

Slovak University of Technology in Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Tomáš Frtala

Dissertation Thesis Abstract

Improving Comprehensibility of Organizational Patterns by Animation, Scenarios, and Variants

to obtain the Academic Title of *Philosophiae Doctor (PhD.)*

Degree course:	Intelligent Information Systems
Field of study:	Computer Science
Form of study:	External
Place of development:	Institute of Informatics, Information Systems and Software Engineering, FIIT STU BA

Bratislava 2022

Dissertation Thesis has been developed at the Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Submitter: Ing. Tomáš Frtala
Institute of Informatics, Information Systems and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava

Supervisor: doc. Ing. Valentino Vranić, Ph.D., Assoc. Prof.
Institute of Informatics, Information Systems and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava

Oponents: doc. Ing. Miroslav Bureš, Ph.D., Assoc. Prof.
Dept. of Computer Science, FEE
Czech Technical University in Prague

MSc. Marko Ranković, Ph.D., Assoc. Prof.
Faculty of Information Technology and Engineering
University Union-Nikola Tesla in Belgrade

Dissertation Thesis Abstract was sent:

Dissertation Thesis Defence will be held on at pm at the Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava (Ilkovičova 2, Bratislava).

Prof. Ing. Ivan Kotuliak, PhD.
Dean of FIIT STU in Bratislava

ABSTRAKT

Organizačné vzory sú kľúčom k postupnému prijatiu agilných a lean prístupov a postupného rastu agilnej a lean spôsobu organizovania práce. Vzory môžu byť použité či už na riešenie konkrétnych problémov v rámci organizácie, alebo na vybudovanie novej formy organizácie.

Hoc výskumníci sa zaoberali problematikou agilných metód a lean vývoja softvéru, a tiež vzormi a jazykmi vzorov, málokto však skúmal a hľadal spôsob, ako môžu organizačné vzory pomôcť zlepšiť agilný a lean vývoj softvéru. Účinné využívanie organizačných vzorov môže pomôcť pri identifikovaní najlepšieho možného riešenia pre zmiernenie problémov v neustále sa meniacich dynamických trhových podmienok a zlepšenie procesu vývoja softvéru. Okrem toho zvládnutie správneho rozpoznávania protichodných problémových síl v kontexte a spôsob ich zmiernenia vhodným uplatňovaním organizačných vzorov, môže viesť k lepšiemu strategickému rozhodovaniu vo vývoji softvéru.

V tejto práci predstavujeme niekoľko prístupov, nových metód a navrhnutých zmeny s cieľom zlepšiť pochopenie organizačných vzorov:

- Prístup animovania organizačných vzorov. Zapojením účastníkov do zažitia činností prostredníctvom rôznych rolí cez série scén v textových dobrodružných hrách. Pôvodné situácie sú modelované s použitím rôznych nástrojov na vyjadrenie podstaty vzoru vo viacerých fázach—každá vytvárajúca cenný artefakt pre účastníka.
- Prístup modelovania zmien vzoru. Pôvodne opísané situácie možno vnímať aj inak a účastníci ich môžu uplatňovať s malými zmenami, ktoré vytvárajú varianty vo forme, vzťahoch a scenároch. Účastníci môžu sledovať, manažovať a zlepšovať vývoj zacytením variantov organizačných štruktúr.
- Prístup merania agilnosti pomocou novej metriky. Môže sa použiť ako vstupný bod pre analýzu aktuálneho stavu organizácie. Alternatívne môže analyzovať rozsah uplatneného rozhodnutia a jeho dôsledky na organizáciu.

Každý prístup bol vyhodnotený s použitím netriviálnych príkladov vrátane zapojenia externých spoločností. Okrem toho, prezentujeme aj reálne situácie, ako možno všetky tieto prístupy môžu byť uplatnené či už samostatne alebo spoločne v synergií.

ABSTRACT

Organizational patterns are the key to a stepwise adoption of agile and lean approaches and to a piecemeal growth of agile and lean organization of work. They can be applied to correct particular problems within an organization or to build a new organization.

Although there has been much research done on the agile methodologies, and lean software development, and in particular about patterns and pattern languages, little work has been done in finding how organizational patterns can enhance agile and lean software development. Using organizational patterns efficiently can help with identifying the best possible solution for tempering down problems of the ever-changing dynamic market conditions and enhancing the software development process. Furthermore, by mastering the correct recognition of contradictory problem forces in a context and how to mitigate them by appropriately applying organizational patterns can lead to improved strategic decision-making in software development.

In this thesis, we present several approaches and novel methods, proposed various changes in order to improve organizational patterns comprehension:

- An approach to animating organizational patterns. By taking participants to experience actions from different roles by passing through a series of scenes in the textual adventure games. Original situations are modeled using various tools to express the essence of a pattern at multiple stages—each producing a valuable artifact for a participant.
- An approach to modeling pattern variations. Originally described situations can be perceived differently, and participants may apply them with small changes that generate variants in their form and relationships and a scenarios. Participants can track, manage, and enhance evolution by capturing variants of organizational patterns.
- An approach to measuring agility using a novel metric. It can be used as an entry point for analyzing an organization's current state. Alternatively, it can analyze the extent of applied decision and their consequences on the organization.

Each approach was evaluated using nontrivial examples also including the involvement of external companies. Furthermore, a real-life situations are shown to illustrate how all these approaches can be applied either standalone or together in synergy.

Table of Contents

1	Introduction	1
1.1	Thesis Statement	1
1.2	Approaches Proposed in This Thesis	2
2	Capturing Experience	3
2.1	Organizational Patterns	3
2.1.1	Structure of a Pattern	4
2.1.2	Relationships Between Patterns	4
2.2	Comprehension of Pattern	5
3	Animating Organizational Patterns	6
3.1	Serious Games and Importance of Scenario	6
3.2	Power of Imagination	7
3.3	Organizational Pattern as Text Adventure Game	7
3.4	Evaluation	8
3.5	Results	10
4	Improving Organizational Pattern Comprehensibility by Varying Their Form and Relationships	12
4.1	Types of Variations	12
4.2	Community Web Portal	13
4.3	Evaluation	14
4.4	Results	14
5	Measuring and Improving Agility with Organizational Patterns	16
5.1	Organizational Patterns as Agile Metrics	16
5.2	Definition of Metrics	17
5.3	Evaluation	18
5.4	Results	18
6	Conclusions	20

References

22

A Publications

A-1

1 | Introduction

The popularity of agile and lean approaches to software development in general—and Scrum in particular—is still on the rising curve for their flexibility and effectiveness while being close to the natural human attitude towards work [28].

