

Quality Quantification Framework

Aradhana Yadav

Research Scholar, Department of Computer Science, Sai Nath University, Ranchi.

Dr. Birendra Goswami

Professor, Department of Computer Science, Sai Nath University, Ranchi.

ABSTRACT

Within monetary and time table regulations, software program development is supported and encouraged via measuring Quality at the layout level of the improvement cycle. It is highly-priced and vulnerable to blunders to modify the layout as soon as coding has started out that allows you to increase Quality. A Quality Quantification Framework for object-oriented software program at the design degree (QQFOOD) is proposed in this chapter in light of the heritage records. The suggested framework links design attributes and metrics, as well as Quality components and item-orientated design metrics. The created framework bridges the gap among metrics, Quality, and features of item-orientated layout. The framework makes it viable to create fashions for evaluating item-oriented software program's Quality early within the layout procedure, which helps to provide a dependable stop product within financial and agenda regulations.

INTRODUCTION

Below is a listing of problems that induced us to create a Quality Quantification framework:

- The modern-day object-orientated design (OOD) Quality models and metrics lack specificity, which restricts their applicability.
- The absence of a Quality Quantification approach for OOD best evaluation that consists of clear layout-level metrics.
- Validation of quality attributes and values to assist designers create nice-pushed designs.
- Provide step-with the aid of-step steering and instructions on the way to measure OOD Quality and construct models to produce what clients want.

FRAMEWORK IMPACT

The following are the framework's affects: • It gives a useful version introduction tenet for gauging item-oriented software program's Quality in the course of the design phase.

- It can be useful to become aware of the tendencies of item-oriented layout instead of gauging the Quality of software.
- Assessing a chunk of software program's Quality can assist decorate item-oriented layout development methodologies.
- It aids in the improvement of reliable, testable, and software program.
- It allows the process of growing software that satisfies software requirements.

QUALITY QUANTIFICATION FRAMEWORK (TEF^{OOD})

Estimates are viewed as tips or gadgets through the object-oriented Quality Quantification framework, which assesses the effectiveness and Quality of quantifying procedures. In the quantification and prediction stages of the software program Quality Quantification procedure, the counseled Quality Quantification framework is carried out. This framework may be used as a guide for academics and enterprise specialists to create fashions that examine software Quality at the design degree of the development life cycle.

That is the prescriptive framework in Figure three.1. There are seven levels to the suggested framework: The proposed framework's seven stages are blanketed in below figure.

Recognition of Quality Factor

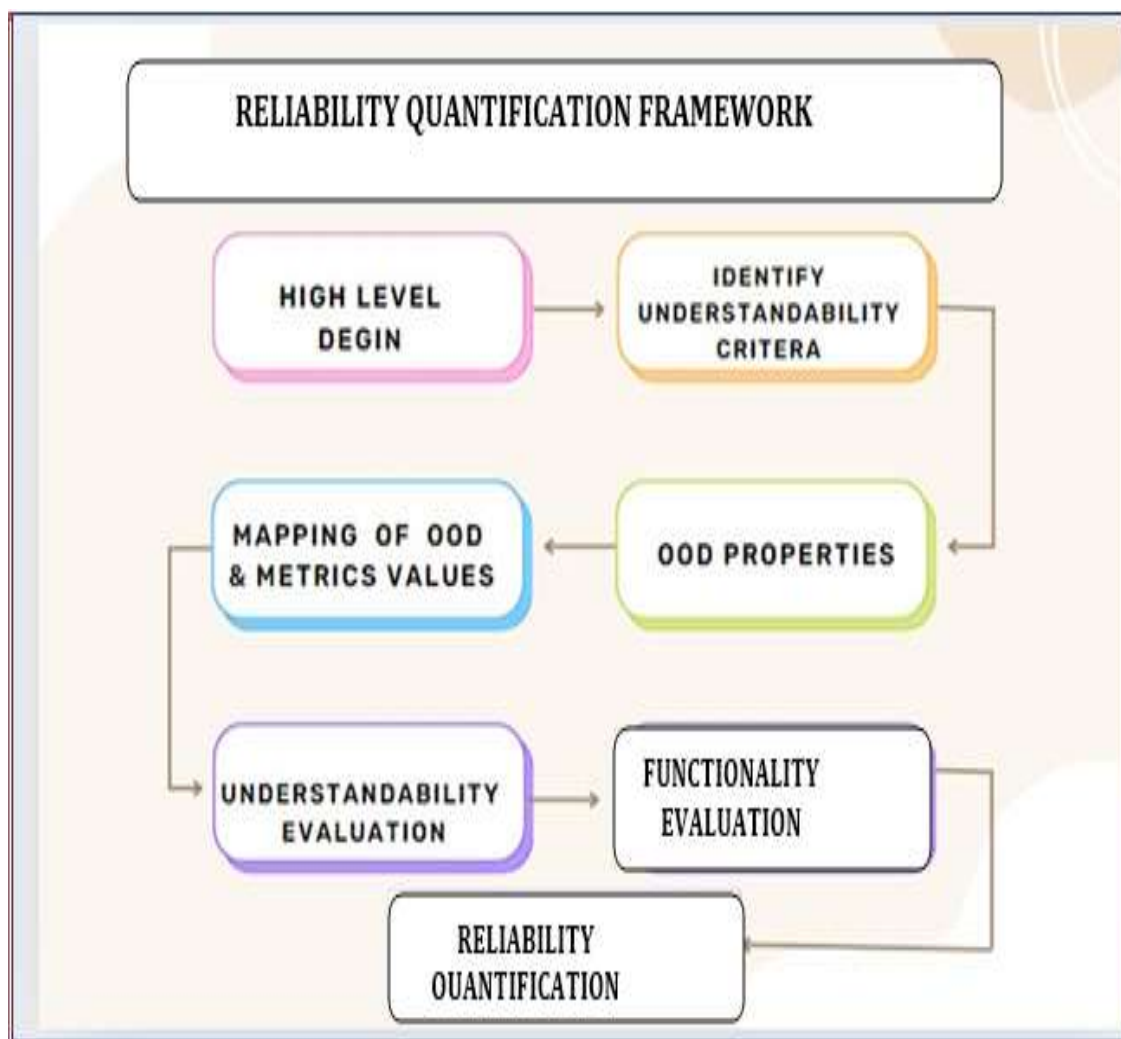
Quality has a big effect on software program exceptional. The direct Quantification variables ought to be understood on the way to calculate Quality. Every element of Quality desires to be documented at this stage. Finally, the layout section additives are completed, taking into consideration their effect on general Quality.

