

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/20225> holds various files of this Leiden University dissertation.

Author: Heijstek, Werner

Title: Architecture design in global and model-centric software development

Date: 2012-12-05

Architecture Design in Global and Model-Centric Software Development

Werner Heijstek

Architecture Design in Global and Model-Centric Software Development

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus Prof.mr. P.F. van der Heijden
volgens besluit van het College voor Promoties
te verdedigen op woensdag 5 december 2012
klokke 10:00 uur

door

Werner Heijstek
geboren te Dordrecht
in 1982

Promotiecommissie

- Promotor : Prof. dr. J.N. Kok
Copromotor : Dr. M.R.V. Chaudron
- Overige Leden : Prof. dr. Th.H.W. Bäck
Prof. dr. P. Runeson (*Lund University*)
Prof. dr. ir. P. Avgeriou (*University of Groningen*)



The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

Parts of this research were supported by the Accelerated Delivery Center (ACD) of Capgemini the Netherlands.



Parts of this research were supported by the Israeli Ministry of Foreign Affairs Division for Cultural and Scientific Affairs via the Netherlands Organization for International Cooperation in Higher Education (NUFFIC).

Parts of this research were supported by the Platform Outsourcing the Netherlands (PON).



Images used on the cover:

- (1) “*Carte des Indes orientales I. feuille, dans la quelle on represente les Indes deca la Riviere de Ganges, le Golfe de Bengale, Siam, Malacca, Sumatra dressee par Mr. de Tobie Mayer de la Societe Geograph.*” (1748)
- (2) “*AIM UML Class diagram.*” from “*The caBIG™ Annotation and Image Markup Project*” D. S. Channin et al. *Journal of Digital Imaging* Vol. 23 Issue 2 (2010)

Typeset by \LaTeX , printed by Ipskamp Drukkers.
ISBN: 978-94-6191-525-2

Copyright © 2012 Werner Heijstek

“Einstein argued that there must be simplified explanations of nature, because God is not capricious or arbitrary. No such faith comforts the software engineer. Much of the complexity that he must master is arbitrary complexity, forced without rhyme or reason by the many human institutions and systems to which his interfaces must conform. These differ from interface to interface, and from time to time, not because of necessity but only because they were designed by different people, rather than by God.”

— Frederick O. Brooks, Jr.

Contents

1	Introduction	1
1.1	Central Concepts	1
1.1.1	Global Software Development	2
1.1.2	Software Architecture	4
1.1.3	The Unified Modeling Language	6
1.1.4	The Rational Unified Process	8
1.1.5	Model-Driven Development	10
1.2	Problem Statement	11
1.3	Research Objectives	11
1.4	Research Methodology	12
1.4.1	Case Studies	13
1.4.2	Controlled Experiments	13
1.4.3	Grounded Theory	13
1.5	Data Collection Techniques	13
1.5.1	Software Repository Mining	14
1.5.2	Document Analysis	15
1.5.3	Interviews	15
1.5.4	Data Sources	15
1.6	Contributions and Outline	16
1.7	Publications	17
2	Comparison of Industrial Process Descriptions for GSD	21
2.1	Introduction	21
2.2	Objectives	22
2.3	Related Work	23
2.4	Method	24
2.4.1	Research Environment	24
2.4.2	Approach	25

2.5	Results	25
2.5.1	Process Sections	27
2.5.2	General Process	28
2.5.3	Additional Documentation	28
2.6	Discussion	31
2.6.1	Process Design Comparison	31
2.6.2	Process Rationale	31
2.6.3	Intended Process Use	32
2.6.4	Process Maintenance	33
2.7	Validity	34
2.8	Conclusions and Future Work	34
3	Architecture and Design Process Evaluation Through Effort Visualization	37
3.1	Introduction and Objectives	38
3.2	Related Work	39
3.2.1	RUP Humps	39
3.2.2	Other Related Work	40
3.3	Methods	41
3.3.1	Project Context	41
3.3.2	Data Collection	42
3.3.3	Visualizing Effort Data	42
3.3.4	Validation	43
3.4	Results	44
3.4.1	Project A	44
3.4.2	Project B	47
3.4.3	Project C	48
3.4.4	Project D	50
3.5	Threats to Validity	52
3.6	Discussion	52
3.7	Conclusions and Future Work	54
4	A Multiple Case Study of Coordination of Software Architecture Design in GSD	55
4.1	Introduction	55
4.2	Objectives	57
4.3	Related Work	57
4.3.1	The Role of Documentation	57
4.3.2	Software Architecture Understanding	59
4.3.3	Knowledge Transfer in GSD	59
4.4	Research Method	60
4.4.1	Data Collection	60
4.4.2	Interview Design	60

