken zal hier iets generieks over gezegd moeten worden. Dit kan door middel van een abstract model. Om de ontwikkelaar te helpen, zonder allerlei richtlijnen te moeten lezen en in acht te moeten houden, kan model driven engineering ingezet worden. Dit biedt namelijk een lijdraad, waarbij een aantal stappen geautomatiseerd kunnen worden, maar toch dingen als nadenken over ontwerp afgedwongen worden.

Bij model driven engineering is een model nodig. Hiervoor hebben we naar de wat bekendere en gemakkelijk te leren diagrammen gekeken, namelijk het class diagram, data flow diagram en state machine diagram. Verder worden een aantal Eclipse plugins, wat in feite tools voor de Eclipse ontwikkelomgeving zijn, geëvalueerd. Dit zijn: "Cheat sheets", "UML2 Tools", "Eclipse Modeling Framework" en "Smart Developer Environment for Eclipse". Geen van deze bestaande mogelijkheden dekt alle knelpunten af.

Uit de inventarisatie is gebleken dat de meeste tools iets met data doen, zoals bijvoorbeeld "code generators" en "data converters". We definiëren het "tool data flow diagram", waarmee de informatie stromen binnen een projectspecifieke tool gemodelleerd kunnen worden. Dit diagram kan ingezet worden in combinatie met model driven engineering en is gemakkelijk te leren, wat een ontwikkelaar van Info Support aangetoond heeft.

Een willekeurige projectspecifieke tool, de "stored procedure generator", is ontwikkeld als proof of concept. Voor deze tool is in meerdere iteraties een tool data flow diagram gemodelleerd en aan de hand hiervan een implementatie ontwikkeld. Dit is een interessante case study, want deze tool is eerder ontwikkeld en toen kwamen alle knelpunten voor. We tonen aan dat het invoeren van model driven engineering in combinatie met het tool data flow diagram alle knelpunten afdekt, terwijl het ontwikkelproces er niet moeilijker van wordt.

De conclusie is dat projectspecifieke tools op een ad-hoc wijze ontwikkeld worden, omdat er gedacht wordt dat ze maar een gering aantal keren nodig zullen zijn. In de praktijk blijken ze echter veel langer nuttig, maar dan is er niet nagedacht over ontwerp en is het aanpassen ervan lastig of zelfs niet mogelijk. Door model driven engineering in combinatie met het tool data flow diagram toe te passen in het ontwikkelproces worden alle knelpunten afgedekt.