Factors Influencing Design Quality and Assurance in Software Development: An Empirical Study

Cristina Valdaliso
Rutgers/NJIT University, Newark, NJ
Valdaliso@earthlink.net

Osama Eljabiri
New Jersey Institute of Technology, Newark, NJ
Omae@home.net

Fadi P. Deek
New Jersey Institute of Technology, Newark, NJ
Deek@njit.edu

Abstract

The purpose of this research was to investigate the factors influencing the design quality and the quality assurance in the design phase of the software process. The dependent variable in this study was design quality and the independent variables were availability of financial resources, architectural style implemented, methodology adopted, employees' training, and employees' involvement. The survey collection method comprised a 30-item questionnaire. This comprehensive study included all 55 programmers from two different companies in the population framework. Respondents were 54.5% of the population size. The data analysis to test the study hypotheses included mainly the use of K-S test, F-test, Pearson correlation, ANOVA, Spearman non-parametric test, Cronbach's Alpha test and stepwise regression test. The results showed that 93.9% of the variance in design quality was explained by the five independent variables, and that financial resources was the most significant predictor of design quality at 92.4%. Several recommendations have been made to enhance design quality.

1. Introduction

Software design entails a multiplicity of tasks. In the software development life cycle, or software process, the design phase is difficult to measure. Structuring a high-quality design continues to be an inadequately defined process. And, to speak of design quality becomes even more confusing, since the Webster's Unabridged Dictionary has ten different meanings of the word "quality." [1]

Now, that we know the word "quality" can be defined in many different ways we must first decide how to define this equivocal word before we can further discuss design quality within the software process. In this paper a relevant definition to the nature of this study indicates that quality is "any character or characteristic which may make an object good or bad." Therefore, the feature "good or bad" ought to be related to the specifications of the object to be designed. The object/design will be considered good if and only if the specifications are satisfied and bad otherwise. [1,2]

1.1. Background information

Within the software design phase there are basically three procedures that take place. These procedures are the architectural design, detailed design, and design testing. The input to the design processes is the specification document that provides the description of what the product is to do. The output is the design document, which is the description of how the product is to accomplish its stated goal(s). So one must realize that if the specification of the product is erroneous in any way even the most resourceful design will fail to satisfy the customer and will only supply the right answer to the wrong question. Within the design phase it is imperative to review the specifications, in other words, the quality assurance of the design must include checking that the client's specification implies the actual user requirements [3]. However, one cannot measure the quality of software design based only upon the client's judgment but rather it must be measured with some type of metrics. [4]

During the architectural design, a modular decomposition of the product is developed. That is the specifications are carefully analyzed, and a module structure that has the desired functionality is produced. The output from this activity is a list of the modules and a description of how they are to be interconnected and how will they interact. From the perspective of abstraction, what is going on during architectural design is that the existence of certain modules is assumed and the design is then developed in terms of those modules.

The next step is detailed design also known as modular design, during which each module is designed in detail. Once again, from the viewpoint of abstraction, during this step the fact that the modules are interconnected to form a complete product is ignored. [4]

This two-stage process is characteristic of abstraction. First, the high-level design, which is the general product, is designed in terms of modules that do not yet exist. Then each module is designed without regard to its being a component of the complete product. [4]

The final stage is testing, which is a vital part of design just as it is a vital part of the entire software development and maintenance process. One must realize that testing is not something that is performed only after the architectural design and detail designs have been completed [4]. "It has been widely reported that the cost penalty for error varies greatly, depending on the phase of the development cycle at which the error is introduced" [3]. Therefore, it has become essential and constructive to control quality throughout the development cycle, right from the earliest stages in order to predict end-product quality [3].

The purpose of testing at the design phase is to verify that the specifications have been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. It is important that any faults in the design are detected before coding commences, otherwise, the cost of fixing the faults will be considerably higher. Detection of design faults can be achieved by means of design inspections, as well as design walkthroughs [4].

1.2. Importance of this study

The importance of this study is to ascertain whether certain metrics should be considered more imperative than others within the design phase. If we can conclude that certain metrics appear to have a greater affect on the design quality and assurance it will be a practicable use to software engineers since they will be able to reduce overall project cost and development time. The issues raised will lead to important developments in making designs more understandable, reusable, maintainable, modifiable, portable, testable, and efficient [5].