The work of Christopher Alexander Pattern Language [2] caused revolution at perceive, express and chronicle the nameless qualities. His reflections and manners has influenced not only architectural engineering, but also software engineering. Patterns from his book inspired also game designer Will Wright to adapt its principles like a models for simulation game SimCity [3].

Inspired by the work of Christopher Alexander, Coplien and Harrison published Organizational Patterns of Agile Software Development [5], which brings a different perspective on agility in organizations. Organizational patterns are the key to a stepwise adoption of agile and lean approaches and to a piecemeal growth of agile and lean organization of work. However, their text description is not easy to comprehend. When applying organizational patterns in an organization, we can face difficulties related to the exerted convention, mindset of people, and overall firm culture, which is also the case of applying agile methods [15].

However, as Zdun [30] claims, applying patterns also requires a certain amount of gained expertise. It is necessary to understand how the solution presented in the pattern fits the overall system and what consequences this pattern can bring to the system. Finally, to know what new possible problems are created, which will probably be resolved by other patterns.

1.1 Thesis Statement

Challenges in the management of software development projects require perfect collaboration from the participants in order to be successful and profitable. Applying organizational patterns may minimize some problems by addressing their contradictory forces. But making the right decisions during the software development

on how, when, and what organizational pattern to use for solving problems can sometimes be demanding on participants because they may not have the necessary knowledge or skills at a specific time in a particular situation. Decisions—good or bad—affect involved participants across the various levels, but their consequences also affect other participants and the resulting product. Therefore, the thesis statement is as follows:

Exposing the practitioners to effective examples of the fundamental situations on which organizational patterns and their variants are based will improve the comprehensibility of the organizational patterns.

1.2 Approaches Proposed in This Thesis

According to design science research practice [24], we constructed and assessed several complementary approaches for addressing the thesis statement. To be able to use organizational patterns efficiently, we have to ensure that participants master organizational patterns. Therefore we:

- **Animating organizational patterns** - although patterns capture tested and verified solutions to problems based on previous experiences, the original text is extensively described, and participants may find it hard to get the essence of a pattern. An organizational pattern is transformed into text adventure games using NLP processing with force field analysis to visualize contradictory forces. The scenario is modeled using a UML state machine diagram. Final text descriptions intended to display for participants are modeled using a linguistic causal modeling process.
- **Modeling variants of organizational patterns** - Coplien [5] also confirms that since organizational patterns were written as catalog, it does not mean that they can not change in the future. Originally described situations can be perceived differently, and participants may apply them with small changes that generate variants in their form and relationships and a scenarios. Participants can track, manage, and enhance their evolution by capturing variants of organizational patterns.
- **Measuring and improving agility with organizational patterns** - based on our research about existing agile metrics, which we found unsatisfactory, we created a custom metric that is not only useful in comparison with other metrics but also can be used for evaluating our approach.

2 | Capturing Experience

It is essential to realize that agile and lean are actually not new, nor did they appear suddenly. On the contrary, they are based on recurring organizational patterns that have been around long before they have been qualified as agile or lean. As Coplien and Harrison [5] showed, these *organizational patterns* can be identified in successful projects and presented in a concise, written form in a similar manner as design patterns in software development [10] or patterns in building architecture, where this idea was first proposed by Alexander [1].

However, despite vast efforts, there have been recurring difficulties in the adoption of agile and lean approaches. As Vijayarathy [25] confirms, they are even being rejected at different levels of organizational structure, including developers themselves. This is paradoxical, since agile and lean bring more freedom and involvement.

Part of the reasons probably lies in not providing practitioners, who often have been conditioned (i.e., trained) to following heavyweight development processes, with an opportunity to *experience*—and not merely *understand*—what agile and lean are.

In software engineering, sociologist Melvin Conway observed interesting interconnection between the shape of software artifacts and the organizational structure of a company. The result of his research claims followings:

Organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations. [4]

2.1 Organizational Patterns

Organizational patterns are the key to a stepwise adoption of agile and lean approaches and to a piecemeal growth of agile and lean organization of work. They can be applied to correct particular problems within an organization or to build a new organization. Scrum has been expressed in terms of Coplien and Harrison's

organizational patterns [20], and there was even an ongoing work to define Scrum-specific patterns [23] which just recently finished [22].

Considering that these patterns are applicable through many roles (not merely tied with software development) in a enterprise, sometimes arise problem when one wants to apply these patterns without enough authority.

The primary key element for successful adoption of agile and lean approaches seems to change not only the organizing of developers but also the management.

2.1.1 Structure of a Pattern

Coplien and Harrison, as a basis for the visual and structural representation of an organizational pattern, used Alexander's pattern form [2]. It comprises text descriptions that are split into three parts each divided by three symbols of an asterisk

Title *

Context

Problem with accompanying forces

Solution

Consequences

The content (text description) of patterns according to Coplien was built as follows:

Many of these patterns are derived from stories we have picked up throughout our travels — stories of real problems in real organizations and the real solutions those organizations applied. Many of our patterns start with what we believe to be a particularly poignant or appropriate selection from among such stories. [5, Chapter 3.1.3, p. 25]

2.1.2 Relationships Between Patterns

By applying an organizational pattern the original problem and their contradictory forces may diminish or dissolve. There can be already applied other organizational patterns which probably solve similar problem so they may be contextually closer. As was previously mentioned in Section 2.1.1, patterns can indicate their relation to other patterns in the context part or consequences part. It is possible to apply patterns standalone, but some patterns may work well in conjunction with others and multiply their benefit. Patterns can be composed into an arrangement that analogically resembles language and can be used similarly by parts or units. The arrangement of patterns into a pattern language can tell a reader how one can build a whole, a system.

