

# VU Research Portal

## Architectural Technical Debt: Identification and Management

Verdecchia, Roberto

2021

### **document version**

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

### **citation for published version (APA)**

Verdecchia, R. (2021). *Architectural Technical Debt: Identification and Management*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

### **E-mail address:**

[vuresearchportal.ub@vu.nl](mailto:vuresearchportal.ub@vu.nl)



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	2
1.2	Types of Technical Debt . . . . .	6
1.3	Overview of Technical Debt Research Trends . . . . .	8
1.3.1	Technical Debt Research Gaps . . . . .	9
1.3.2	Further Technical Debt Research Opportunities . . . . .	10
1.4	State of the Art Overview . . . . .	10
1.5	Architectural Technical Debt . . . . .	11
1.6	Research Goal and Research Questions . . . . .	13
1.7	Research Methodology . . . . .	15
1.8	Outline and Contribution . . . . .	16
1.8.1	Thesis at a Glance . . . . .	16
1.8.2	Authorship overview . . . . .	18
1.8.3	Other contributions . . . . .	19
<b>I</b>	<b>Architectural Technical Debt in Software-Intensive Systems</b>	<b>23</b>
<b>2</b>	<b>Architectural Technical Debt Identification: The Research Landscape</b>	<b>25</b>
2.1	Introduction . . . . .	28
2.2	Related work . . . . .	29
2.3	Study Design . . . . .	30
2.3.1	Research Goal . . . . .	30
2.3.2	Research Questions . . . . .	31
2.3.3	Search and selection . . . . .	31
2.3.4	Data Extraction . . . . .	34
2.3.5	Data Synthesis . . . . .	34
2.3.6	Study Replicability . . . . .	35

## Contents

---

2.4	Results - Publication trends (RQ1.1)	35
2.4.1	Publication year	35
2.4.2	Publication types	36
2.4.3	Publication Venues	36
2.5	Results - Research focus (RQ1.2)	37
2.5.1	Level of abstraction	38
2.5.2	ATDI Definition	38
2.5.3	Analysis Type	40
2.5.4	Analysis Input	42
2.5.5	Temporal Dimension	43
2.5.6	ATD Resolution	44
2.5.7	Tool Support	44
2.6	Results - Potential for Industrial adoption (RQ1.3)	46
2.6.1	Tool Availability	46
2.6.2	Industry Involvement	46
2.6.3	Rigor and Industrial Relevance	47
2.7	Threats to validity	50
2.8	Conclusions	51
<b>3</b>	<b>ATDx: An Architectural Technical Debt Index</b>	<b>53</b>
3.1	Introduction	56
3.2	The ATDx Approach	58
3.2.1	Definitions	58
3.2.2	ATDx Formalization	59
3.2.3	ATDx Building Steps	62
3.3	Empirical Evaluation Planning	67
3.3.1	Goal and Research Questions	67
3.3.2	Empirical Evaluation Design	68
3.4	Empirical Evaluation Execution	73
3.4.1	Phase 0: Selection of the SonarQube Tool	73
3.4.2	Phase 1: $AR^{SQ}$ Identification and Classification	73
3.4.3	Phase 2: Software Portfolio Identification	77
3.4.4	Phase 3: $AR^{SQ}$ Dataset Building	80
3.4.5	Phase 4: ATDx Analysis	81
3.4.6	Phase 5: Identification of Relevant Contributors	83
3.4.7	Phase 6: ATDx Report Generation	83
3.4.8	Phase 7: Report Distribution and Survey Invitation	85
3.4.9	Phase 8: Online Survey	85
3.5	Results	86

