Challenges in Creating Exceptional Software Systems

Software engineers around the world strive for excellence when it comes to creating software systems; however, this is not a simple task. Some of the major challenges faced in this process include requirements volatility, addressing quality attributes, and carrying out a well-established design process.

The primary challenge in creating excellent software systems is the constant change of requirements. "Software's own nature allows itself to change to provide different or new functionality to systems [1]." Although this is a beautiful feature, it adds complexity to the project by impacting current development efforts. Along with this, it also causes responsibilities of project members to change in an unplanned fashion. Software architecture provides flexibility to such volatility. It helps to "consider how the application may need to change over time to address new requirements and challenges, and build in the flexibility to support this [2]." Along with this, software architecture gives engineers the ability to decompose complex systems into smaller components. "In partitioning an application, the architect assigns responsibilities to each constituent component [3]." Through these techniques, architecture plays an important role in resolving the issue of requirements volatility.

The next challenge in creating reliable software systems lies in effectively addressing the quality attributes of the system. Quality attributes "define an application's requirements in terms of scalability, usability, and so on [3]." Software architecture can provide built-in structure within parts of the system to help address specific attributes. "Quality attribute requirements are satisfied by the various structures designed into the architecture, and the behaviours and interactions of the elements that populate those structures [4]." Satisfying these requirements helps benefit the users of the system, as it provides the necessary attributes. In this way, architecture can assist in addressing the quality attributes of a system; allowing software engineers to create reliable systems for their stakeholders.

Another challenge faced in creating software systems lies in carrying out a well-established design process. Designing a product or system has many stages, and designs can constantly change. This makes it difficult for engineers to understand and document information about a design. Software architecture and patterns within the framework are beneficial in this regard. "The power of architecture patterns stems from their utility, and ability to convey design information. When an architecture is based around patterns, it also becomes easy for team members to understand a design [3]." In this manner, software architecture provides a method for improving the design process of a given system.

In conclusion, it can be seen that software architecture plays an important role in the creation of excellent software systems. Engineers use architecture in many ways to assist them, and by doing so are able to create more effective and reliable systems.

References

[1] "Software design challenges," IT performance improvement, 2012. [Online]. Available: http://www.ittoday.info/ ITPerformanceImprovement/Articles/2012-06Otero.html. [Accessed: May. 10, 2016].

[2] "Chapter 1: What is software architecture?," Microsoft developer network, 2003. [Online]. Available: https:// msdn.microsoft.com/en-us/library/ee658098.aspx. [Accessed: May. 10, 2016].

[3] I. Gorton, Essential Software Architecture. Germany: Springer, 2006.

[4] "Understanding quality attributes in software architecture," Pearson: informIT, Oct. 31, 2012. [Online]. Available: http://www.informit.com/articles/article.aspx?p=1959673. [Accessed: May. 9, 2016].