2.2 Comprehension of Pattern

Alexander's [2] pattern language is intended to be used with the architecture of the real material world. In such an environment, the context consists of buildings, places, people, habits, and situations. These aspects are well recognizable because they are linked to physical objects (mostly static) and perceived by simple observation. As a result of previous, context identification is more manageable, and finding potential solutions for specific problems can be well imagined.

Likewise, but intended to use in the non-material virtual world of software are Gamma et al. [10] design patterns. The context consists of relationships between classes, objects, attributes, and situations. These aspects are directly observable and moreover, they are static (do not have moods, emotions or needs).

On the contrary, the organizational patterns are applied in a dynamic social environment where it is sometimes hard to identify a context and its forces through observing a behavior. Therefore, the assumption is that the context will be similar to reality, ideally the same or at the most minimal dissimilarity, in order to successfully repeat organizational patterns application.

Furthermore, the text description of the pattern needs to be designed in such a way that it can immerse the reader into the situation and support comprehension.

Therefore one of our primary concerns is how to improve comprehension of organizational patterns. Comprehension—not simple learning or acceptance—is achieved by experience, which is typical of human nature.

As Coplien [5, Chapter 2.1.3, p. 15] confirms, one way to gain experiences without risking failure in a real setting is to employ sociodrama in which participants can play the situation from the perspective of different roles. However, people may not be willing to participate due to shyness or discomfort from conflicts that, though played, do arise and may evoke strong feelings among participants. Also, sociodrama can be time-consuming and moreover requires the time coordination of many people.

3 | Animating Organizational Patterns

It is really hard to cut through all those forces, principles, and situations by which organizational patterns are typically described. It is even harder if you are to apply what you read in real life. Human nature requires experience. One way to gain it without risking failure in a real setting is to employ sociodrama in which participants can play the situation from the perspective of different roles. Such example of using sociodrama approach can be found in research by Vranić et al. [13, 26, 27]. Suppose participants are enthusiastic and assimilate enough with their new role. In that case, a good role-play in sociodrama can bring up the identification of the core principles accompanied by the “aha” effect. However, people may be not willing to participate due to shyness or discomfort from conflicts that, though played, do arise and may evoke strong feelings among participants. Sometimes the sociodrama can bring to the surface negative things, and persons are confronted with this kind of feeling among other participations that result in discomfort. Also, sociodrama is time consuming and moreover requires time coordination of many people.

Anyone who experienced developing software remembers feelings and memories associated with particular moments in situations. These feelings and memories are saved in our subconsciousness and we remember them longer in compare with for example read text. Feelings are tied to specific moments and when we experienced similar situation as before we tend to act near just like in the memorized situation.

3.1 Serious Games and Importance of Scenario

One solution to the problems of employing sociodrama is offered by virtual games where the person can act without feeling discomfort and, as a bonus, can be involved anytime in any place. Using the game format is not new and has been applied in learning software engineering under limited conditions [7], agile software development [19], or lean approach [14, 16, 21, 31], in addition also by using virtual reality [9]. This type of game is known as serious games and takes part in various non-entertainment sectors, whether it is for learning or training. In order to motivate users to be more active,

serious games often borrow fun elements from entertainment games; this technique is called gamification [8].

Although the technical quality of the game is essential, in serious games, the scenario is crucial to the perception of the situation by the user [11, 12]. If scenario is involving, players easily identify with the main character, as we all know from popular games, despite simple graphics (recall Pacman) or no graphics at all: text adventure games.

When mimicking sociodrama by a serious game, the goal is to make players feel the forces driving the situation in the underlying organizational pattern as if they have been exposed to them. The scenario should sufficiently emphasize moments specific to that situation to achieve this.

3.2 Power of Imagination

We are dealing with software development professionals or people that—albeit themselves do not participate in programming on a daily basis—have a good idea of how software is developed and have been exposed to real critical situations that arise during software development. Using right words but avoiding binding to any specific programming languages or other notations, one can create a general setting capable of immersing a person being conditioned to software development process by simply being exposed to the respective environment into a desired fictional situation applicable to such environment.

The power of imagination is huge and can substitute other perceptions. For example, if someone says to you: “Your program suddenly stopped functioning after you implemented that new big algorithm,” your mind employs the imagination and your memories and unfolds this condensed description into a specific situation that comprises also your feelings related to such situation. Thus, despite the description mentions no programming language, you instantly fill in your own. Accordingly, you gain a feeling of being in specific programming context, e.g. database development, mobile application development, etc.

3.3 Organizational Pattern as Text Adventure Game

To ease their mastering, we propose to represent organizational patterns as adventure games. An adventure game is a long-time known type of video games based on an involving story rather than action (as opposed to arcade games). The game consists of a number of scenes and the player as a protagonist is supposed to successfully solve the tasks in each scene in order to finish the game.

The simplest form of realization of adventure games—known as text adventure games—involves only textual scene description and command line interface. This does not necessarily constitute a disadvantage as the stress is put on the game idea and puzzling tasks that often span over several scenes. Considering our audience to be highly educated, we assume that the textual description would be appropriate for organizational pattern adventure games. Moreover, as can be experienced from literary works, omitting direct graphical or video representation provides an opportunity to readers to develop their own representation they can more easily identify with while still maintaining conformance with the writer’s intent in essential aspects.

The following sections present the proof of concept for our approach as an adventure game for the ARCHITECT ALSO IMPLEMENTS organizational pattern [5].

This pattern puts the architect’s role in a direct and active contact with software artifacts. This keeps architecture within the reality of implementation while providing an opportunity for “ordinary” developers to take part in it.

There are two key roles in this pattern: architect and developer. It should be possible to play the game from the perspective of each role separately. Consider the UML state machine diagram. States represent scenes, each of which is presented to the player in the form of text description.

Transitions between scenes are determined by triggers and guards (introduced in square brackets as common). Triggers are expressed by events that represent conscious decisions from the perspective of player’s role, e.g. *Adapt the architecture document*. After reading the scene description, the player chooses one of the events.