4.4.3	Data Analysis	62
4.4.4	Grounded Theory	62
4.4.5	Shared Mental Model Analysis	64
4.4.6	Case Studies	65
4.5	Case A	65
4.5.1	Case-Specific Problems and Generalizability	67
4.5.2	Architecture Development Process	67
4.5.3	Architecture Dissemination and Clarification Process	68
4.5.4	The Software Architecture Document	70
4.5.5	Architecture Compliance	73
4.5.6	Shared Mental Model Deviations	74
4.6	Case B	76
4.6.1	Case-Specific Problems and Generalizability	76
4.6.2	Architecture Development Process	77
4.6.3	Architecture Dissemination and Clarification Process	78
4.6.4	Software Architecture Document	81
4.6.5	Architecture Compliance	82
4.6.6	Shared Mental Model Deviations	83
4.7	Case C	83
4.7.1	Case-Specific Problems and Generalizability	84
4.7.2	Architecture Development	84
4.7.3	Architecture Dissemination and Clarification Process	86
4.7.4	Software Architecture Document	88
4.7.5	Architecture Compliance	88
4.7.6	Shared Mental Model Deviations	88
4.8	Conclusions and Future Work	89
4.8.1	How is Software Architecture Design and Dissemination Organized?	89
4.8.2	How Is Software Architecture Documentation Used?	89
4.8.3	What is the Role of the Architect in the Development Life Cycle ?	90
4.8.4	How is Architecture Compliance Organized?	90
5	A Theory of Coordination of Software Architecture Design in GSD	91
5.1	Introduction and Objectives	91
5.2	Data Collection	92
5.3	Theory Building	93
5.3.1	Cost Reduction	94
5.3.2	Knowledge Gap	96
5.3.3	Implementation Focus	96
5.4	Implications	97
5.5	Recommendations and Best Practices	97
5.5.1	General Recommendations	97

5.5.2	Best Practices	98
5.6	Conclusions and Future Work	101
6	Experimental Analysis of Representation of Software Architecture Design	103
6.1	Introduction	103
6.2	Related Work	104
6.2.1	Software Architecture Representation in Practice	104
6.2.2	Use of UML for Architectural Representations	105
6.2.3	Use of Design Documentation	106
6.2.4	Experimental Analysis of Software Design Representations	107
6.3	Objectives	108
6.4	Experimental Design	109
6.4.1	Experiment Planning	111
6.4.2	Data Collection Process	111
6.4.3	Material and Question design	113
6.4.4	Ordering Process	116
6.4.5	Data Coding	116
6.5	Results and Discussion	117
6.5.1	Media Effectiveness	118
6.5.2	Media Effectiveness for Topological Properties	122
6.5.3	Media Preference	122
6.5.4	Guesses and Suppositions	123
6.5.5	A Case Against Overlap	124
6.5.6	Conflicting Information	125
6.5.7	Participant Characteristics as Performance Predictors	125
6.6	Threats to validity	130
6.6.1	Internal Validity	130
6.6.2	External Validity	131
6.6.3	Conclusion Validity	132
6.7	Recommendations	132
6.8	Conclusions and Future Work	133
7	Contrasting Model-Driven Development with Code-Centric Development	135
7.1	Introduction	136
7.2	Objectives	136
7.3	Related Work	137
7.3.1	State of Empirical Research	137
7.3.2	Impact on Productivity	138
7.4	Case Study Design	138
7.4.1	Context	138
7.4.2	Specificities of the MDD Approach	139
7.4.3	Data collection	141

7.5	Results	141
7.5.1	Model Size	141
7.5.2	Model Complexity	142
7.5.3	Development Effort	144
7.5.4	Defects and Changes	152
7.5.5	Defect Discovery	153
7.6	Conclusions and Future Work	155
8	Analysis of the Consequences of Model-Driven Development for GSD	157
8.1	Introduction	158
8.2	Objectives and Data collection and Analysis Methods	158
8.3	Related Work	159
8.3.1	General Impact on the Software Architecture Process	160
8.3.2	MDD in Global Software Development	160
8.4	Results	162
8.4.1	Model-Centrism	164
8.4.2	Code Generation	166
8.4.3	Model Reuse	171
8.5	Impact of MDD on GSD	173
8.6	Conclusions and Future Work	174
9	Conclusions	177
9.1	Summary of Findings	177
9.1.1	RQ1: How is Software Architecture Represented, Disseminated and Coordinated in the Context of Global Software Development?	177
9.1.2	RQ2: How can we design software architecture documentation so that it is understood well by developers in the context of global software development?	178
9.1.3	RQ3: How does the application of model-driven development tools and techniques relate to the problems associated with global software development?	179
9.2	Contributions	180
9.3	Recommendations to Industry	181
9.4	Future Work	182
9.4.1	In-dept Evaluation of the Role of Documentation in GSD	183
9.4.2	Quantifying the Relation Between Developer Architecture Design Understanding and Software Quality	183
9.4.3	Facilitating industrial application of MDD	183
	Samenvatting	185
	Curriculum Vitae	189

Acknowledgments	191
List of Illustrations	193
Figures	193
Tables	194
Titles in the IPA Dissertation Series since 2006	197
Bibliography	205
Glossary	231