The Impact of Design Patterns On Software Quality and Maintainability

Sief Addeen Bashar Ali Alhunaiti[1] and Mohamed Samiulla Khan[1#]

[1]Middle East College, Muscat, Oman
[#]Advisor

ABSTRACT

Designing Software is a challenging and complex task, maintaining quality and high standard in this process is a bigger challenge. Design Patterns commonly known as reusable solutions to some standard problems have been adopted in many software development projects to solve challenging tasks of software design and development . The following research paper examines and studies the impact of design patterns on software quality and maintainability. The study explores whether design patterns are the best approach to solving common software design problems, or if there are better alternatives. The research explores existing literature on Design Patterns and their impact on software development process. (Alexander, 1977)

Introduction

The concept of design patterns in software development process was inspired by the book “A Pattern Language” written by Christopher Alexander in 1977 . In 1994 "Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, published the book "Design Patterns: Elements of Reusable Object-Oriented Software". This book is considered the most influential and fundamental in the field of software development. This book for the first time identified 23 different design patterns for commonly occurring problems in software component design and implementation. After this many books ,research papers are written on this topic and it is continuously evolving. Design patterns are said to have several advantages, including support for better design decisions, improved software maintainability, reusability, software developer communication. It also aids in saving costs and in assuring non-functional requirements of the system. It can be characterized as a description or model for a problem-solving approach that can be applied in a variety of contexts. Design patterns have become essential for software developers, architects, and project managers in making informed decisions during the software development process to ensure software quality and maintainability. (Gamma,1994)

Literature Review

The studies that was read and reviewed discusses the talks about how design patterns are not completely evaluated in software engineering they're only partially evaluated and that's because the research on the effect of design patterns has only been contentious it is almost impossible to draw firm conclusions on that software the study refers to a study by Prechelt which is called PatMain the study assisted the main disability of applications and taking advantage of design patterns in comparison to more straightforward options.

There is search examined the accuracy and the time of maintenance for expert users by using four different systems the examiners found out and came to the conclusion that applying a design pattern was way more beneficial, but it is best to use the engineer's common sense one thing that this season whether or not because sometimes a straightforward solution is more optimal. When the engine is not sure about it is better to use design patterns that would be the better approach. For the maintaining of programs deep knowledge and high rates of understanding of design patterns is more useful.

The study stresses the value of more empirical studies on how design patterns change and affect software quality where it discusses the PatMain experiment, the experiment aimed at finding and determining the effect of patterns on software maintainability the experiments happened in two investigations.

Prechelt and Liebenberg's other study had 13 students and two of the four systems from the original experiments the final result supported the original study conclusions but only one significant discovery was found which is that one of the systems was more maintainable and extendable before using the pattern format it was better in the non-pattern form the second study which was done by Juristo and Vegas aimed at three patterns first composite second decorator and last abstract factory they studied 2 software systems and two different languages they used eight participants for the text doesn't tell what was found on that study.

One study used the JHotDraw software and found a huge correlation between frequency and the maintainability of design patterns in the source code.

Another different study used systems to review literature in order to find evidence that discusses the benefits and the usefulness of patterns and helping and providing A framework for maintenance, but the result was unable to come up with concrete solid recommendations for the best most efficient application for specific certain patterns. (Alghamdi, F. M., & Qureshi, M. R. J 2014).

One study inspected verified data that were already available on how patterns affect the quality of software like maintainability, performance, and the likeability to fail, and according to that study, design patterns had a dangerous damaging, and harmful effect on maintainability and evolution however based on the infestation the fact on the performance on the likeability to fail was non-identical. Several and many other studies suggested ways to help designers with the selection of the most optimal design pattern unattractive one approach used a tool that led designers and helped them through a set of comparison criteria and suggestions therefore study did a comparison between three different design patterns with two different alternate solutions.

The study concluded that while patterns can be useful in specific situations it is still very unknown how much they will affect the level of integrity of our software in order to get more reliable evidence on the effect of how much design patterns have on software and that's maintenance more studies and more research is required. (Wedyan, F, & Abufakher, S. 2020).

The relation between the application of his patterns and the maintainability of software is explored in this study whenever a design pattern was in over 300 times of JHotDraw a Java framework that uses GUI and relies on them found that quality attributes have gone better and improved. They found a very strong relationship between the density of design patterns and estimated rentability ratings the study discusses and makes the core of the research the design patterns increase the level of code maintainability but more balance studies and research is needed. (Hegedűs, P,et al 2012).

The influence of the GoF on software is viewed using the valid empirical data that are currently available the study had two goals the first goal was to evaluate and check the empirical evidence's coverage of quality and design patterns and to give an outline and a run down on how does design patterns affect these aspects the findings showed that there is no connection and no agreement about the effect of design patterns which was only studied in relation of four quality points did not cover all of the GoF design patterns either into consideration the cofounding factors measurements implementations and practices the study intended to explain the conflicting and hard and very difficult to compare results of studies on the influence of design patterns on the quality of software the researchers also wanted to offer a plan in order to enhance the the implementation of patterns and organization to help further advance the quality of software they did a thorough assessment of 50 top primary research that was between 2000 and 2018 and the result was that they discovered but the quality is obviously clearly had an impact and influence binary recording of patterns the number of pattern classes and the degree of patterns the researchers recommend that further studies should focus on enhancing pattern modularity and put unified measures in order to come to an agreement regarding the effect of patterns on systems and their quality. (Ali, M., & Elish, M. O. 2013, June).

