Initial Post

Initial Post

by Jordel Davidson-Swann - Tuesday, 26 March 2024, 12:45 AM

The article by Padhy et al. (2018) brings forth an analysis on reusability of assets in object-oriented programming. In this article, 11 separate assets that influence the reusability of a piece of object-oriented software.

Below are the components that I have found most important:

1. Requirement Analysis (RA): It is important to gather the requirements that are necessary for the project. Without this, the project would be incapable of getting off the ground.

2. Knowledge Requirement (KA): The experience and knowledge that has been gathered in one project can always be reused in another in some way shape or form.

3. Architecture driven approach (ADP): When thought is given on the overall structure of a project, it can be said that it is possible to reuse this for another project as they may have a similar feature.

4. An algorithm used in the program (AP): Algorithms can be reused in the design of various programs which has the effect of increasing the efficiency of the project.

5. Documentation in Project (DIP): The documentation of past and current projects is very important in the reuse of software components.

6. Design Patterns (DP): Reusing design patterns will prove effective as it can help in the creation of new projects.

7. Modules in the program (MIP): models promote modularity and separation of tasks which enables developers to reuse the components themselves within a given project.

In my opinion, the other factors such as: Used in data projects (UD), Modules in the Project (MP), Test Cases (TCTD), and Service Contacts (SC) are not as important as the rest of factors that were mentioned above, hence they can be interchanged in ranking between each other.

Padhy, N., Satapathy, S., Singh, R.P. (2018) State-of-the-art object-oriented metrics and its reusability: a decade review. In Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 1, 431-441. Springer Singapore.

Permalink Reply

Re: Initial Post

by Oi Lam Siu - Wednesday, 27 March 2024, 6:51 AM

Hello Jordel,

Your analysis provides a clear understanding of the factors impacting reusability in object-oriented software. I agree with yo the first priority is Requirement analysis (RA) as it is the foundation for project initiation and ensures the identification of reusable components and modules early in the software development life cycle. Other than our common view on the first priority, I also agree that service contracts (SC), while important for communication between developers and users, are not direct factors influencing reusability.

Maximum rating: -

Permalink Show parent Reply

Re: Initial Post

by Anda Ziemele - Tuesday, 2 April 2024, 11:49 PM

Hi Jordel,

Thank you for sharing your opinion on the assets listed by Padhy et al. (2018). It is valuable to observe your ranking of Requirements Analysis (RA) as first, similarly to our peers. As I have alluded to earlier, given how time and resource-consuming this process can be among teams, there is a lot of value in reusing requirements internally among projects within an organisation, and this appears to be validated in existing research (Carillo de Gea et al., 2016). However, this raises the question on whether this increases the risk of duplication, therefore RA should be reused with caution. However, I would like to challenge the lower ranking of Design Patterns (DP). DPs are industry and application-agnostic, and continue to be applied in innovative contexts (Bellavista et al., 2023) for multiple decades since their release (Gamma et al., 1995), therefore are now considered common practice. However, it has been found that certain DPs and also the way DPs are implemented and documented may affect software quality, both positively and negatively (Feitosa et al., 2019; Wedyan & Abufakher, 2020).

Bellavista, P., Bicocchi, N., Fogli, M., Giannelli, C., Mamei, M. & Picone, M. (2023) Requirements and design patterns for adaptive, autonomous, and context-aware digital twins in industry 4.0 digital factories. Computers in Industry, 149: 103918.

Carrillo de Gea, J.M., Nicolás, J., Fernández-Alemán, J.L., Toval, A. & Idri, A. (2016) Are the expected benefits of requirements reuse hampered by distance? An experiment. SpringerPlus, 5: 1-26.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995) Design patterns: elements of reusable object-oriented software. Pearson Deutschland GmbH.

Feitosa, D., Ampatzoglou, A., Avgeriou, P., Chatzigeorgiou, A. & Nakagawa, E.Y. (2019) What can violations of good practices tell about the relationship between GoF patterns and run-time quality attributes?. Information and Software Technology, 105: 1-16.

Wedyan, F. & Abufakher, S. (2020) Impact of design patterns on software quality: a systematic literature review. IET Software, 14(1): 1-17.

Permalink Show parent Reply

Re: Initial Post

by George Koridze - Monday, 8 April 2024, 11:06 PM

Peer Response

You have articulated the importance of various assets well for reusability of object-oriented software as identified by Padhy et al. (2018). I agree with the emphasises of the foundational role of Requirement Analysis (RA), Knowledge Requirement (KA), and Architecture-Driven Approach (ADP).

I also agree with your insight on the Knowledge Requirement (KA), emphasizing that experience and knowledge accumulated from past projects are invaluable for future endeavors. This asset is indeed critical as it underpins the creation of a knowledge base that can significantly reduce the learning curve and time to market for new projects.

It may also be beneficial to consider giving higher priority to Documentation in Project (DIP) and Design Patterns (DP). Comprehensive documentation ensures that the software can be understood, maintained, and reused by someone other than the original developer, especially if the engineering teams change, and this is a critical aspect of reusability. Similarly, design patterns provide solution frameworks that can be effectively applied to similar recurring use cases, making them indispensable tools for developers seeking to apply reusable concepts.

While you have given Test Cases/Test Design (TCTD) less priority, I believe it is also important for software reusability. Testing the solution is an indispensable part of the development process, and standardising test cases is an asset that can save considerable time when new components are developed or existing ones are integrated into new projects.

References:

Padhy, N., Satapathy, S., & Singh, R.P. (2018) 'State-of-the-Art Object-Oriented Metrics and Its Reusability: A Decade Review', in: Satapathy S., Bhateja V., Das S. (eds) Smart Computing and Informatics. Smart Innovation, Systems and Technologies. 77. Springer.

Permalink Show parent Reply

◄ Initial Post Summary Post ►