UNIVERSITI TUNKU ABDUL RAHMAN

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UECS2363

SOFTWARE CONSTRUCTION AND CONFIGURATION

Software Engineering

Lecturer: Dr. Farizuwana Akma Binti Zulkifle

Assignment 1: Individual Report

	Name	Student ID	Course	Practical Class
1	Tan Ying Yao	1703648	SE	P3

Summary

The talk is aptly named “The Mind Behind Linux” as it began with Chris Anderson probing into the character behind the creator of Linux, Linus Torvalds. There is an air of surprise as we find out that the headquarters for Linux is a small walking desk utilised fully by Linus. Linus prefer minimalistic design in his work environment and prefers little to no sound pollution to concentrate properly. Linus Torvalds presents himself as a mellow and solemn man that got his start through a snowballing effect when working on Linux. Eventually the project got too popular and he started applying for an open-source license for his operating system. The project became a collaborative effort when people started contributing ideas and the feedback received became very crucial in aiding the development.

However, Linus admitted he is not sociable by nature and prefers to not meet and discuss issue with the system in public but rather in his own personal space of choice, email. It was then revealed that Linus also contributed to a management system named Git and he began to explain his role in the software. Git was the solution to a multi-scale project in which each individual can provide their efforts without conflict to the system. The development of Linux and Git was an unintended consequence of Linus desire to not interact and work with too many people personally. Linus then told the public that he was a geek when he was young and very stubborn to boot. This headstrong nature leads him to working in Silicon Valley but also caused a lot of conflict with his co-workers. Linus explained that his bold nature causes people to think of him as too aggressive when he is just trying to solve the issue directly. He admits he is not a people person but he requires help in areas he is lacking in such as his inability to create an UI.

The talk then went into a different subject regarding Linus taste in coding. Linus clarifies that taste is developed personally when the code can be effectively managed in the larger picture and instinctively contributes to its longevity. When asked whether he is a visionary or not, Linus humbly said that he is merely an engineer that can only fixes task at hand. The open source nature of Linux contributed greatly to the understanding of software among the public. Linus reiterates that he is glad that he is able to contribute meaningfully and hopes that the public can rise up and become a visionary that propels the future of software development.

Trend of Software Development Industry

The advent and popularity of GitHub had propelled open source software development greatly. The adoption of Cloud service had seen a great increase as many companies starts to shift to open source in the public cloud. Open source can be used to scale companies cloud service. Cloud-based applications are more common now and runs more efficiently. Open source provides transparency and delivers a unified experience across all platforms. Using open source removes the need of licensing fees and the reduces the learning curve of a closed language. This generates innovation and results in the growth of the industry. The growth of innovation rapidly increases the pace of developing software. Contributors are able to provide multiple solutions and results in the community benefiting from the act.

Open-source security had been improved as well due to the transparency of each component involved. This allows the public to identify flaws of the system and works toward to fixing it and reduces mistakes. Many more projects become open-source to continue the maturity of the project. Open source is shaping up to be a major theme for the IT industry with lots of development adopting this trend to further their causes.

Moving on to the current software development trends, the growth of blockchain industry rises to meet the demand of cryptocurrency. The demand for blockchain developer continues to grow as a result as more startups tries their luck at this bountiful venture. There is a need for expertise in blockchain industry such as Etherium or Bitcoin experts to satiate their want for a highly-skilled blockchain developer.

There is an increase in purchasing of wearables such as Fitbit and Apple Watch which functions as data-collecting device. These devices tracks the user and collect enormous amount of data and IT companies are keep up to process these information. Edge computing is going to play an important role as it uses a mesh of micro data center to process data near the device. The process saves time and money by porting all of the data to a centralized data center. Internet of Things (IoT) Devices will be able to perform faster real-time analytics even when present in poor connection area. Advance database and network engineers will be called upon to create the infrastructure of the IoT future. Edge computing will continue to impact each and every layer of the IT industry.

Cybersecurity is the top priority for most companies nowadays as they strive to protect their confidential data. This translate to new resources flowing into new solutions to prevent any theft of data. However, there is a severe lack of cybersecurity talent as many organizations reported a shortage of cybersecurity employees. This shortage of employees resulted in consequences in big business franchise as they face risk of confidential information. Cybersecurity will be a viable path to explore for developers who want to use their skill to stay in relevancy in the foreseeable future.

Challenges of Software Development

There are various challenges in developing, delivering and maintaining a software in the current industry. The first common challenge is the infrastructure of the project. An unproven work environment can severely impact the delivery of a software project. The lack of a good work environment can result in underbudget and failure to proceed with delivery in time. To ensure there is an efficient development, the pre-production environment should be made clear and effective during development and testing phases. A solid and good infrastructure can lead to the creation of a better software development environment.

The second major issue is the constant change in requirements during development. Products with unspecific requirements presents itself as a huge challenge as the importance of requirement gathering cannot be understated in the delivery of a project. A proven process and great communication had to be established to assure that product produce is correct with expectations and requirements. The scope of the project must be handled with care along with the users need and requirements. Establishing a good communication and expectation is vital in the development phase. Major stakeholders should be involved to ensure a clear and concise requirements document is being managed properly. A prototype can solve many requirements issue and problem to refine the solution.

