# **Standard Software Systems Development Questions**

B. Eng. Alejandro Montes Rivera

Colorado State University Global

CSC505-1: Principles of Software Development

Dr. Lori Farr

Sun Aug 22

# **The Nature of Software**

The Nature of the Software is constantly changing, the technologies that we have today are way far than what we had a few years ago. As a Software Developer, it’s quite hard to be on the edge of the technology. We must be constantly studying and learning new frameworks, languages, and programming methodologies.

According to Bruce R. Maxim “Today, software takes on a dual role. It’s a product and the delivering a product.”.

The Software Industry has become a dominant factor in economies. Every single company has dependencies on software today. From the simplest integration to the most complex and customized one.

(Pressman & Maxim, n.d.) states that the same questions asked in the past are the same ones we have nowadays even though we have Teams of software specialists, each focusing on one part of the technology required to deliver a complex application.

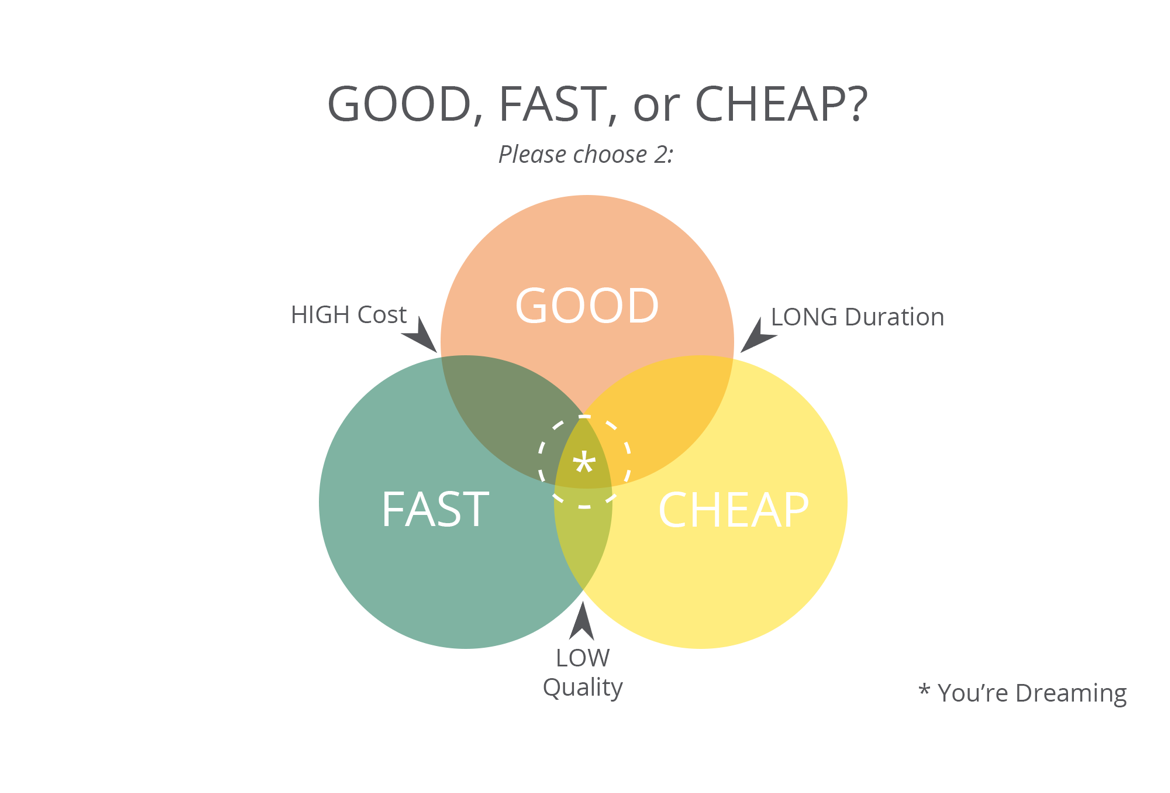
**• Why does it take so long to get software finished?**

Usually, a customer would ask for new functionality. There’s an expectation of the product to develop that may not align with the initial plan.

Some issues could come during the requirements, design, or at any point of the Project Lifecycle, negatively impacting the release date. It takes such a long time because the business logic must be transmitted, and the program would need to be adapted to meet the expectations.

**• Why are development costs so high?**

Every software development is unique and requires a different set of skills, technologies, or expertise. Depending on the requirements and the release intended plan, the price could go up or down.



**Retrieved from: The Good, Fast, Cheap Complex, June 29, 2015**

Project Management is critical while pricing a project correctly. Great software requires a strong strategy and a brilliant collaboration through all the resources. Nowadays and being Software Development on the most well-paid carriers. It would help if you considered all the guidance and time that your resources would allocate.

**• Why can't we find all errors before we give the software to our customers?**

While a common response to the software crisis is to suggest that training should be improved. Practitioners and managers do not always do what they know they should. A wrong project estimate or an accelerated release would lead to duplicated code, bad practices, and bad quality code.

Even though a good planning is done, and a testing phase is completed, there’s always the possibility of a bug in the system. However, doing the proper planning and having skilled resources would reduce drastically the support after the integration and will likely help to maintain the best practices standards.

**• Why do we spend so much time and effort maintaining existing programs?**

Modern systems are usually designed and implemented in a reusable way to be escalated or integrated into different projects.

Legacy systems, on the other hand, usually take a lot of effort to be maintained. Most of those Legacy Systems have dependencies in old libraries that are out of support. Because of that, custom integrations are constantly added.

**• Why do we continue to have difficulty in measuring progress, as software is being developed and maintained?**

There could be multiple factors, a PM should be following closely the advance with the business, but in some cases, they don’t know what they need. And they may change the requirements at any given point during the lifecycle. While Software is defined as a set of instructions that lead to a specific result, in most cases the instructions or the business logic is not well transitioned.

# **Python Script**

As a small investor, it would be awesome to use artificial intelligence to predict the market behavior for a given stock. For this practice, I decided to build a small script using yfinance library to retrieve the historical graph given a stock symbol.

“YFinance” installation: <https://pypi.org/project/yfinance/>

Text

Description automatically generated

**Screenshots**

A picture containing text, screenshot, computer

Description automatically generated

## References

David Fields (2015). The Good, Fast, Cheap Complex

Software Process Improvement: Concepts and Practices. (1999). United Kingdom

Pressman, R. S., & Maxim, B. R. (n.d.). Software Engineering: A Practitioner’s Approach.

YFinance Library: <https://pypi.org/project/yfinance/>