Design Pattern Impact on Reliability and Usability an SMS

Brandon Uehlein and Isaac Griffith
Informatics and Computer Science
Idaho State University
Pocatello, Idaho, 83208
Email: {uehlbran@isu.edu, grifisaa@isu.edu}

***Abstract*—Blah:**

***Index Terms*—**

1. Introduction

In software engineering, quality attributes such as usability and reliability are used as measures of software quality. In total, there are six main software quality attributes listed in the software quality model written by the International Organization for Standardization located in ISO 9126-1 [citation]. Usability deals with the level of difficulty users experience when using a product. Low usability is a good indicator that a product is difficult to use and is likely to be rejected by users. ISO 9126-1 defines usability as "A set of attributes that bear on the effort needed for use, and on the individual assesment of such use, by a stated or implied set of users." [citation]. ISO 9241-11 and ISO/TR 16982:2202 define usability as "" [citation]. ISO 25010:2011 defines uability as "" [citation]. This tells us that usability has it's own set of attributes, subcharacteristics, that determine the usability of a product. ISO 9126-1 defines these subcharacteristics as: understandability, learnability, operability, attractiveness, and usability complicance. However, throughout this paper we will be using Jakob Nielsens definition of usability which is: learnability, efficiency, memorability, erros, and satisfaction. More information on usability can be found in section 2.3.

Reliability as defined by ISO 9126-1 is "A set of attributes that beat on the capability of software to maintain its level of performance under stated conditions for a stated period of time." [citation]. Like usability, reliability can also be broken down into subcharactertistics: maturity, fault tolerance, recoverability, and reliability compliance.

2. Background and Related Work

2.1. Design Patterns

In software engineering we as developers tend to encounter a lot of the same problems regardless of the actual work we do. In fact, most of the problems we as developers solve are not brand new. These common, reocurring problems are the reason that design patterns exist. Design patterns give us a repeatable solution that we can use to solve these problems. However, design patterns are not actual, concrete solutions to these problems, but rather they give us a template that we can work from in order to solve the problem. These templates allow us to incorporate best practices into our code while allowing us to focus on what really matters.

Design patterns first appeared around 1977 as an architectural concept discovered by Christopher Alexander [citation]. Close to a decade later, Beck et al began experimenting with the use of design patterns in programming. Not to long after the work done by Alexander and Beck et al. a book on design patterns was released. The book, "Design Patterns: Elements of Reusable Object-Oriented Software" was published in 1994 by a group of software engineers known as the "Gang of Four" [citation - gamma, erich et al].

2.2. Reliability

2.3. Usability

Usability is a quality attribute that is used to assess the level of difficulty associated with using a system [1]. Usability has been defined in multiple standards including: ISO 9126, ISO 9241-11, ISO/TR 16982:2202, and ISO 25010:2011 as well as multiple papers, articles, and books written by other usability experts. The definition were going to use in this paper is from ISO 25010:2011 which is an updated version of ISO 9126. ISO 25010:2011 defined usabilty as "". Usability can be further subdivided into subcharacteristics known as usability quality components. There are five of these subcharacteristics in total: learnability, usability, memorability, errors, and satisfaction. These will be discussed in more detail in section 2.3.1.

When most people think about usability the first thing that usually comes to mind is the user interface (UI). However, usability isn't limited to just the UI of a system. In fact, usability deals with the entire interaction betweem users and a system. A system with low usability is going to be difficult to use, hard to understand,

ISO 9126 defines usability as "A set of attributes that bear on the effort needed for use, and on the individual

assesment of such use, by a stated or implied set of users." [ISO 9126]. ISO 9241-11 and ISO/TR 16982:2002 define usability as the "extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfation in a specified context of use." [ISO 9241-11; ISO/TR 16982:2002]. ISO 25010:2011 defines usability as "" [citation].

2.3.1. Usability Quality Components. Nielsen defined five quality components that make up usability. According to Nielsen, usability is a quality attribute that assesses how difficult a user interface (UI) is to use.

- **Learnability -** Deals with the level of difficulty a user will experience when first interacting with a system to accomplish basic tasks.
- **Efficiency -** Looks at how long it takes a user to complete a certain task once they've gained familiarity with the system.
- **Memorability -** Focuses on a users ability to return to a system after a period of time and regain efficiency using it.
- **Errors -** Looks at the frequency, severity, and recoverability of errors users make while interacting with the system.
- **Satisfaction -** Deals with the amount of fulfillment a user gets from using the system.

[2] Deals with this topic

2.3.2. Usability Inspection. According to Nielsen et al, usablity inspection is a set of methods where evaluators inspect a user interface (UI) to identify usability issues, severeity of issues, and the overall usability of a design. There are four basic types of usability inspection methods: automatic, empiracally, formal, and informal. Automatic usability inspection is accomplished by running a UIs specifications through a computer program. Empiracal usability inspection invovles using actual users to test the UI. Formal usability inspection is done using models and formulas to calculate a usability score. Finally, informal usability inspection is based on general guidelines and principals as well as the skill and experience of the evaluators testing the UI [3]. Next, will take a look at the actual methods involved in usability inspection.