The third major issue currently is quality assurance among the software development. Not reviewing code or suppressing errors are bad practice by developers whom failed to meet deadlines. A certified quality assurance team is imperative for a successful development of a software product. Good code development practice should be established to prevent developers from cutting corners and to create an efficient environment.

The fourth challenge in the industry is undefined quality standards. Identifying defect is inevitable during functionality testing. This must be done even if products had gone through unit testing. The test approach, conditions and scenarios should cover all the requirements that are to be delivered by planning several cycles of testing.

The fifth challenge is the adapting to the latest market trends. We must always cater to the latest technological requirements to ensure our products are efficient. Each and every developer should be aware of the relevancy of the modern technology to ensure insights of new technology and trends are being practiced.

Design influences also rears its ugly head as an issue as stakeholders can affect the development of a product. A proper management of these influences is essential for maximizing the quality of systems and their related influences on future business opportunities. The accessibility of applications currently had lead to growth in user expectations. The streamlining of design and consistency should be used to offer a consistent experience across all products involved.

Lastly, project management also resulted in a major issue in the current market. A terrible management of multi-tasking several project can result in failure to deliver a useful product. A good planning and project management should be able to keep track of all project and resources. Meeting deadlines is crucial in managing projects and task allocation should be handled with care.

Best Practices of Software Development

One of the best practices is knowing and performing proper testing as it is essential. Each code should go through proper testing as nobody is perfect in programming and there will always be mistake found. Testing should be conducted to create strong protocols and establish quality of the code. A good unit testing should ensure mistakes are fixed before the product is pushed into production. There will never be a truly bug-free product but we can ensure the critical ones are remedied.

Establishing a proper repository to ensure mistakes can be fixed should be standard practice among programmers. Good version control system such as CVS, Subversion and Git makes it possible to experiment with coding and fix any issue found. A repository can help track the code and be able to find out where the bug came from. A repository can help synchronize works, track differences and merge work of others when developing projects.

Development methodologies should be priorities by ensuring a strategic plan is conducted. The various programming methodologies should be selected for different projects depending on the requirements. There can be no project without coordination and thus a methodology should be followed to ensure a myriad of people can work together.

Code maintenance should also be practiced when developing a program. Each software contains their own flaws and limitation. Developers may move on but the code they created will stay forever and thus proper documentation and maintenance should be conducted. Code maintenance ensure that the age of a software product can be elongated through good documented code with modular interfaces. The software life cycle of a product can be maintained by ensuring the future programmers can understand the code properly.

Best practices should be conducted by all software engineers to ensure a smooth development process. The basic development cycle should include good coding, proper testing and documentation. These great habits will better the quality of the software created and ensure a proper development process.

Relevant Questions Asked

Personally, I agree with Chris Anderson in asking Linus Torvalds whether he feels that he contributed a lot to the project. Chris is asking this as a way to probe into the character of Linus by assessing his answer. Linus does not respond proudly when he was complimented as a genius coder and his stubborn nature, rather he admits his faults in not being comfortable to being in the talk. He emphasises that he is not a visionary and is perfectly happy with others that have wild ideas. This is rather comforting as we often face strange requirements from stakeholders that simple do not make sense. He then mentions that he can only fixes potholes in front of him rather than looking at stars. Linus also stated that he is more of an Edison than Tesla in which he is the villain the whole debacle surrounding the two mythic figures. This is because he is not a nice person like Edison whom is an intellectual and not a visionary.

Opinion of the Talk

Linus Torvalds is an interesting figure in the way that he presents himself as someone who is socially inept yet provides meaningful solutions to the problem he faces. Throughout the talk, Linus is able to give a speech about how all the issues he faces is due to his hermit nature and personality. Like many other software engineer undergraduates that I know personally also submits to this stereotype of not being socially active. Torvalds also discuss his belief that code either works or it breaks. However, this seems rather contradicting as Linux kernel is the largest collaborative project thus far and contains more than 20 million lines of code and more than ten thousand contributors currently. Ultimately Linus has the final word on whether the code provided is effective or efficient. Linus seems to be myopic to other people's feeling is rather unsurprising as most software engineer undergraduates also exhibits signs of artificial intelligence. The nature of open-source allows different people to work together in harmony without the need to meet in person. Linus was listed as the most influential person by TIME magazine back in 2004. His contribution to business of software development cannot be understated. It is inspiring to see Linus be so humble when he mentions that he is merely an engineer and is only capable of fixing problems at hand.

Interesting Question

The one question that interest me is when Chris Anderson mentioned that the Linux project snowballed when people started to see the potential of it. Linus then stated that the contribution of ideas is the key in his development of the system. Like many other youths currently, he was only 21 with no understanding on how his project will grow. Linus inability to see the size and scope of his project rings true to every software development students around the globe. As I've encountered similar experience in underestimating and overestimating the requirements of projects before. Interestingly enough, Linus hatred of socialising push him to great heights and resulted in the development of Git and Linux.