Guards represent conditions beyond the player’s role perspective. These may come from the environment (*Problems have been found*) or other roles (*The architect joins you*). The game could be configured to allow the player to choose among the guards in the same manner as with events, to let the game make a random choice, or to always take the default choice (without displaying the guards at all). Default guards are marked by the «default» stereotype.

In case no conscious decisions is available to the player, the generic *Continue* event is used. This means the player just continues to the next scene. Of course, if there are guards on a transition with the *Continue* event, the transition is determined by the guards.

3.4 Evaluation

We tried to evaluate our approach on the broad spectrum of stakeholders (students, managers, architects, developers, and testers) participating in the software development

process.

Firstly, we designed the questionnaire to compare before and after state effects directly. Results of its comparison can bring the primal confirmation or rejection our hypothesis. Secondly, we will use an inductive approach and design a questionnaire that can help draw theoretical and general conclusions. Therefore we designed an additional questionnaire to support inductive reasoning. Finally, the result of comparison with the conjunction of acquired theoretical conclusions will be used to deduce the final judgment of our hypothesis.

Questionnaire for Measuring by Comparison. Scale between -5 is worst and +5 is best whereas 0 is neutral.

Research Question 1: *Expression* How is the form of expressing organizational patterns?

Research Question 2: *Comprehension* Do you identify similar situations in reality, like in organizational patterns?

Research Question 3: *Agile and Lean* How does your agile and lean organization changed?

Questionnaire for Inductive Reasoning .

Scale between 5 is strongly disagree and 1 is strong agree whereas 3 is neutral.

Topic 1: Changing expression of original organizational patterns. Although we chose animation as the active form for gaining experience, we would like the level of comprehension of the passive form (original book) before they applied our solution.

1. Should be the original organizational pattern written in a more concise structure?
2. Should be the original organizational patterns supported by some diagrams?
3. The original text descriptions of organizational pattern involved you in the scenario?
4. Did you find yourself in similar situations in reality?
5. Were you confused by interconnections of organizational patterns?

Topic 2: The extent of improvement on organizational pattern comprehension. After our approach was applied, we would like to know how good or bad our solution was in specific parts.

1. The animated scenario evoked in you similar experiences from reality?
2. The text descriptions of animated scenes involved you in the scenario?
3. The text descriptions of animated scenes were understandable?
4. Were you confused by interconnections between animated organizational patterns?
5. Would you carry out similar actions in your role as in the animated organizational pattern?
6. Would you carry out similar actions when you know what effect you caused on a different role in the animated organizational pattern?

3.5 Results

By direct **comparison** of before and after results, we were trying to observe what implications of applying animated organizational patterns were. The results from the direct comparison (Figure 3.1) showed confirmed significant improvements in all three measured variables when using animated organizational patterns compared to using an only book.

Furthermore, we wanted to capture user experiences after our study in order to draw a general conclusion about using organizational patterns. Therefore we used **induction** and designed an additional questionnaire that covered two topics. The results (Figure 3.2) indicate that the animated version of organizational patterns mitigates specific concerns about the comprehension and applicability of organizational patterns.

Finally, from the direct comparison results, we observed significant improvements by introducing animated organizational patterns. Complementary to these results, findings by induction confirm that animated organizational patterns mitigated specific concerns about organizational patterns. Therefore from the results of each part of the evaluation, we are deducing a **deduction** that our chosen approach to animating organizational patterns supports correctness that by changing the expression of organizational patterns, it is possible to improve the extent of their comprehension of a broader spectrum of stakeholders.

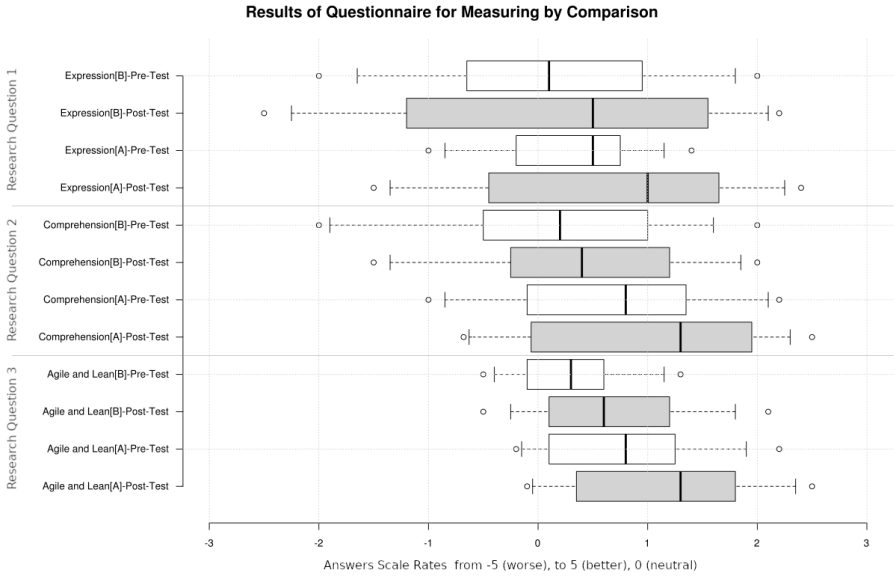
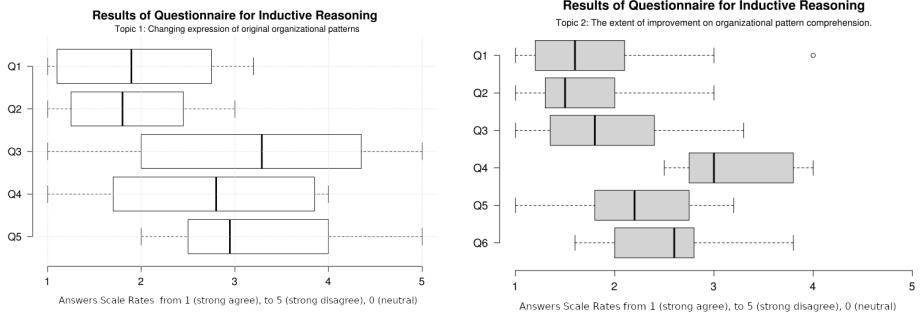


Figure 3.1: Assessment of questionnaire from the direct comparison between two groups which used [B]-book, [A]-animated.



