The Importance of Program Design Patterns Training

Viggo Holmstedt
University College of South-East Norway
School of Business, Department of Business and IT
Horten, Norway
vh@usn.no

Shegaw A. Mengiste
University College of South-East Norway
School of Business, Department of Business and IT
Horten, Norway
sme@usn.no

***Abstract*—Design Patterns for Object Oriented Systems constitute an important tool for improving software quality by providing reusable design. Many academic institutions believe in their relevance, and do courses accordingly. This paper explores practitioners' perception of the relevance their patterns knowledge has for their work. The paper also assesses how managers' perception of pattern knowledge conforms with practitioners' perceptions. We found convincing evidence for practitioners' confidence in pattern knowledge and its positive influence on their coding abilities. Based on our findings we claim that training of design patterns is important for practitioners.**

***Index Terms*—Design Patterns, training, empirical assessment, practitioners, managers.**

I. Introduction

The success rate of systems development was 30% in the year 2015. The Standish Group Chaos Reports told us this. Such low rate of success indicates that system development is a complex process and need to be addressed with proper planning and guiding. In systems development, earlier design decisions can have a significant impact on software quality; they can also be the most costly to revoke [1]. Design Patterns (hereafter DP, used in plural form) constitute an important tool for improving software quality by providing reusable solutions for recurring design problems.

Design patterns are best practices of specifying and allocating responsibilities to program elements, like classes, packages and components. DP also support the construction of mechanisms based on patterns of class cooperation. Industrial usage and success over a long time typically establishes and confirms a specific design pattern, accepted as a guide to construct mechanisms in complicated object oriented systems.

Due to the growing importance and value of design patterns in software development, many academic institutions run courses on Design Patterns for Object Oriented systems.

Much interesting research explore DP quality attributes that can influence software quality. Those studies often ignore the human courage and coder enthusiasm as important attributes in the development environment. This paper explores practitioners' perception of the relevance of their design patterns knowledge and its impact in their professional practices. The paper also assesses how managers' perception of design patterns knowledge conforms with practitioners' perceptions.

We acknowledge that many other researchers have investigated the power of DP to improve the software produced under pattern rules. Among many others we appreciate the works of Khomh et al [2] and Wydaeghe [3] who study and evaluate DP quality attributes. Productivity by several types of reuse is treated by Ampatzoglou et al [6]. They depict DP as a starting point for whitebox reuse, and allocate adhoc and opportunistic reuse to smaller companies in great quanta. We have investigated smaller companies only in our research.

Our research is different. We want to fill a gap in DP research by focusing the practitioners' perceptions, self-esteem and social aspects of DP knowledge and usage. Moreover, we trace knowledge of DP to bear confidence and enthusiasm to the design and coding activities, boosting the human energy needed to perform great coding.

The bottom-line for our investigation is to assess the value or relevance of DP to help software developers to produce better software by guiding them in code production. More specifically, we would like to assess the different perspectives on the relevance of running courses in DP, particularly in terms of the experience and minds of the social world of practitioners and their employers. To address this research problem, we formulated the following research questions:

Q1: How, when and why do DP trained practitioners perceive relevance of DP knowledge?

Q2: How mutual is practitioners and managers' understanding of the relevance of DP?

By exploring these research questions, we can contribute to the ongoing research debate between research of DP as a tool to improve software versus DP as a tool to improve thinking, courage and self-efficacy. We believe the latter factors contribute to the improved quality of the practitioner.

II. Conceptual Framework

In the context of the study, we adopted the social worlds perspective as our theoretical framework [4]. We identified two important social worlds: the social world of software developers (practitioners), and the social world of managers (practitioners' superiors). The agencies of both worlds are the production of software, including the learning of best practices to enhance the return on investments.

III. Research Approach and Methods

This study is an empirical research in the context of *engaged scholarship* [5]. The data collection was prepared by the identifying and surveying around 170 potential respondents in Norway, of which a total of 28 fully completed the survey. Among the 28 respondents, there were 20 practitioners and 8 managers. The two groups responded to a prepared set of interview keywords customized for each group.

Each of the 170 candidates in the list had one or more states, like "is contacted", "promised to answer", "is not relevant", "has contributed", "pushed once" and several more. To keep track of the status of each candidate respondent, we developed a CMS tool. We could then practice precise candidate pushing and facilitate respondent accounting. The NVivo tool, a software package for qualitative data analysis, facilitated the analysis.

IV. Findings and Conclusion

Our focus was to know the perspectives and views of the two social worlds (practitioners and managers) on the relevance and value of design patters in work settings. We separate our findings accordingly for the two social worlds, before we make a comparison of our findings in the two stakeholder groups.

One of our main findings revealed that DP is depicted as a knowledge framework that lets participants discuss and elaborate solutions across social worlds. The box is summing up some results for Q1:

> DP has a big role in practitioner's evolution, studying DP has been great, makes much better coders, allows for architectural perspective, keeps code maintainable, is timesaver

From the research Q2 perspective, we include and compare the management perceptions. We can list the response analysis with the following few main points:

- The communication between management and practitioner profits on a mutual understanding of tools and methodologies.
- There is support and evidence for positive management perceptions.
- There is evidence for a mutual understanding between practitioners and managers with regard to the relevance of DP.
- There is also a concern about the combination of quality of code and economy
- Managers believe in the positive influence of DP on code improvement
- Managers do have concern for DP knowledge when new hires are considered
- DP infer better coding, keep the code maintainable and even give coders a view into architectural considerations

Overall, our findings revealed that there are differences between IT managers and practitioners in how they perceive DP as a vehicle to enhance performance of development team. Practitioners expressed high level of relevance for the knowledge of DP, while IT managers put a lower level of relevance to DP knowledge. However, both groups believed in DP's ability to act as a communication tool.

References

[1] Folmer, E. and J. Bosch, A pattern framework for software quality assessment and tradeoff analysis. International Journal of Software Engineering and Knowledge Engineering, 2007. 17(04): p. 515-538.

[2] Khomh, F. and Y.-G. Guéhéneuc. Do design patterns impact software quality positively? in Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on. 2008. IEEE.

[3] Wydaeghe, B., et al. Building an OMT-editor using design patterns: An experience report. in Technology of Object-Oriented Languages, 1998. TOOLS 26. Proceedings. 1998. IEEE.

[4] Clarke, A.E. and S.L. Star, *The social worlds framework: A theory/methods package.* The Handbook of Science & Technology Studies, 2008. **3**: p. 113-137.

[5] Van de Ven, A.H., *Engaged scholarship : a guide for organizational and social research*. 2007, Oxford ; New York: Oxford University Press. xii, 330 p.

[1] Ampatzoglou, A., et al., *An empirical investigation on the impact of design pattern application on computer game defects*, in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. 2011, ACM: Tampere, Finland. p. 214-221.