1.3. Study objectives

The objective of this study is to investigate the quality assurance within the design phase of the software process. The goal of which is to identify the significant design metrics, such as availability of financial resources, methodology adopted, architectural style implemented, employees' training, and employees' involvement that have a direct, positive or negative, impact on the quality of design and on the quality assurance of the software design, principally with an analysis to reduce and minimize development cost and to assure customer satisfaction.

1.4. Problem statement

To what extent does the methodology adopted, architectural style implemented, availability of financial resources, employees' training, and employees' involvement affect the quality assurance within the design phase of the software process?

1.5. Literature review

Requirements specification plays a vital role in determining the quality of the end product due to the fact that if the specifications of the product are wrong even the most resourceful design will disappoint the customer. However, if the specifications are correct, the decisive factor is design. "No matter how clearly one plans the production process, the wrong design will fail the product"[3].

Acknowledgment of the importance of the early stages of the development process has come astonishingly slowly, but in today's industry a great deal of endeavor has been dedicated to advance the development and use of "design methodologies" and to the establishment of formal specification methods into software and systems engineering practices [3, 6]. The reason being that it is becoming widely recognized that mistakes committed and poor decisions made during these initial stages produce the most costly and intricate problems [6]. In other words, the cost penalty for error varies greatly depending on the phase of the development cycle at which the error is introduced as seen in figure 1 [3, 4].

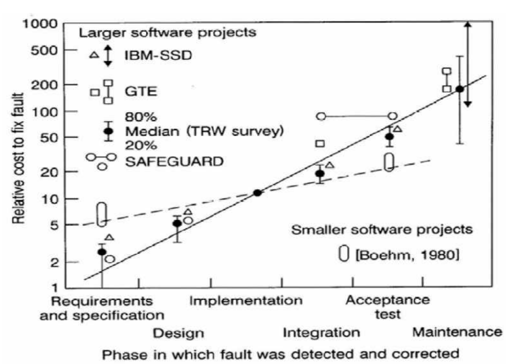

Figure 1 Relative cost of fixing a fault at each phase of the software life cycle. The solid line is the best fit for the data relating to the larger software projects, and the dashed line is the best fit for the smaller software projects.

"The reason that the cost of correcting a fault increases so steeply is related to what has to be done to correct a fault" [4]. For example, in the early development life cycle the product basically exists only on paper and correcting the problem may basically mean using an eraser and pencil but if the product has already been delivered to the customer, there will be a need, at the very least, to edit the code, recompile and relink it, and also retest it [4].

Studies have shown that between 60 and 70 percent of all faults detected in projects are specification and design faults. Specification and design faults constitute for such a large percentage of all faults within the final product, making it more important to implement design quality measures [4]. This means that by implementing design quality assurance in software systems and adopting proper design metrics, which have become a key element in the development process due to their potential to provide feedback [6], developers can avoid added cost to the project and reduce the product's development time by ensuring that the correct measurements are taken from the beginning and before actual coding commences.

Decisions concerning the architectural structure of the design has a major bearing upon many significant behaviors of the resultant software, particularly the degree of development involvement required, reliability, reusability, understandability, modifiability, and maintainability of the final product [5, 6]. All of these qualities play an important role in assessing the overall design quality and may be affected positively or negatively based upon the architectural approach implemented during this phase.

Studies have found that the structural factors are highly related to the design quality when the distribution of information flow is held to be significant as opposed to the absolute number of flows [6]. However, when discussing the overall information flow metrics, it is believed that the informational fan-out metrics and information flow complexity metrics are the most useful since they are available earlier in the development life cycle as opposed to the branch counts or lines of code [7].

Lastly, some believe that better communication is the key to a better design. By implementing an open company standards approach or open community approach in which many can give feedback to the designers, in order to make improvements or better the design as well as fix any minor problems themselves, the design quality will be improved [8].

2. Theoretical framework

The variable of primary interest to this research is the dependent variable of software design quality. Five independent variables are used in an attempt to explain the variance in the quality of software design within the software process. These five variables are: availability of financial resources, methodology adopted, architectural style, employees' training, and employees' involvement. As inferred from the literature, management expertise is expected to play a moderating role between methodology adopted, architectural style and design quality.

2.1. Study model

Following, in figure 2, is the representation of the theoretical framework in the form of a study model.