Object Oriented Software Program Characterization

Throughout this level, aspects of object-oriented software program that improve Quality estimates are located. The four key components of object-orientated design are inheritance, coupling, brotherly love, and encapsulation. It also appears at the contributions made by means of each feature to the layout.

Design Metric

An object-orientated layout's Quality can only be determined by way of cautiously deciding on your measurements. To make a category testable, software metrics are measurements which can be used to quantify exceptional components of software program merchandise, software program improvement assets, and software program development strategies. At this stage, your layout properties need to be covered via a set of object-orientated metrics.



Quality Quantification Framework (QQF^{OOD})

Correlation Establishment

Within the recommended framework, that is the most crucial segment. We determine the connection among item-oriented design functions and Quality elements on this step. We may additionally verify the Quality factors related to the design capabilities by drawing a regression line the usage of the design metrics.

Quality Quantification

This degree entails growing a model based totally on the Quality factors to assess the software's Quality the usage of the tried-and-proper hierarchical more than one regression evaluation. Statistical strategies should be employed to verify and validate the connection between those variables and Quality.

Finalization

After this step's outcomes are analyzed, design adjustments can be counseled. The recognized designs must be reviewed and modified in mild of the findings.

Design Assessment

Every layout is tested and stepped forward for Quality primarily based on the finalization phase results. A detailed examination of the layout traits is important to gain an acceptable index value.

Review and Revision

Many critiques and revisions occur at each degree of the Quality Quantification framework. This method entails reviewing and revising every stage of the framework. This step consists of all pointers and upgrades. The evaluation and revision can keep in mind the modifications that occur at every stage.

Summary

We provide a Quality Quantification approach for item-orientated design in this bankruptcy. A sequence of moves that may be finished to expand a Quality-size version are mentioned on this framework. The framework has seven steps: Quality factorization, object-orientated software characterization, metrics discovery, correlation establishment, Quality Quantification, and finalization, in addition to the commonplace step of layout verification. The sequence wherein every phase of the Quality Quantification method is achieved is made clean with the aid of the recommended framework. A widespread approach for creating Quality Quantification fashions is furnished with the aid of the framework. We cowl Understandability Quantification in the approaching bankruptcy.

REFERENCES

1. Lyu, M. R. (2007). Software Reliability Engineering: A Roadmap. In *Future of Software Engineering* (pp. 153-170). IEEE. DOI: 10.1109/FOSE.2007.24.
2. Arora, D., Khanna, P., Tripathi, A., Sharma, S., & Shukla, S. (2011). Software Quality Estimation through Object-Oriented Design Metrics. *International Journal of Computer Science and Network Security*, 11(4), 100-104.
3. Abdullah, Dr., Srivastava, R., & Khan, M. H. (2014). Modifiability: A Key Factor to Testability. *International Journal of Advanced Information Science and Technology*, 26(26), 62-71.
4. Rooney, J. P. (1994). Customer Satisfaction. In *Proceedings of the Annual Reliability and Maintainability Symposium* (pp. 376-381).
5. Musa, J. D., & Okumoto, K. (1984). A Logarithmic Poisson Execution Time Model for Software Reliability. *Bell Labs Technical Journal*, 63(4), 230-238.

6. Shooman, M. L. (1987). Yes, Software Reliability Can be Measured and Predicted. *Polytechnic University, Division of Computer Science*, 121-122.
7. Abdullah, Dr., Srivastava, R., & Khan, M. H. (2014). Testability Measurement Framework: Design Phase Perspective. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(11), 8573-8576.
8. Sharygina, N., Browne, J. C., & Kurshan, R. P. (2011). A Formal Object-Oriented Analysis for Software Reliability: Design for Verification.
9. Weber, P., & Jouffe, L. (2006). Complex System Reliability Modelling with Dynamic Object-Oriented Bayesian Networks (DOOBN). *Reliability Engineering and System Safety*, 91(2), 149-162.
10. Tsantalis, N., & Chatzigeorgiou, A. (2005). Predicting the Probability of Change in Object-Oriented Systems. *IEEE Transactions on Software Engineering*, 31(7), 601-614.
11. Cristescu, M., & Ciovisa, L. (2010). Estimation of the Reliability of Distributed Applications. *Informatica Economică*, 14(4), 19-29.
12. Abdullah, Dr., Srivastava, R., & Khan, M. H. (2013). Testability Estimation of Object Oriented Design: A Revisit. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(8), 3086-3090.
13. Yadav, A., & Khan, R. A. (2011). Reliability Estimation of Object Oriented Design. *IUP Journal of System Management*, IX(2), 28-41.