(a) Topic 1 both groups participated book/animated.

(b) Topic 2 only the control groups participated.

Figure 3.2: Assessment of questionnaire for inductive reasoning.

4 | Improving Organizational Pattern Comprehensibility by Varying Their Form and Relationships

The patterns as such have a particular shape and properties that are affected by the environment in which they are used. One of the apparent variations—that can be perceived at first glance—lies in a language mutation or variation based on the culture of a country. For our purposes we used only English language. However, there can be other factors that we can take into consideration.

4.1 Types of Variations

Pattern Forms It is imperative for the reader to understand the context of a problem which is given in some pattern. Also, the reader must be able to distinguish the correct pattern for solving their problem from a large number of patterns. The quality of a pattern description depends on the author writing skills.

However, many pattern forms consist of different structures (pattern elements) that describe various properties of a pattern. Table 4.1 presents various identified pattern forms that are usually used.

Alexandrian form	Canonical form	Coplien form	GoF form	Compact form	Portland form	Beck form	Fowler form
Title	Name	Name '...	Name	Name	Name	Title	Title
	Alias (optional)	Alias (optional)	Also Known As				
Context	Problem	Context	Intent	Context	Context	Context (optional)	
***	Context	***	Applicability				
Problem description	Forces	Problem discuss forces	Motivation	Problem		Problem	Summary of the pattern
Therefore:	Solution	Therefore:	Participants, Structure, Collaborations, Implementation	Forces	Therefore,	Forces	
Solution	Example (optional)	Solution	Sample Code	Solution	Solution	Solution	Discussion
***	Resulting Context	***			But,		
	Rationale (optional)	Consequences	Consequences	Resulting Context	Consequences	Resulting Context	
	Known Uses		Known Uses				
Related patterns	Related Patterns	Related patterns	Related Patterns				

Table 4.1: Pattern Forms

Pattern Elements Patterns can be made more abstract and viewed from a larger perspective so that their applicability increases significantly. As Wieck [29] confirms,

users can apply solutions differently from those encountered (and perhaps even intended) by the pattern author.

This approach is not necessarily wrong. On the contrary, it can help to create new, better solutions to some specific issues. In this way, new variations of patterns and new patterns emerge. Variations arise naturally due to the inhomogeneous environment where the problems and the authors of the solutions themselves find themselves. Variability is greatly affected by forces limitations related to the problem.

Pattern Language A pattern language is created explicitly in the pattern as such. Different links between patterns can produce distinctive paths of an original pattern language. Since contexts link patterns together into a pattern language, altering pattern context will affect the operation of the pattern, for example, specifying the size, scope, place of use, or programming language.

4.2 Community Web Portal

One of the earliest attempts to gather the information about various known types of patterns is documented in the community wiki-like portal named the Portland Pattern Repository [17]. Its pages are dedicated to the topics like pattern writing guiding, a repository of various types of patterns, and pattern form [18].

Based on the Portland Pattern Repository, we created an auxiliary tool in the form of a simple web portal for allowing us to gather data about variations. Users can search and display details about a pattern in pattern languages. To support variability, the web portal offers common CRUD (create-read-update-delete) operations with patterns. Also, the form the patterns are displayed in can be switched. We choose the canonical form as the basic pattern form because other pattern forms are derived from it.

Users can create, edit, and compare variations at any level. Variability can be represented from a linguistic point of view (textual) and the point of view of diagrams (in models, i.e., pattern interconnections in a pattern language). To express variability visually, changes are marked by colors in their commonly used meaning: *red*—removed, *green*—added, and *blue*—modified. The detection of textual variants has been performed by simple diff based on granularity. The detection of diagram variants has been performed by comparing pattern language maps, which holds relationships between patterns.

Transforming originally entered pattern from a less structural pattern form can be problematic. To support transforming patterns into more structural pattern form, we use the text extraction summarization technique with the help of natural language processing (NLP).

4.3 Evaluation

Five teams from two different companies participated in our survey. All teams operate in the IT sector and are mainly focused on product development.

Each team received its own web portal (described in Section 4.2) as docker container. The initial setup of database contained almost 50 predefined organizational patterns in their corresponding pattern language (originating from the book on organizational patterns [5]).

User interaction like creating, editing, deleting and inserting were recorded. Users were instructed to use the web portal as the catalog of organizational patterns when they would need to look for some pattern during their normal everyday software development. Users can also discuss patterns, and if they feel the need for making changes to a pattern, they can create a variation to it. This also applies to pattern language. When users need to adjust some pattern sequences or relationships in a pattern language, they are free to do so. Finally, after more than two months of interacting with the web portal, we stopped the experiment and collected the data.

4.4 Results

The evaluation results showed various interpolations that we further describe in the following subsections.

Pattern Form Variants We wanted to know which pattern form users mostly interacted with by creating new variants. Figure 4.1 shows that the pattern form the participants interacted with most was the compact form.

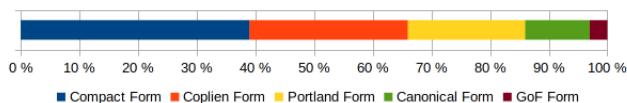


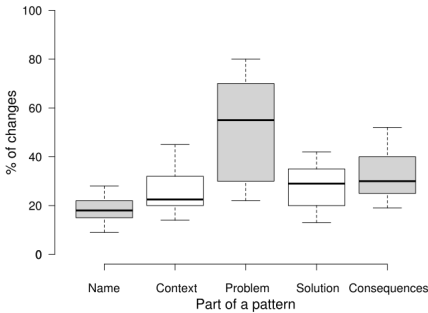
Figure 4.1: Statistics of the most frequently modified forms.

However, the results show several other exciting things. Users tend to create variants by interacting with a more structural pattern form. Contrary to previous findings, the extensive structural pattern forms were used far less.