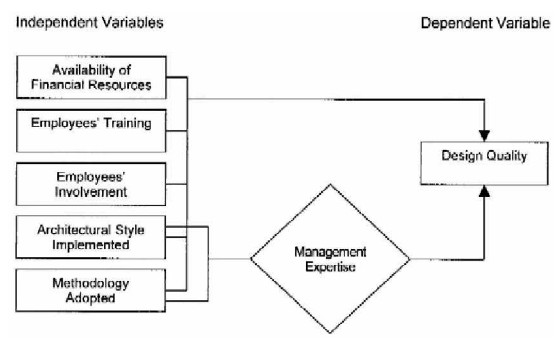

Figure 2 Representation of the theoretical framework as a study model

2.2. Dependent variable definition

Based on the literature review, the following measures are extracted to examine the variables defined in the theoretical framework adopted in this study

Software Design Quality measures:

High quality designs should have attributes that lead to quality products. These attributes encompass understandability, reusability**,** maintainability**,** modifiability, portability, testability, and efficiency. In an ideal software design, relationships between modules should also show loose coupling and tight cohesion, two important design quality measures.

2.3. Independent variables definitions

Type of Methodology Adopted

Three major software development methodologies were assessed in this study as to address their different impacts on software design. These methodologies are:

- Structured based methodology
- Object oriented methodology
- A combination of structured based and object oriented methodologies.

Architectural Styles Implemented

Six major architectural styles were considered in this study as to assess their various impacts on quality of software design. These styles are:

- Pipes and Filters
- Implicit Invocation
- Layering
- Process Control
- Client-Server
- Object Oriented Design

Availability of Financial Resources

Availability of financial resources was measured in terms of the following aspects:

- Company capital is adequate to cover the IS department's projects
- There are no shortage in resources and quality assurance tools/equipment are available
- Employee's salaries and benefits are competitive

Employees' Training

The level of employees' training was measured in terms of the following aspects:

- Employees are qualified to perform assigned tasks
- Employees are trained properly in their job responsibilities
- Employees are continuously learning new technology

Employees' Involvement

The extent to which employees are involved and committed to the requirements of software design was measured in terms of the following aspects:

- Employees are dedicated to their jobs and assigned tasks
- Employees enjoy working for the company
- Employees work hard and achieve expected goals

3. Hypothesis

From the theoretical framework discussed above, six hypotheses were formulated for this study. They are as follows:

3.1. General hypothesis

Ho There is no relation between the independent variables as a whole and the dependent variable.

H_1 There is a relation between the independent variables as a whole and the dependent variable.

3.2. Subsidiary Hypotheses

Methodology Adopted and Design Quality

Ho There is no significant difference in software design quality among object-oriented design, structured design and a combination of these two methodologies.

H_1 There is a significant difference in design quality among object-oriented design, structured design and a combination of these two methodologies.

Architectural Styles Implemented and Design Quality

Ho There is no significant difference between the distinct types of architectural styles implemented on design quality.

H_1 There is a significant difference between the distinct types of architectural styles implemented on design quality.

Availability of Financial Resources and Design Quality

Ho There is no relationship between availability of financial resources and design quality.

H_1 There is a relationship between availability of financial resources and design quality.

Employees' training and Design Quality

Ho There is no relationship between employees' training and design quality.

H_1 There is a relationship between employees' training and design quality.

Employees' involvement and Design Quality

Ho There is no relationship between employees' involvement and design quality.

H_1 There is a relationship between employees' involvement and design quality.

4. Method section

4.1. Study population

The population of this study is drawn from the frame population of programmers from two different transportation companies. This frame population totals a number of thirty programmers, fifteen from each company.

4.2. Study sample

The researchers have adopted the comprehensive survey approach where the entire population from two different transportation companies became the sample. This in which was adopted can be attributed to the following reasons:

- The number of population elements is relatively small.
- Time restrictions.
- To reduce standard error and increase degrees of freedom which lead to better reliable results in hypothesis testing, since the ideal sample size should range from 30-500 subjects.

Therefore the sample size is 55 programmers.

4.3. Data collection methods

Primary data collection methods have been used and adopted. A special questionnaire was developed to cover all dimensions of the study variables, methodology adopted, architectural style implemented, availability of financial resources, employees' training, employees' involvement, and design quality.

