**Module 1, Critical Thinking Option 1**

Jesse E Stripe

Department of Computer Science, CSU Global

CSC 505-1 Principles of Software Development, Spring – A 2022

Dr. Steven Evans

Due March 27th, 2022

Module 1, Critical Thinking Option 1

## Why Does it take so long to get software finished?

Software developed for critical tasks in enterprise must exhibit a high degree of reliability and be able to cope with potentially high volumes of transactions (Pressman, 2019). From personal experience given the complexity of some applications identifying and testing all potential use cases to ensure a high quality product can be very time consuming.

## Why are development costs so high?

Development costs depend on several factors including, the scope of the project, the platform(s) to be supported, and the technology to be used (Kurilo, 2020). Generally speaking what you invest into a project tends to dictate the end quality of that project. Trying to cut corners to save money can often lead to buggy or incomplete software.

## Why can’t we find all errors before we give the software to our customers?

No matter how through we are in the definition of the uses cases and testing of the software we cannot account for the variations in the devices the software is installed on or used by. End users may also issues with the usability of the software itself that the design team did not expect.

## Why do we spend so much time and effort maintaining existing software?

Using the maintenance and engineering (M&E) system used at my employer as an example the simple answer is cost to replace it. While the system was developed back in the 1970’s the cost of adopting a new system combined with the potential compliance issues and need to change a significant portion business processes makes is prohibitive. It is more economical to maintain what we have.

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

Software is not a tangible good, the inputs used to create it may not be directly reflected in the output. Measuring hours worked or commits pushed do not equate to the complexity or volume of code produced. These measures can also lead to negative work habits potentially leading to burnout and employee turnover (Lyman, 2020).

References

Kurilo, A. (2020, February 17). What does it cost to develop new software. business.com. https://www.business.com/articles/the-cost-of-software-development/

Lyman, I. (2020, December 10). Can developer productivity be measured? Stack Overflow Blog. https://stackoverflow.blog/2020/12/07/measuring-developer-productivity/

Pressman, R., Maxim, B. (2019). *Software Engineering: A Practitioners Approach.* McGraw Hill.