Sufficient Level of Software Maintainability

dr. Adam Vanya

INTRODUCTION

In our previous blog post titled “Measuring Software Maintainability”, we explained why it is important to measure the maintainability of software systems and how it can be done. Based on the measurement results, some questions arise. For instance: Is the measured level of maintainability sufficient? Which level of maintainability should be targeted? How far is the measured level of maintainability from the targeted one? By when do we want to reach the targeted level of maintainability? Answering these types of questions is crucial since the answers given determine the decision and actions one must take to properly control maintainability. If you are interested in those answers please keep on reading through this blog post.

WHEN IS THE LEVEL OF MAINTAINABILITY SUFFICIENT?

The answers to the questions in the introduction depend a lot on the circumstances and the goals of the stakeholders. Let us consider, for example, the situation when a given software system has served its purpose but changing it according to the drastically changed requirements would cost more money than building a new system from scratch. In such cases, the life of the software system will come to an end soon. Therefore, investing money into the new system may be more important than increasing the maintainability of a system that will be phased out soon anyway.

Software systems are built with the intention to generate as high profit as possible and for as long as possible. To achieve this goal, the vicious cycles of maintainability have to be avoided (see also our blog post titled “The Effects of Software Maintainability”). Not doing so would quickly drive the software system into a legacy state where modifications become unacceptably long and expensive. Maximizing the lifetime of a newly built software system can therefore be supported by aiming for a maintainability level that is higher than market average. This aim is also reasonable considering that new technologies used to build new systems allow the production of source code that is better maintainable. The longer we want a software system to keep generating profit, the higher maintainability level should be targeted and the more should be invested into maintainability.

Once indirect maintainability measurement results (related to source code properties and developers) have hit the targeted level, direct maintainability measurement results (indicating development process efficiency) can help us decide whether the efficiency with which developers are implementing changes and bug fixes is sufficient or not. Based on these observations, maintainability goals can be fine-tuned. There is absolutely no need (and would be unwise) to stick to them without good reasons.

From a financial point of view, an acceptable balance has to be found between 1) the costs of increasing and keeping up the maintainability level of a software system and 2) the money not spent thanks to the increased level of development efficiency. When figuring out where this balance should be, we also have to keep in mind that not reaching the legacy state for a software system and therefore being forced to re-implement the whole system from scratch save us a lot of money and headaches. In order to find the suitable balance, cost measurements should be executed and the results should be processed.

The level of maintainability can also radiate a relevant marketing message. A software system which has a high level of maintainability is in a good shape and will be able to live a longer life. Since services and products often rely on software systems in the background, their availability and evolution is tightly coupled to the software they rely on. Since investors and partners value stability they prefer investing in and cooperating with companies where they see the future of the supporting software system ensured. This is the reason why companies have the maintainability of their software systems measured and certified. The resulting certification with the maintainability results can be made public and the certification ceremony may attract a lot of attention too. Since software changes quickly due to its changing environment, certificates expire after a given period of time after which one may decide about certifying the software system again.

SUMMARY AND CONCLUSIONS

One way or another, software maintainability is affecting you. It can have a significant impact on your career if you are working for a software development company and it is a major factor to consider when you invest in such companies or buy custom software. Therefore, we believe maintainability has to be understood and explicitly be controlled to achieve your goals.

To help you make informed, fact-based decisions about software maintainability, we discussed how the software and developer related measurement results should be interpreted. One such possible decision is to significantly increase the maintainability of a software system. What concrete actions we can take to achieve that goal is the topic of our next blog post.

In case there are certain things described in this blog post you do not or not completely agree with or if you think it should be extended in some ways, please feel free to contact us and tell us about your idea: adam.vanya@frontendart.com