

Seminar Design Patterns und Anti-Patterns

ARCHITECTURE ANTI-PATTERNS

Arkadiusz Gabrys
Matrikel-Nr. 21895081
Department of Computer Science

July 10, 2015

Abstract — Content of this document describes three common Architectural Anti-Patterns. Sources and causes, symptoms and consequences, examples and solutions will be discussed.

Contents

1	Introduction	1
1.1	Software Architecture	1
1.2	Architecture Anti-Patterns	1
2	Stovepipe Enterprise	2
2.1	Form & Causes	2
2.2	Symptoms & Consequences	2
2.3	Example - authorization system	2
2.4	Solution	2
2.5	Exceptions	2
3	Stovepipe System	3
3.1	Form & Causes	3
3.2	Distinction	3
3.3	Symptoms & Consequences	3
3.4	Example	3
3.5	Solution	3
3.6	Exceptions	3
4	Vendor Lock-In	4
4.1	Form & Causes	4
4.2	Symptoms & Consequences	4
4.3	Example	4
4.4	Solution	4
4.5	Exceptions	4
5	Summary	5
	List of Figures	6
	References	7

1 Introduction

1.1 Software Architecture

Good architecture is a critical factor in process of transforming high level models into actual implementation [2]. Unfortunately the engineering discipline of software architecture is relatively immature and there is even no official definition [3]. SEI provides three types of software architecture definitions [4]:

- Modern definition – The set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both.
- Classic definitions – An architecture is the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization.
- Bibliographic definitions – Software architecture is the study of the large-scale structure and performance of software systems. Important aspects of a system's architecture include the division of functions among system modules, the means of communication between modules, and the representation of shared information.

The common aspect of those definitions is the placement of software architecture as a layer between high level abstract model and specific implementation.

1.2 Architecture Anti-Patterns

Anti-Pattern is a commonly used solution that results in bad consequences, describing the practice itself as well as solution [5]. Architecture Anti-Patterns are those patterns that occur at architecture level, so called system-level or enterprise-level [1].

In this document, while describing anti-pattern, firstly the general form of a bad solution will be presented both with it causes. Next the main symptoms and consequences and after that some real life example will be presented to describe example solution. In case of anti-patterns there occur some exceptions, situations when an anti-pattern can be accepted and that will be the last chapter for each one of them.

This whole project (excluding graphics) was created using L^AT_EX technology and its sources are available at https://github.com/gabr/dpap_aap

2 Stovepipe Enterprise

2.1 Form & Causes

2.2 Symptoms & Consequences

2.3 Example - authorization system

2.4 Solution

2.5 Exceptions

3 Stovepipe System

3.1 Form & Causes

3.2 Distinction

3.3 Symptoms & Consequences

3.4 Example

3.5 Solution

3.6 Exceptions

4 Vendor Lock-In

4.1 Form & Causes

4.2 Symptoms & Consequences

4.3 Example

4.4 Solution

4.5 Exceptions

5 Summary

List of Figures

References

- [1] Alexander Shvets *AntiPatterns The Survival Guide*, published by sourcemaking.com
- [2] *Nordic Workshop on Model Driven Engineering* ISBN: 978-91-7295-985-9, 27-29 August 2007 Ronneby, Sweden p. 5
- [3] *Architektura oprogramowania*, web: https://pl.wikipedia.org/wiki/Architektura_oprogramowania 10 July 2015
- [4] Software Engineering Institute, Carnegie Mellon University *What is your definition of software architecture?*, web: <http://www.sei.cmu.edu/architecture/start/glossary/definition-form.cfm> 10 July 2015
- [5] Brad Appleton *Patterns and Software: Essential Concepts and Terminology* 19 September 2008 12:06 AM p. 7