

Investing Into Software Maintainability

dr. Adam Vanya

INTRODUCTION

Hopefully, the reader is convinced by the earlier blog posts that software maintainability is a relevant topic that needs careful attention and also actions taken. In our previous post titled “Improving Software Maintainability”, we focused on those actions which can be carried out to increase the level of software maintainability. Executing these actions costs significant effort. What we should keep in mind when reasoning about such investments is the main topic of this blog post.

REASONING ABOUT SOFTWARE MAINTAINABILITY INVESTMENTS

Decision makers need to be aware about the financial consequences of their decisions because their performance and added value to the company are judged also considering financial matters or because the return on their investment in the company depends on those consequences. In the case of software development companies, one such decision is whether to invest in improving the current maintainability level of the software developed and if so, by how much. Even if this issue remains unaddressed, an implicit decision of not investing will be made and the related decision makers can be and will be held accountable for the consequences.

Some decisions when kept implicit do not change the status quo. For instance, not thinking about changing the logo of a company will result in an unchanged logo and people will probably relate to that logo as before. Other implicit decisions, like not investing in improving the maintainability level will lead to serious financial consequences, beside others.

In [1] Bakota et al. pointed out that not putting explicit effort into increasing the maintainability level of a constantly changing and expanding software system will lead to an exponential deterioration of the maintainability level of that system. In our previous blog post titled “The effects of software maintainability”, we enumerated some of the major consequences. These include beside others a strongly decreased software development process efficiency and an increased chance that the software will need to be replaced due to reaching the legacy state.

As we see, these consequences may cost a significant amount of money that needs to be spent on a regular basis. It may also impact the business enormously. Not all companies can easily afford the reimplementation of their software systems so not caring about maintainability may even lead to an existential issue. Therefore, it is crucial to explicitly decide about software maintainability related investments.

From the described nature of software maintainability follows that even preserving the current level of maintenance efficiency requires investment. With this status quo preserving investment one can avoid the extra costs caused by maintenance activities getting slower and therefore less efficient. Furthermore, this type of investment ensures that the level of software maintainability will not become a reason for a shorter than intended lifetime of the software system developed. Money not spent on reimplementing a software system is money gained when we assess the investment described. How much time there is before the software reaches its legacy state and needs to be replaced without investing into preserving its current maintainability level depends of course on many factors. These include the current level of maintainability and how many change and feature requests will need to be implemented.

In some cases, preserving the current level of maintenance efficiency is not enough. You know this if features are implemented and bugs get fixed regularly much slower than expected. To speed up maintenance and implement more in less time, extra investment is needed on top of the preserving type of investment described above.

What can be done to preserve or to improve maintainability is described in one of our previous blog post titled “Improving software maintainability”. The actions discussed in that post have related costs that lead to the total costs of the investment. These costs include the cost of source code refactorings, the money spent on providing maintainability and related trainings to developers, architects and management, the price of products and services used to measure and monitor the level source code maintainability (internal maintainability) and the efficiency of the maintenance process (external maintainability), the effort spent on implementing and executing motivational strategies.

Just as any investment, investing into software maintainability needs to pay off. Therefore, one should compare the total costs of maintainability related investments and the gains the investment brings before deciding about the investment. Once again, the gains also include money that did not need to be spent. When calculating the gains, one also needs to know how much slower maintenance activities would get in a specified amount of time without the investment and how much faster it would become after investing. The model described in [1] can also be of help to calculate this.

Because of the described nature of software maintainability, the question is usually not whether to invest in software maintainability, but rather how much. The answer depends heavily on the organizational circumstances and needs to be evaluated with care.

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 reach your goals.

In this blog post, we argued that it is relevant to explicitly decide about investing into software maintainability. We also provided suggestions on how this can be achieved. We concluded that the emphasis is on how much to invest and not on whether to invest.

In case there are certain statements 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

REFERENCES

- [1] Bakota, T., Hegedűs, P., Ladányi, G., Körtvélyesi, P., Ferenc, R., & Gyimóthy, T. (2012). A cost model based on software maintainability. 2012 28th IEEE International Conference on Software Maintenance (ICSM), 316–325.