4.4. Sample characteristics

The total number of the population was 55 programmers and the respondents were 30 programmers. This number reflects some communication difficulties that were encountered during data collection phase. Some programmers were on vacation or unavailable, while others did not wish to participate or did not respond.

4.5. Statistical methods used

- Cronbach's Alpha test for reliability.
- Simple and multiple regression testing on the assumption of normal distribution.
- Step-wise regression test to examine the power of the model. In other words the ability of the independent variables to explain the dependent variable.
- Person Correlation Matrix tests.
- Analysis of variance to examine the explanation significance.

4.6. Reliability of measurement tools

The Cronbach's alpha coefficient was obtained for the six variables. The result indicates that Cronbach's alpha for the six variable items was 90.37%. Cronbach's alpha for the 30 questionnaire items was 98.51%.

Reliabilities less than 60% are generally considered to be poor, those in the 70% range, to be acceptable, and those over 80% to be good. The closer the reliability coefficient gets to 1.0 (i.e.: 100%), the better. Therefore, the internal consistency reliability of the measures used in this study can be considered to be very good [9].

5. Results

5.1. General Hypothesis Testing

The calculated value for *F* regarding this hypothesis was 35.743: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variables as a whole and the dependent variable.

5.2. Subsidiary Hypotheses Testing

Methodology adopted and Design Quality

Calculated statistical test results for this hypothesis where as follows:

F	R^2	Correlation
4.336	**.134**	**.366**

The calculated value for *F* regarding this hypothesis was 4.336: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variable and the dependent variable. This result is clear because the value of R^2 is equal to .134. This implies that the change in methodology adopted could explain 13.0% of the change in design quality.

The positive correlation relation, .366 between the dependent and the independent variables shows that they are relatively and positively correlated. As a result, the type of methodology adopted affects design quality.

Architectural Styles Implemented and Design Quality

Calculated statistical test results for this hypothesis where as follows:

F	R^2	Correlation
27.415	**.495**	**.703**

The calculated value for *F* regarding this hypothesis was 27.415: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variable and the dependent variable. This result is clear because the value of R^2 is equal to .495. This implies that the change in architectural style implemented could explain 49.5% of the change in design quality.

The positive correlation relation, .703, between the dependent and the independent variables shows that they are highly and positively correlated. As a result, the type of architectural style implemented has a critical impact on design quality.

Availability of Financial Resources and Design Quality

Calculated statistical test results for this hypothesis where as follows:

F	R^2	Correlation
163.795	**.854**	**.924**

The calculated value for *F* regarding this hypothesis was 163.795: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variable and the dependent variable. This result is very clear because of the high value of R^2 that is equal to .854. This implies that the availability of financial resources could explain 85.4% of the change in design quality.

The positive high correlation r (.924), between the dependent and the independent variables shows the significant role financial resources play in assuring quality in software design. It is also clear that the more financial resources became available the more quality in software design will considerably improve.

Employees' training and Design Quality

Calculated statistical test results for this hypothesis where as follows:

F	R^2	Correlation
121.459	**.813**	**.901**

The calculated value for *F* regarding this hypothesis was 121.459: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variable and the dependent variable. This result is very clear because of the high value of R^2 that is equal to .813. This implies that employees' training could explain 81.3% of the change in design quality.

The positive high correlation relation, .901, between the dependent and the independent variables shows that well-trained employees can make a tremendous difference in software design quality.

Employees' involvement and Design Quality

Calculated statistical test results for this hypothesis where as follows:

F	R^2	Correlation
30.292	**.520**	**.721**

The calculated value for *F* regarding this hypothesis was 30.292: F (calculated) > F (tabulated)

Thus we reject Ho. This implies that there *is a relation that has a statistical significance* between the independent variable and the dependent variable. This result is clear because of the value of R^2 that is equal to .520. This implies that employees' involvement could explain 52.0% of the change in Design Quality.

The positive high correlation relation, .721, between the dependent and the independent variable shows that the degree of staff involvement, dedication and commitment to software design tasks can significantly impact its quality.

5.3. Stepwise Regression Analysis

To test the power of the model we have used stepwise regression method where the rank of the interpretation coefficients (R^2) was as follows:

Rank	Independent variables	R^2
1	Availability of financial resources	0.854
2	Employees' training	0.813
3	Employees' involvement	0.520
4	Architectural style implemented	0.495
5	Methodology adopted	0.134

This implies that availability of financial resources is the most explanatory variable in interpreting the software design quality.

