Module 1: Critical Thinking

Standard Software Systems Development Questions

Nolan Byrnes

CSC505 – Principles of Software Development

Colorado State University – Global Campus

Professor Steven Evans

July 24th, 2022

Module 1: Critical Thinking

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

Pressman identifies 5 framework activities that are involved in all software projects, which include communication, planning, modeling, construction, and deployment (Pressman, 2020, pg. 10-11). In order to begin a software project, communication with the stakeholders is paramount in order to understand the stakeholders’ objectives for the project and gathering requirements that define what the features and functions should be included. A software project plan also must be created. Pressman explains that a software project plan "defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule" (Pressman, 2020, pg. 10). The software developer must then model the solution to help better understand the requirements, and what design will best accomplish the requirements. The software engineer must then construct the design that they modelled, which includes the code generation, as well as testing the solution that the developer is implementing. The developer then deploys the completed or incomplete solution to the stakeholders, and receives feedback. Pressman states that "For many software projects, framework activities are applied iteratively as a project progresses" (Pressman, 2020, pg. 10). Due to the iterative nature of the framework activities, and the amount of work that comes with each project iteration, it takes time to conclude a software project.

**Why are development costs so high?**

Development costs are high due to the amount of key players that are involved with developing the software as well as the specialized knowledge required to develop software. Software developers work with stakeholders throughout the development life cycle to gather requirements, perform user acceptance testing, and obtain the approval of the software being developed. For a one-hour meeting including the developer, project manager, and three stakeholders, if they all are being paid $100 an hour, that meeting would cost $500. This adds up for the amount of meetings that may occur during the lifecycle of the software project. Development also requires software engineers who are knowledgeable and up to date on the latest technologies being used to develop the solution to make sure that the solution being created is up to date with technological advancements, effective, and secure.

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

We cannot find all errors before handing over software to our customers due to amount of external factors that could affect the software, such as the variance in devices that the software could be run on and different versions of third-party software that the software relies on. Pinedo states that “It’s nearly impossible to account for all possible scenarios given the vast number of browsers, versions, extensions, and other factors that can impact your application’s stability” (Pinedo, 2018, para. 13). In 2015, there were more than 24,000 unique android devices, and the amount of time to test a mobile application on each of those android devices would take a large amount of effort (Mirani, 2015, para. 2).

Even if the developer tested every device, version, and configuration that the software could be run on, by the time that the software is released, new devices could be released and third-party software that the software relies on can be updated.

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

We spend a lot of time and effort maintaining existing programs due to the program’s dependencies being upgrades over time and changes in business needs. When a programs dependencies get upgraded over time, it increases the probability of introducing compatibility issues to the program. Developers will need to make changes to the code in order to resolve the compatibility issues with the new versions of the dependencies to have the software continue to work. In 2020, Python 2.7 reached its end of life, which means that that version of Python is no longer supported, even if there are discovered security issues (McLaughlin, 2019). If I had developed the Python calculator using version 2.7, I would have had to upgrade the program to run using Python 3 to ensure that the software that it is using still is supported, and will have applied bug fixes and security patches, which keeps the software stable and secure.

When business needs change, it would be easier to upgrade the existing program rather than developing an entirely new program. Let’s say company A uses the calculator I developed in Python, and all their employees have the calculator installed. If they now need additional features such as obtaining a square root of a number, it would be easier for me to update the existing code to include the new feature, rather than developing an entirely new application with the new feature. It would be easier for the employees of the company to have the current software upgraded rather than having to uninstall and install a new calculator program.

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

Edsger Dijkstra claimed “If debugging is the process of removing software bugs, then programming must be the process of putting them in” (Pinedo, 2018, para. 1). As more code is included in the software being developed, more bugs are being introduced to the code base. When I was developing the calculator in Python, I decided to take in the math problem as a string input, and split the string based on a whitespace character, which would allow me to determine which characters were the numbers, and which were the operators. This worked at first, until I realized that if I did not include spaces in the math problem, the program would be unable to separate the numbers and operators. I had this resolved by removing all whitespace from the input and placing a white space around all operators (+, -, \*, /), before splitting the string into a list. This lead to the issue that users could introduce math problems where they are adding multiple numbers, such as “1 + 2 + 3”. To resolve this, I performed a loop on the list of numbers and operators, to have the program handle multiple operators in the expression. The point is, as I was developing the program, I was introducing more bugs with each feature that I included in the program, and in my case, when I was developing, I was not anticipating these bugs from occurring while I was developing these features. It is difficult to measure progress on the software being developed because when a feature is “completed”, there are typically bugs that are also being introduced to the code that the developer may not have anticipated and will need to have resolved.

**REFERENCES**

McLaughlin, K. (2019, November 20). Switching from Python 2 to Python 3: What you need to know. Opensource.com. <https://opensource.com/article/19/11/end-of-life-python-2>

Mirani, L. (2015, August 5). There are now more than 24,000 different Android devices. Quartz. <https://qz.com/472767/there-are-now-more-than-24000-different-android-devices/>

Pinedo, K. (2018, July 25). Not all bugs are worth fixing and that's okay | Bugsnag blog. Bugsnag. <https://www.bugsnag.com/blog/not-all-bugs-are-worth-fixing>

Pressman, R. S., & Maxim, B. R. (2019). Software engineering: A practitioner's approach. McGraw-Hill Higher Education.