- **Heuristic evaluation -** is an informal usability inspection method where evaluators, usability experts, independently judge a UI based on its compliance with a set of usability principles known as heuristics. This method relies solely on the skills and experience of the evaluators to find potential issues in the design.
- **Cognitive walkthroughs -** is an formal usability inspection method where evaluators, usually the developers and designers of the software, take turns role playing as users of the software to perform specific tasks while the other members evaluate and document the process for later refinement.
- **Formal usability inspections -** is a formal usability inspection method and is a combination of heuristic evaluation and the cognitive walkthough methods. It is a six step process that involves: assembling a team, assigning roles to each member, distributing appropriate documentation (design, technical, etc), giving out instructions on what is to be done, independent evaluation of the software, and finally, a discussion of issues and concerns that were identified during the process.
- **Pluralistic walkthroughs -** is an informal usability evaluation where a group of individuals, users, developers and designers, and usability experts walk through a scenario together and discuss any issues or concerns they might have.
- **Feature inspections -** is an formal usability inspection method where a set of features used to accomplish a specific task are evaluated based on the given user story.
- **Consistency inspections -** is an informal usability inpsection method where designers from multiple projects inspect an interface based on how similiar it is to their own projects.
- **Standards inspections -** is an informal usability inspection method involving usability experts with knowledge of interface standards use their skills and experience to evaluate a UI based on its compliance with said standards.

[3], [4].

2.3.3. Usability Testing Methods.

- **Hallway testing -**
- **Remote usability testing -**
- **Expert review -**
- **Automated expert review -**
- **A/B Testing -**

3. Systematic Mapping Approach

A systematic mapping study (SMS) is used to provide a broad overview of a particular research area by identifying previous primary studies whose results are then categorized to create a visual summary, the map. In this paper, the systematic approach used is defined by Kitchenham et al. and later refined by Peterson et al. There are five steps to this process: definition of research questions, searching for relevant papers, screening of papers, and keywording of abstracts. We chose to use this approach because it is well-defined, allows for more general conclusions, and in doing so helps to reduce bias. We decided to go with an SMS instead of a systematic literature review (SLR) for two reasons. The first reason is because we wanted a broader overview of the field and secondly, we were more interested in general trends rather than the in-depth analysis that an SLR provides. Our goal with this paper is to identify what research has already been done with regards to design patterns and their affect on usability and reliability

to determine where new or better research can take place **???**, [5].

3.1. Research Questions

- **RQ1:** What design pattern types do research study when considering usability and reliabiity?
- **RQ2:** What recomendations have been made regarding application of design patterns and in the context of usability and reliability?
- **RQ3:** How is the impact on usability and reliability from design patterns currently evaluated?
- **RQ4:** What type of projects or domains were studied?
- **RQ5:** Where are papers concerning design patterns and reliability and usability published?
- **RQ6:** What types of studies are conducted regarding design pattern impact on usability and reliability?
- **RQ7:** In what phase of development do proposed results apply?
- **RQ8:** What tools are utilized for the research and to which languages do they apply?

3.2. Data Sources

The data sources for this paper included IEEEXplore, ACM Digital Library, SpringerLink, Web of Science, and the Science Direct database. In order to provide a broad overview of the field and ensure relevant papers were selected we limited our search to the years between 2009-2019.

3.3. Search Queries

The search query used in this paper was created from keywords based on our research questions abd included: usability, reliability, design patterns, and pattern.

The specific search query used in this paper was: (usability OR useable OR learnability OR learnable OR efficiency OR satisfaction OR memorability OR memorization) AND (reliability OR reliable OR integrity OR maturity OR compliance OR "fault tolerance") AND ("design pattern" OR pattern)

3.4. Inclusion Criteria

- Only papers whose main focus is on design patterns and their effect on usability or reliability were available for inclusion.
- For papers where multiple iterations of the same paper exist only the latest version was eligible for inclusion.
- Papers that included either reliability or usability and design pattern or pattern as keywords to identify the paper topics.

3.5. Exclusion Criteria

- Papers not written in the English language.
- Papers whose title or abstracts did not contain the selected keyword phrases.
- Papers which discussed the impacts of design patterns on quality attributes exclusing reliability or usability or their sub-characteristics.

3.6. Quality Criteria

3.7. Snowballing Approach

We utilized the snoballing approach defined by wohlin et al. [X] to dinf further sources overlooked by the initial search. Specifically, we utilized the following approach:

4. Threats to Validity

This is a blah

4.1. Conclusion Validity

4.2. Construct Validity

4.3. Internal Validity

4.4. Reliability

5. Results

6. Conclusion

7. Recommendations

References

[1] N. Juristo, A. M. Moreno, and M.-I. Sanchez-Segura, "Analysing the impact of usability on software design," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1506–1516, Sep. 2007.

[2] X. Ferre, N. Juristo, H. Windl, and L. Constantine, "Usability basics for software developers," *IEEE Software*, vol. 18, no. 1, pp. 22–29, Jan. 2001.

[3] R. Mack and F. Montaniz, "Usability Inspection Methods," J. Nielsen and R. L. Mack, Eds. New York, NY, USA: John Wiley & Sons, Inc., 1994, pp. 295–339.

[4] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, 1990, pp. 249–256.

[5] B. A. Kitchenham, D. Budgen, and O. Pearl Brereton, "Using mapping studies as the basis for further research – A participant-observer case study," *Information and Software Technology*, vol. 53, no. 6, pp. 638–651, Jun. 2011.