As a result giving the variables with higher R^2 more concern should have a positive impact on design quality. The independent variables as a whole explain 0.939 of the design quality. This high degree of explanation is enough to judge the power of the independent variables in their degree of impact on the dependent variable. Clearly, by simply improving the levels or degrees of the independent variables one can significantly anticipates an increase in design quality.

6. Conclusions and Recommendations

6.1. Conclusion

The main conclusions of this study can be summarized as follows:

1. There were many relations that have statistical significance between the independent variables and design quality due to regression.
2. All independent variables showed positive correlation with design quality. This implies that each of the factors in our theoretical framework has a positive impact on design quality. However, these correlations varied widely between these variables.
3. Availability of financial resources had the highest correlation with design quality and employees' training was the second. This indicates that when these two factors receive more concern, design quality is expected to be significantly and positively affected. While, methodology adopted had the lowest correlation it is still high enough to have to be considered as a factor in influencing design quality.
4. Because availability of financial resources and employees' training explain the highest percentage of design quality, and have the highest positive correlation with it, concentrating on them is extremely critical to improve the design quality of software products.
5. The independent variables as a whole explain 0.939 of the design quality. This high degree of explanation is sufficient to judge the power of the independent variables in their degree of impact on the dependent variable. Consequently, the combined effect of all the independent variables is as important as the effect of each one of them separately.
6. The issues raised by this study have lead to increasing importance in making software designs more reusable, maintainable, understandable, modifiable, portable, testable, and efficient.

6.2. Recommendations

1. Software design quality assurance is critical since increasing it will reduce overall project cost and development time.
2. All factors in this study should receive adequate attention from software developers, project managers and business firms. According to the study results, improving any of these five factors software design quality should improve greatly, since all of them are positively correlated to design quality. However, it seams that the most critical factors are availability financial resources and employees' training, which should receive the highest priority.
3. More studies should be conducted to determine other factors that may also have a significant influence on software design quality.
4. Since most of the measurements of this study are business-oriented, we can generalize the importance of these five factors on any other business sector. However, further studies should be conducted to examine any significant differences that might have distinguished the transportation sector from other business sectors as for the importance of the factors considered in this study.

7. Acknowledgments

This work was supported by New Jersey Information-Technology Opportunities for the Workforce, Education and Research (**NJ I-TOWER**), a New Jersey Institute of Technology (NJIT) project funded by the New Jersey Commission on Higher Education "High Technology Workforce Excellence Grant" initiative, award # 01-801020-02.

8. References

[1] Aas, E.J.; Steen, T.; Klingsheim, K., "Quantifying design quality through design experiments," *IEEE Design & Test of Computers*, Volume: 11 Issue: 1, Spring 1994, Page(s): 27 – 37.

[2] Aas, E.J. , "Design quality and design efficiency; definitions, metrics and relevant design experiences," *Quality Electronic Design, 2000. ISQED 2000. Proceedings. IEEE 2000 First International Symposium on*, 2000, Page(s): 389 – 394.

[3] Kaposi, A.A.; Myers, M., "Quality assuring specification and design," *Software Engineering Journal*, Volume: 5 Issue: 1, Jan. 1990, Page(s): 11 –26.

[4] Schach R. S., *Software Engineering with Java*, Irwin/McGraw-Hill, New York, NY, 1997.

[5] Pfleeger, L.S, *Software Engineering Theory and Practice*, Prentice Hall, Upper Saddle River, NJ, 1998.

[6] Shepperd, M., "Design Metrics: An Empirical Analysis," *Software Engineering Journal 5*, January 1990, Page(s): 3-10.

[7] Kitchenham, B.A.; Pickard L.M.; Linkman S.J., "An Evaluation of Some Design Metrics," *Software Engineering Journal 5*, January 1990, Page(s): 50-58.
[8] Goldman, R.; Bartleson, K., "Tool interoperability is key to improved design quality," *Quality Electronic Design, 2000. ISQED 2000. Proceedings. IEEE 2000 First International Symposium on*, 2000, Page(s): 407 –410.
[9] Sekaran, Uma., "Research Methods For Business ", John Wiley and Sons Inc.,2nd edition , N.Y, 1992, p 287.