ANTWERPEN MANAGEMENT SCHOOL

Applying Normalized System Theorem on Restful APIs using the C#.NET programming language

Author:
Gerco Koks

Supervisors:
Prof. Dr. Ing. Hans Mulder
Frans Verstreken, Mcs

A thesis submitted in fulfillment of the requirements for the degree of Master of Enterprise IT Architecture

April 30, 2022

Declaration of Authorship

I, Gerco Koks, declare that this thesis titled, "Applying Normalized System Theorem on Restful APIs using the C#.NET programming language" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

Abstract

Gerco Koks

Applying Normalized System Theorem on Restful APIs using the C#.NET programming language

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

Physical Constants

Speed of Light $c_0 = 2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

For/Dedicated to/To my…

1 Introduction

"*Pantha Rhei*" is, according to *Plato*, one of the famous philosophical statements first described by the Greek philosopher *Heraclitus*.[1] Translated as "everything flows" this statement is an unambiguous commitment to ubiquitous dynamics of everything that exists. "Life is flux", one of the constants in life is change and its best we act accordingly.

In the realms of Software Engineering the "laws of software evolution" [2] refers to a series of laws described by Lehman starting from 1974. With these Laws, he describes the balance between the forces driving new developments on the one hand (a change), and the forces that slow down progress on the other hand. Based on *Heraclitus* philosophical statement we assume a software engineering project frequently will be subjected to change, possibly due to changing functional requirements and technological progress. As these changes emerges, the complexity of these software projects will gradually increase over time. If the system is not adapted appropriately the combinatorial effects of these changes will result in ever-increasing complexity and render the software system eventually obsolete, according to Lehman [2].

As the competitive environments of contemporary organizations are changing continuously, the speed at which changes follow each other is also increasing. IT organization are attempting to cope with this trend by adopting agility and maturing its agile practices [1]. Agility is defined as a measure for contemporary organizations to adept to new environments and to cope with rapid change [3].

The subjects discussed in previous paragraphs depict the current challenges of software evolvability

1.1 Problem statement

1.1.1 Normalized Systems Theorems

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

1.1.2 Clean Architecture

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

[1] https://plato.stanford.edu/entries/process-philosophy/

Companies that apply the Normalized Systems Theory research into their products are primarily using Java EE as a programming language. The company NSX for example has implemented their generation tools, modelling suite (Prime Radiant) and expander using this programming language. Java EE is still a very popular programming language for enterprise-, and IT organizations. Many software solutions are created and maintained using this programming language. The Normalized Systems Theorem is not only applicable to Java EE. The principles and design patterns that derive from the Normalized Systems Theorem are in fact applicable for any object-oriented programming languages.

Another example of a popular programming language in enterprise-, and IT organizations is C#. There is however no documented research, or proof of experiences on C# software projects using Normalized Systems Theory with the aspects of integration, expansion and rejuvenation.

1.2 Research questions

2 Theoretical background

2.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

2.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

2.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

2.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

3 Research methodology

3.1 Research model

3.2 Research approach

3.2.1 Literature study

3.2.2 Interviews

3.2.3 Expert group

A Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

[1] Leon Kappelman et al. “The 2014 SIM IT Key Issues and Trends Study: Appendix”. In: *MIS QUARTERLY EXECUTIVE* 13.4 (Dec. 2014). ISSN: 1540-1960.

[2] M.M. Lehman. “Programs, Life Cycles, and Laws of Software Evolution”. In: *Proceedings of the IEEE* 68.9 (1980), pp. 1060–1076. ISSN: 0018-9219. DOI: 10.1109/PROC.1980.11805. URL: http://ieeexplore.ieee.org/document/1456074/ (visited on 04/25/2022).

[3] Seev Neumann. *Strategic Information Systems: Competition through Information Technologies*. New York : Toronto : New York: Macmillan College Publishing Co. ; Maxwell Macmillan Canada ; Maxwell Macmillan International, 1994. 258 pp. ISBN: 978-0-02-386690-6.