This study was made to figure out whether patterns increase software quality and understandability to researches where done the first experiment was the Redo of a controlled experiment comparing the main stability of a pattern-implemented program to a less sophisticated complex one the second study looked up if using pattern designs made the software easier to understand both experiments and studies results showed that pattern designs did not enhance maintainability or understandability albeit the participants had the system designs prior to performing maintenance tasks there was no connection or correlation and between understandability and the ability to maintain it the participants were graduate-level participants and they were the ones that were in this test. (Nanthaamornphong, A., & Carver, J. C. 2011, September)

In this research a theoretical framework that is analytical is proposed for concentrating design patterns with multiple different designs in terms of the number of quality aspects of the theatrical research because I want to go not related to actual systems but the results were based on existing actual issues that's research examined three different patterns with very different approaches the approach was based on the number of classes they may have antiquities that represented the values of key structural qualities the results and the findings suggest that having that choice to use design pattern is a most likely trade-off since design patterns do enhance certain elements of quality and software while reducing the quality of from others the ability of decision making during system designs can get a lot of help from the suggested technique which can be used to put a comparison between alternative designs and patterns . (Ampatzoglou, A, 2012).

To reach strong experimental realism the study provides our application of experiments that have already been done on design patterns and maintenance the capability of applying pattern design to software maintenance is studied and investigated as well as a pause on whether different types of patterns are more challenging to apply than other types the results find and suggest that specific patterns are way simpler to comprehend and use than others and the design patterns have to be applied in a way that is relevant and suitable to the situation and the user while the Observer and Decorator patterns were understood intuitively and used even by users with less pattern knowledge the Visitor pattern was much harder to understand and it was confusing the final result conclude that when added to our program and applied with a designated plan even a user with little to no training may and could increase the speed and the level of quality of maintenance tasks. (Vokáč, M., et al. 2004).

Methodology

To find out how design patterns affect software maintainability, the researcher reviewed relevant literature. The researcher looked through studies, journals, and Google Scholar.

Empirical valid research examining the impact of design patterns on software maintainability met the inclusion criteria for the study.

Results & Analysis

Many studies discovered and showed that the engineering community is divided on the effect and influence and affect of design patterns on software quality & maintainability. The review showed that design patterns may affect software maintainability in good and bad ways. many studies that stood behind the usage of design patterns in software maintenance were found by the researchers. Design patterns, according to specific studies, may make situations worse and code maintenance difficult and harder to do.

A further finding of the research was that some patterns were more in favor than others. In one study the system JHotDraw, discovered that there is a very strong relation between maintainability and the prevalence of design patterns in the code.

Conclusion & Recommendations for Future Research

In conclusion after reviewing many types of research on the influence of design patterns on software quality has been disputed and ambiguous, to put it easily. While few studies said that design patterns can raise the caliber of software, many say that they are harmful. The research findings and results differ based on the quality component being looked at, such as the level of failure to occur, assessment, performance, and maintainability. An abundance of studies show an outline for designers to help them choose the appropriate design patterns for each specific situation, but it is still unclear and unknown how each of these patterns affects and influences software maintenance. In order to give more trustworthy valid evidence of the influence of design patterns on software quality measures, much research is required and needed. Future research should concentrate and focus on improving pattern modularity, using unifying metrics, and delineating how design patterns affect various elements of software quality. It is recommended that software engineers exercise common sense when deciding whether to use a design pattern or not and consider it as a trade-off between enhancing certain aspects of software quality while potentially detracting from others.

References

Alexander, C. (1977). *A pattern language: towns, buildings, construction.* Oxford university press.

Gamma, E. (1994). Design Patterns-Elements of Reusable. *Object-Oriented Software.*

Alghamdi, F. M., & Qureshi, M. R. J. (2014). Impact of design patterns on software maintainability. *International Journal of Intelligent Systems and Applications*, *6*(10), 41. https://www.researchgate.net/profile/M-Rizwan-Qureshi/publication/272853405_Impact_of_Design_Patterns_on_Software_Maintainability/links/57c2803e08aeda1ec38cf95c/Impact-of-Design-Patterns-on-Software-Maintainability.pdf

Hegedűs, P., Bán, D., Ferenc, R., & Gyimóthy, T. (2012). Myth or reality? analyzing the effect of design patterns on software maintainability. In *Computer Applications for Software Engineering, Disaster Recovery, and Business Continuity: International Conferences, ASEA and DRBC 2012, Held in Conjunction with GST 2012, Jeju Island, Korea, November 28-December 2, 2012. Proceedings* (pp. 138-145). Springer Berlin Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-35267-6_18

Ali, M., & Elish, M. O. (2013, June). A comparative literature survey of design patterns impact on software quality. In *2013 international conference on information science and applications (ICISA)* (pp. 1-7). IEEE. https://ieeexplore.ieee.org/abstract/document/6579460
Wedyan, F., & Abufakher, S. (2020). Impact of design patterns on software quality: a systematic literature review. *IET Software*, *14*(1), 1-17. https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-sen.2018.5446

Nanthaamornphong, A., & Carver, J. C. (2011, September). Design patterns in software maintenance: An experiment replication at University of Alabama. In *2011 Second International Workshop on Replication in Empirical Software Engineering Research* (pp. 15-24). IEEE. https://ieeexplore.ieee.org/abstract/document/6148333

Ampatzoglou, A., Frantzeskou, G., & Stamelos, I. (2012). A methodology to assess the impact of design patterns on software quality. *Information and Software Technology*, *54*(4), 331-346. https://www.sciencedirect.com/science/article/abs/pii/S0950584911002151

Vokáč, M., Tichy, W., Sjøberg, D. I., Arisholm, E., & Aldrin, M. (2004). A controlled experiment comparing the maintainability of programs designed with and without design patterns—a replication in a real programming

environment. *Empirical Software Engineering*, *9*, 149-195.
https://link.springer.com/article/10.1023/B:EMSE.0000027778.69251.1f