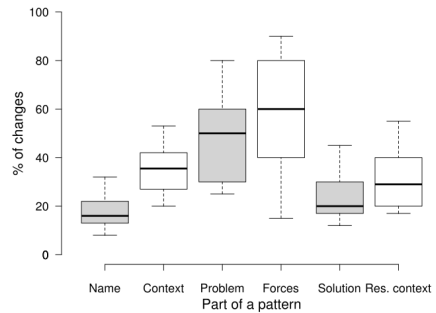
Pattern Element Variants Next, we focused on finding which specific parts of a pattern form mainly were modified. We take into consideration the comparison of the

two most used pattern forms.

As shown in Figure 4.2a in the Coplien form, most users modified the problem part, which generated most variants of the pattern.



(a) Most modified parts of the Coplien form.



(b) Most modified parts of the Compact form.

When a user transforms an organizational pattern from the Coplien form into the compact form, the text processing dissects the problem part into separate parts (the problem statement and forces). As it can be seen from Figure 4.2b, after transforming, users continue with fine-tuning both parts. However, we noticed that the part of the forces is prone to create other possible variants.

In both cases, some parts of pattern elements are prone to produce more pattern variants. Others—like name, context, solution, and consequences—share common similarities and therefore are less prone to produce new pattern variants.

Pattern Language Variances The participants were conservative in the case of changes to links between patterns, and only about 10% of connections between organizational patterns changed among all pattern languages.

5 | Measuring and Improving Agility with Organizational Patterns

The famous proverb that is usually associated with management theorist Peter Drucker says: “*what gets measured gets managed*,” is traditionally used by managers and executives to make decisions. One of the prevalent concerns about agile software development methods is how effective they are and to what degree they can reach changes, and what are the outcomes of changes. These types of concerns stem from not having an appropriate style of management which in turn affects the organizing of teams and their performance.

5.1 Organizational Patterns as Agile Metrics

However, existing agility metrics do not evaluate sufficiently how agile a team is. They can provide us with visibility of agility at various levels of the software development process. Still, they are not sufficient to measure agility at the higher levels of the organization.

By contrast, organizational patterns cover all layers of an organization’s structure, with the focus on the highest levels. Additionally, as in the book by Coplien et al. [6], the authors state that Organizational Patterns [5] has laid the foundations for the emergence of agile methodologies.

Considering that an organizational pattern captured solutions to problems from experienced situations, we can assume that these situations can occur in the software development process repeatably. Therefore the presence of such situations—which are currently trying somewhat to solve contradictory problem forces—resembles the principle of the organizational pattern. When the current setup manages the problems and gets them under control, we can assess if the adopted solution resembles a solution from an organizational pattern.

It is possible to state the following:

Suppose that each applied organizational pattern increases an organization's agility. In that case, we can evaluate the extent to which the organization is agile by the degree of occurrence of identified applied organizational patterns.

5.2 Definition of Metrics

Each organizational pattern language can represent one specific agility metric. The individual questions in the questionnaire will be based on the organizational patterns that are in a pattern language. According to the answer to the question, the system will evaluate the degree of compliance with the current processing organizational pattern. As for the basis for the formulation of questions will be the patlets—minimalized versions of organizational patterns compressed using *If/Then* construction. From these patlets we will reuse the *if* condition as a form of visual presentation for questions that the system will offer to participants.

Primarily, the metric operates with all the organizational patterns. However, the appropriateness of organizational patterns is assessed during the measurement, which influences the maximum number of applicable organizational patterns. As shown in Equation 1, let's define M as the definition of the measured results, and its value will consist of the ratio of achieved and the maximum result expressed in percentage.

Let's define M_a as the achieved measured result. Its value is equal to the sum of not skipped answers after multiplying them by the weighting factor of the question.

Let's define M_m as the maximum possible result. Its value is equal to the sum of the maximum values of the answers to the non-skipped questions after multiplying by the weighting factor of the given question.

As said previously, the result of the metric measurement will therefore be presented both as a percentage of success and as an absolute number of points. An example of such a result can be seen in Equation 2.

It is possible that a system could also visualize the results, both in the context of a specific outcome and in the context of a schematic representation of the pattern language used as a metric.

$$M = \frac{M_a}{M_m} [\%] \quad (1)$$

$$M = \frac{30}{37} (\simeq 81\%) \quad (2)$$

5.3 Evaluation

It is necessary to evaluate whether the created metric is relevant and whether participants can use it to measure agility, comparing and contrasting the measured values with a suitably selected existing and verified agile metric.

The Boehm-Turner model was chosen as the relevant control metric for its similarity to the newly created metric.

To evaluate the newly created metrics and system, ten teams from 7 different organizations cooperated. Participants used the self-assessment method for a questionnaire (measuring via the new metric) and the Boehm-Turner model.

5.4 Results

The results of the evaluation comparing the new agile metric are shown in Table 5.1.

Team	New Agile Metric			Boehm-Tuner model					
	M_a	M_m	M	Imp	Size	Cul	Dyn	Per	Sum
1	29,46	34,50	85,43 %	42	52	42	79	42	257
2	2,00	33,50	5,97 %	80	88	0	0	0	168
3	19,00	35,00	54,29 %	62	68	80	80	24	314
4	10,28	30,5	33,70 %	22	55	4	6	25	112
5	25,50	25,50	100,00 %	62	80	80	62	100	384
6	18,00	29,00	62,07 %	89	71	31	22	62	275
7	18,58	36,00	51,60 %	62	60	62	30	22	236
8	10,48	37,00	28,31 %	42	66	31	12	12	163
9	3,75	35,00	10,71 %	42	77	42	4	4	169
10	9,88	30,00	32,92 %	42	51	22	22	22	159

Table 5.1: Results of the second phase of evaluation

Correlations We used Pearson's correlation coefficient to evaluate the results of the new metric evaluation against Boehm-Turner model. We calculated the correlation between the measured percentage value of the questionnaire and the individual factors of the Boehm-Turner model, as well as the sum of these factors.