3.5.1	Participants Demographics . . . . .	86
3.5.2	(RQ1) On ATDx Representativeness . . . . .	86
3.5.3	(RQ2) On ATDx Actionability . . . . .	89
3.6	Discussion . . . . .	90
3.7	Threats to Validity . . . . .	94
3.7.1	Conclusion validity . . . . .	94
3.7.2	Internal validity . . . . .	95
3.7.3	Construct validity . . . . .	96
3.7.4	External validity . . . . .	97
3.8	Related Work . . . . .	97
3.9	Conclusions and Future Work . . . . .	99
<b>4</b>	<b>Architectural Technical Debt: A Grounded Theory</b>	<b>101</b>
4.1	Introduction . . . . .	103
4.2	Research Method . . . . .	105
4.2.1	Grounded Theory . . . . .	106
4.2.2	Grounded Theory Design and Execution . . . . .	109
4.2.3	Theory Evaluation via Focus Groups: Design and Execution . . . . .	114
4.3	A Theory of Architectural Technical Debt . . . . .	117
4.3.1	ATD Items . . . . .	121
4.3.2	Causes . . . . .	129
4.3.3	Consequences . . . . .	135
4.3.4	Symptoms . . . . .	140
4.3.5	Management Strategies . . . . .	147
4.3.6	Tool . . . . .	152
4.3.7	Artifact . . . . .	153
4.3.8	Prioritization Strategies . . . . .	154
4.3.9	Person . . . . .	155
4.3.10	Communication . . . . .	158
4.4	Related Work . . . . .	160
4.5	Theory Evaluation Results . . . . .	163
4.5.1	C1: Theory <i>Fit</i> to Underlying Data . . . . .	163
4.5.2	C2: Theory <i>Workability</i> . . . . .	164
4.5.3	C3: Theory <i>Relevance</i> . . . . .	164
4.5.4	C4: Theory <i>Modifiability</i> . . . . .	165
4.6	Verifiability and Threats to Validity . . . . .	165
4.7	Conclusion . . . . .	166

<b>II</b>	<b>Architectural Technical Debt in Android Applications</b>	<b>169</b>
<b>5</b>	<b>How Maintainability Issues of Android Apps Evolve</b>	<b>173</b>
5.1	Introduction . . . . .	176
5.2	Background . . . . .	178
5.3	Study Design . . . . .	179
5.3.1	Goal and Research Questions . . . . .	179
5.3.2	Context and Dataset . . . . .	180
5.3.3	Data Extraction . . . . .	185
5.3.4	Data Analysis . . . . .	187
5.4	Results . . . . .	190
5.4.1	RQ4.1. <i>Which are the most recurrent types of maintainability issues in Android apps?</i> . . . . .	190
5.4.2	RQ4.2. <i>How does the density of Android maintainability issues evolve over time?</i> . . . . .	191
5.4.3	RQ4.3. <i>What are the development activities in which maintainability hotspots occur?</i> . . . . .	197
5.5	Discussion . . . . .	199
5.5.1	Observations . . . . .	199
5.5.2	Best Practices for Android Developers . . . . .	203
5.6	Threats to Validity . . . . .	203
5.7	Related Work . . . . .	205
5.8	Conclusion and Future Work . . . . .	207
<b>6</b>	<b>Identifying Architectural Technical Debt in Android Applications through Compliance Checking</b>	<b>209</b>
6.1	Introduction . . . . .	212
6.2	Approach Overview . . . . .	213
6.2.1	Step 1: Android architecture guideline extraction . . . . .	213
6.2.2	Step 2: Android reference architecture establishment . . . . .	215
6.2.3	Step 3: Reverse engineering of implemented architecture . . . . .	215
6.2.4	Step 4: Compliance checking . . . . .	216
6.2.5	Step 5: Quantitative assessment of compliance violations . . . . .	216
6.3	Guidelines for architecting Android apps . . . . .	217
6.3.1	Study Design . . . . .	218
6.3.2	Research questions . . . . .	218
6.3.3	Research Method . . . . .	220
6.3.4	Results . . . . .	225
6.3.5	Threats to Validity . . . . .	238

6.3.6	Related Work . . . . .	240
6.3.7	Conclusions and Future Work . . . . .	242
<b>III</b>	<b>Conclusions</b>	<b>243</b>
<b>7</b>	<b>Discussion</b>	<b>245</b>
7.1	Research Questions Revisited . . . . .	245
7.2	Threads to Validity . . . . .	249
7.2.1	External Validity . . . . .	249
7.2.2	Internal Validity . . . . .	250
7.2.3	Construct Validity . . . . .	250
7.2.4	Conclusion validity . . . . .	250
7.3	Research Implications . . . . .	251
7.4	Replicability . . . . .	253
<b>8</b>	<b>Conclusions, Future Work, and Outlook</b>	<b>255</b>
	<b>Bibliography</b>	<b>281</b>