Business Value of a Clean Architecture

October 2014

A study by National Institute of Standards and Technology (NIST) in 2002 estimated that software quality problems cost the US economy about $60 billion per year. A more recent study by Cambridge University in 2013 found that software bugs cost the world economy $312 billion per year. Companies building complex software-centric systems have responded to these challenges in a variety of ways, such as improved requirements gathering, continuous integration and testing, and defect and vulnerability detection. In this paper, we are concerned with understanding the value of a well-understood and well-managed architecture on software quality.

Since problems in architecture do not directly point to software bugs, there is often a tendency to fix bugs without regard to the overall design. If the tests work and the software functions correctly, then why worry about the architecture? As a result, architecture erodes over time and the original design intent is lost. Software modules get coupled and become hard to understand and maintain. This in turn leads to poor quality software. The consequences of software coupling have been studied and documented in a number of research studies [1, 2].

Indeed, keeping the architecture clean may be one of the best ways to reduce defects and vulnerabilities and improve the maintainability of software. Wipro, using DSM tools, observed a 56% reduction in the number of bugs as well as improved productivity of 45% in fixing bugs. Another study at Vanderlande Industries [3] found that a clean architecture reduced the size of code that needed certification by 66%.

Understanding the Sources of Value

Systems with a clean and modular architecture are faster to develop and far easier to maintain. There are many other benefits:

- Improved developer productivity
- Improved testing efficiency
- Improved employee satisfaction
- Improved customer satisfaction
- Lower training and support cost
- Increased sales/market share

While it is easy to recognize the crucial importance of customer satisfaction or increased market share, it is hard to quantify. However, even when we ignore those sources of value that are difficult to quantify, we see that most organizations will benefit immensely from well-understood and well-managed architectures.

Computing the Value

Before we outline our assumptions, please note the following caveat: the maturity of the development organization is different in every company, as is the complexity of software systems. Therefore, you should ultimately compute the value with numbers that are appropriate for your organization.

Consider a project with 250,000 lines of code, which is being developed and maintained by a development staff of 10 (development, test, build, configuration management, etc.). According to Steve McConnell [4], Microsoft applications had about 10-20 defects per 1000 lines of code during in-house testing. IBM [5] used 10-15 serious errors per 1000 lines of code in their study. For purposes of this analysis, we will use 10 bugs per 1000 lines of code.

We will make the following conservative assumptions about the benefits of a clean and modular architecture: 15% reduction in bugs and a 10% improvement in productivity. We will further assume that, on the average, a serious bug takes 4 hours to fix. We will also assume that the cost of a developer is $150,000/year.

Savings in fixing fewer bugs

Total number of serious bugs: 250,000/100 = 2500

Reduction in the number of bugs: 2500*0.15 = 375

Hourly cost of a developer: ~$75/hour ($150,000 per year)

Total Cost Savings: 375*75*4 hrs = $112,500.

Increased Productivity

The entire team benefits from the increased visibility. Coding becomes easier, testing becomes more efficient and project managers have a better understanding of the impact of change.

Improved productivity: $150,000*0.10*10 = $150,000

This gives us a total cost saving of $262,375. The developers' cost for the project is $1,500,000. Therefore, a clean, well-defined and well-managed architecture will lead to a saving of nearly 17%. Note that in this analysis, we did not consider the many other sources of value such as improved customer satisfaction, reduced support load, etc. As a result, the actual business value is even higher.

Conclusion

A well-managed architecture is crucial for keeping complex software robust and maintainable. Most teams recognize the importance of a clean and modular design and yet the emphasis on design is lost once development starts. The need for timely delivery often leads to shortcuts. It is these shortcuts and the lack of visibility that make the software unmaintainable over time. The erosion of architecture is often considered a key part of technical debt and serves as a tax on the cost of future improvements. Lattix makes architecture management a part of your continuous integration and thereby makes architecture management easy and integral to the development life cycle.

Try Lattix Architect on your Project

Is your software complex, buggy and hard to maintain? Lattix provides support for a wide range of technologies including C/C++, Java, .NET, Fortran, Ada, Javascript, Actionscript, Pascal, Python, UML/SysML, Rhapsody, SparxEA, Oracle, SQLServer, Sybase, LDI and Excel.

Do you want to see what your architecture looks like and what you can do to modularize it? We have helped companies all over the world improve the quality of software and we can help you achieve the same results.

Contact Lattix at sales@lattix.com or call 978-664-5050.

References

1. Linking Cyclicality and Product Quality http://faculty.insead.edu/manuel-sosa/documents/9-sosa_mihm_browning_2012-137.pdf
2. Technical Debt in Large Systems: Understanding the cost of software complexity http://sdm.mit.edu/news/news_articles/webinar_050613/sturtevant_050613.pdf
3. Managing Dependencies Between Software Modules Using DSM http://lattix.com/files/wp/neeraj/Managing%20Dependencies%20Between%20Software%20Modules%20Using%20DSM.pdf
4. Code Complete: A Practical Handbook of Software Construction, by Steve McConnell
5. The Business Value of Software Quality http://www.ibm.com/developerworks/rational/library/4995.html