According to the results of the conducted evaluation, we get the following correlations between the results of the questionnaire and the individual factors of the Boehm-Turner model:

1. *Importance*—the resulting correlation is 0.163. This value represents almost no correlation, which aligns with our expectations.
2. *Size*—the resulting correlation is -0.265 . This value represents a very low negative correlation, which aligns with our expectations.
3. *Culture*—the resulting correlation is 0.716. This value represents a slightly to slightly high correlation, which is slightly inconsistent with our expectations, as a large agreement was expected.
4. *Dynamism*—the resulting correlation is 0.917. This value represents a very high correlation, which is inconsistent with our expectations, as a small to medium match was expected.
5. *Personnel*—the resulting correlation is 0.899. This value represents a high correlation, which aligns with our expectations.

Sum of factors—the resulting correlation is 0.902 which represents a very high correlation.

Based on the calculated correlations of the newly created metric, mainly with the sum of factors of the Boehm-Turner model, we can declare that the newly created agile metric is relevant, and there is high conformity of measured values with the referenced metric. As a benefit, in comparison with the Boehm-Turner model, the newly created agile metric has a much greater granularity, and the measurement method is more accurate.

6 | Conclusions

Organizational patterns are the key to a stepwise adoption of agile and lean approaches and to a piecemeal growth of agile and lean organization of work. They can be applied to correct particular problems within an organization or to build a new organization.

Although there has been much research done on the agile methodologies, and lean software development, and in particular about patterns and pattern languages, little work has been done in finding how organizational patterns can enhance agile and lean software development. Using organizational patterns efficiently can help with identifying the best possible solution for tempering down problems of the ever-changing dynamic market conditions and enhancing the software development process. In this thesis, we presented several approaches and novel methods, proposed various changes in order to fulfill the thesis statement as presented in Section 1.1.

The main contributions of this thesis are as follows:

Animating organizational patterns (Chapter 3). This approach brings possibility to experience different situations on own skin without being afraid of making mistakes. By supporting curiosity, participants may discover uncharted combinations of applied patterns and observe how they work in tandem. As a result of this, it contributes to learning and mastering their usage, which promotes better decision-making.

Modeling variants of organizational patterns (Chapter 4). Patterns may evolve over time, and new variations can emerge based on previous experiences or from the requirement to adopt a new pattern. Each organization gradually got to the point that variations of existing patterns are needed. However, variations can be perceived in pattern languages too. To support variations, we created a tool that can be later used as a community portal for participants dealing with patterns and pattern language.

Measuring and improving agility with organizational patterns (Chapter 5). For the purpose of differentiating the scope of prepared changes, there is usually a need to be able to measure them to distinguish consequences. Measuring

agility is a complex matter due to its abstract complexity, which does not have physical relation. We solved this problem by relying on the organizational patterns, which are used as fundamental detection indicators for searching for the presence of their (agile) elements.

References

- [1] C. Alexander, *The Timeless Way of Building*. Oxford University Press, 1979.
- [2] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.
- [3] R. T. Bakie, "A brief history of video games," in *Introduction to Game Development*, S. Rabin, Ed., Second, Cengage Learning, 2009.
- [4] M. E. Conway, "How do committees invent," *Datamation*, vol. 14, no. 4, pp. 28–31, 1968, Available at <http://www.melconway.com/research/committees.html>.
- [5] J. O. Coplien and N. B. Harrison, "*Organizational Patterns of Agile Software Development*". Prentice-Hall, 2004.
- [6] J. Coplien and G. Bjørnvig, *Lean Architecture: for Agile Software Development*. Wiley, 2011, ISBN: 9780470970133. [Online]. Available: <https://books.google.sk/books?id=lpvY36MPMUwC>.
- [7] A. a. Deshpande and S. H. Huang, "Simulation games in engineering education: A state-of-the-art review," *Computer Applications in Engineering Education*, vol. 19, no. 3, pp. 399–410, Sep. 2011.
- [8] S. Deterding, D. Dixon, R. Khaled, L. Nacke, and S. Deterding, "From game design elements to gamefulness: Defining gamification," *15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pp. 9–15, 2011.
- [9] A Gamlin, P. Breedon, and B. Medjdoub, "Immersive virtual reality deployment in a lean manufacturing environment," *Interactive Technologies and Games*, pp. 51–58, 2014.
- [10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [11] J. P. Gee, "What video games have to teach us about learning and literacy," *Computers in Entertainment*, vol. 1, no. 1, p. 20, Oct. 2003.

-
- [12] C. Hartog, "Scenario design for serious gaming – guiding principles for the design of scenarios and curricula in military job oriented training," Master's thesis, Delft University of Technology, 2009.
- [13] P. Honišek and V. Vranić, "Mining drama patterns in dramatic situations," in *Proceedings of 25th Conference on Pattern Languages of Programs Online, PLoP 2020*, ACM, 2020.
- [14] H. McManus and E. Rebutisch, "Experiences in simulation-based education in engineering processes," *2008 38th Annual Frontiers in Education Conference, S1C-21-S1C-26*, Oct. 2008.
- [15] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM*, vol. 48, no. 2, pp. 72–78, 2005.
- [16] A. G. Ramos, M. P. Lopes, and P. S. Avila, "Development of a platform for lean manufacturing simulation games," *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 8, no. 4, pp. 184–190, Nov. 2013.
- [17] P. P. Repository. (1995). Fedwiki c2, [Online]. Available: <http://wiki.c2.com>.
- [18] —, (1995). Pattern forms, [Online]. Available: <http://wiki.c2.com/?PatternForms>.
- [19] B. Scharlau, "Games for teaching software development," *Proceedings of 18th ACM conference on Innovation and technology in computer science education*, 2013.
- [20] Scrum Pattern Community, *Published patterns*, Available at <https://sites.google.com/a/scrumlop.org/published-patterns/home>, 2015.
- [21] I. d. Silva, A. R. Xambre, and R. B. Lopes, "A simulation game framework for teaching lean production," *International Journal of Industrial Engineering and Management*, vol. 4, no. 2, pp. 81–86, 2013.
- [22] J. Sutherland, J. O. Coplien, and C. Ramos, *A Scrum Book: The Spirit of the Game*, ser. Pragmatic Programmers. Pragmatic Bookshelf, 2019, ISBN: 9781680506716. [Online]. Available: <https://books.google.sk/books?id=NKtYwwEACAAJ>.
- [23] J. Sutherland, J. O. Coplien, J. Østergaard, G. Bjørnvig, and D. Friis, *Scrum as organizational patterns*, Available at <https://sites.google.com/a/scrumorgpatterns.com/www/>, 2011.
- [24] V. Vaishnavi, B. Kuechler, S. Petter, and G. D. Leoz, *Design Science Research in Information Systems*, <http://www.desrist.org/design-research-in-information-systems/>, year = 2019,
- [25] L. R. Vijayarathy and D. Turk, "Agile software development: A survey of early adopters," *Journal of Information Technology Management*, no. 2, pp. 1–8, 2008.

-
- [26] A. Vranić, V. Vranić, and B. Vranić, “Drama patterns: Seeing the patterns from within: Focus group report,” in *Proceedings of the 24th European Conference on Pattern Languages of Programs*, ser. EuroPLOp ’19, Irsee, Germany: Association for Computing Machinery, 2019, ISBN: 9781450362061. doi: 10.1145/3361149.3361196. [Online]. Available: <https://doi.org/10.1145/3361149.3361196>.
- [27] V. Vranić and A. Vranić, “Drama patterns: Extracting and reusing the essence of drama,” in *Proceedings of the 24th European Conference on Pattern Languages of Programs*, ser. EuroPLOp ’19, Irsee, Germany: Association for Computing Machinery, 2019, ISBN: 9781450362061. doi: 10.1145/3361149.3361153. [Online]. Available: <https://doi.org/10.1145/3361149.3361153>.
- [28] V. Vranić, “Promoting natural human attitude towards work: Scrum,” in *Proceedings of Konferencija Mreža 2013 – Internet u edukacionom i poslovnom okruženju (Conference Mreža 2013 -- Internet in Educational and Business Environment)*, Valjevo, Serbia, 2013, pp. 8–12.
- [29] M. Wieck, “Patterns: Lust for glory,” in *Proceedings of 14th Annual Conference of the National Advisory Committee on Computing Qualifications, NACCQ 2001*, 2001, pp. 145–151.
- [30] U. Zdun, “Systematic Pattern Selection Using Pattern Language Grammars and Design Space Analysis,” *Software: Practice and Experience*, vol. 37, no. 9, pp. 983–1016, 2007.
- [31] L. Zhou, Y. Xie, N. Wild, and C. Hunt, “Learning and practising supply chain management strategies from a business simulation game: A comprehensive supply chain simulation,” *Simulation Conference*, 2008.

A | Publications

Zoznam publikačnej činnosti

Autor: Frťala, Tomáš

ADM Vedecké práce v zahraničných časopisoch registrovaných v databázach Web of Science alebo SCOPUS

ADM01 VRANIĆ, Valentino - PORUBĀN, Jaroslav - BYSTRICKÝ, Michal - FRĚALA, Tomáš - POLÁŠEK, Ivan - NOSĀL, Milan - LANG, Ján. Challenges in preserving intent comprehensibility in software. In *Acta Polytechnica Hungarica*. Vol. 12, no. 7 (2015), s. 57-75. ISSN 1785-8860 (2015: 0.544 - IF, Q3 - JCR Best Q, 0.379 - SJR, Q1 - SJR Best Q). V databáze: SCOPUS: 2-s2.0-84948740449 ; WOS: 000366994500004.

AFC Publikované príspevky na zahraničných vedeckých konferenciách

AFC01 FRĚALA, Tomáš - VRANIĆ, Valentino. Animating organizational patterns. In *Proceedings of eighth international workshop on cooperative and human aspects of software engineering CHASE 2015, May 18, Florence, Italy*. 1. vyd. Los Alamitos : IEEE, 2015, S. 8-14. ISBN 978-1-4799-1934-5. V databáze: SCOPUS: 2-s2.0-84946074616 ; WOS: 000380425100002.

AFC02 FRĚALA, Tomáš - VRANIĆ, Valentino. Improving Organizational Pattern Comprehensibility by Varying Their Form and Relationships. In *2022 Zooming Innovation in Consumer Technologies Conference (ZINC), 2022*

BEF Odborné práce v domácich zborníkoch (konferenčných aj nekonferenčných)

BEF01 FRĚALA, Tomáš. Odporúčacie systémy orientované na znovupoužitie kódu. In *Odporúčanie pre softvérových inžinierov*. 1. vyd. Bratislava : Nakladateľstvo STU, 2015, S. 93-101. ISBN 978-80-227-4494-2.

BEF02 FRĚALA, Tomáš. Odporúčacie systémy pre manažment problémov. In *Odporúčanie pre softvérových inžinierov*. 1. vyd. Bratislava : Nakladateľstvo STU, 2015, S. 127-133. ISBN 978-80-227-4494-2.

Štatistika: kategória publikačnej činnosti

ADM	Vedecké práce v zahraničných časopisoch registrovaných v databázach Web of Science alebo SCOPUS	1
AFC	Publikované príspevky na zahraničných vedeckých konferenciách	2
BEF	Odborné práce v domácich zborníkoch (konferenčných aj nekonferenčných)	2
Súčet		5

Štatistika: kategória ohlasov

1	Citácie v zahraničných publikáciách, registrované v citačných indexoch Web of Science a databáze SCOPUS	10
2	Citácie v domácich publikáciách, registrované v citačných indexoch Web of Science a databáze SCOPUS	1
3	Citácie v zahraničných publikáciách neregistrované v citačných indexoch	1
Súčet		12

Štatistika: skupiny kategórií publikačnej činnosti

Skupina B	Publikácie v karentovaných vedeckých časopisoch (ADC, ADD, BDC, BDD, CDC, CDD, AGJ)	1
Skupina C	Publikácie, ktoré nie sú karentované, ale sú registrované v databázach WoS alebo Scopus (ADM, ADN, BDM, BDN)	4
Skupina X	Nezaradené	2
Súčet		7