

UNIVERSITI TUNKU ABDUL RAHMAN

[UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION](http://wble-sl.utar.edu.my/wble-sl/course/view.php?id=8889)

Name : Lim Yu Hao

Student ID : 1504833

Course : LKC FES (Software Engineering)

Lecturer : Dr. Tay Yong Haur

Name: Lim Yu Hao

Student ID: 1504833

Practical: 3

**Industry Talk: DevOps: Myth or Real**

I’ve attended the industry talk titled “DevOps: Myth or Real”. The talk is held at 19 July 2017 (Wednesday) in KB300, KB Block, Universiti Tunku Abdul Rahman Sungai Long Campus, Bandar Sungai Long, Selangor from 5:00 PM to 7:00 PM. The talks were given by Sian Lerk Lau. Sian Lerk Lau is a trained programmer, educationist at heart and engineering process enthusiast. He started his career with Universiti Tunku Abdul Rahman for 7 years before venturing into software development. He enjoys facilitating collaborative efforts between days, ops and testers in solving tough design, infrastructure and programming puzzles in our daily operations. He is currently the Engineering Manager for the CDN division in OnApp Ltd, a cloud management platform. This talk is to give us and overview of the latest development in software practices and identify the basic motivation behind DevOps practices and explores common toolchains used in the software development and delivery process.

The talk started by a simple survey. First, he asked us are we student. Of course, all of us, except Mr. Tay, raise up our hand. Then he asks are we computing students. Again, same amount of person raises up our hand. Next, he asks have we done programming before? Most of us raise up our hand as well. Then he asks us prefer agile or devops. Then he gave us some terms and asks us do we came across or understand those terms. Those terms are “unit test, behavioural test, functional test, deployment pipeline, continuous deployment, pair programming, configuration management”. I only heard of some, understand some, and some terms I never really heard of. Then, he showed us a picture of Silicon Valley. He explains what is Silicon Valley and what do we do here.

Next, he talks about what is agile. He starts to explain it by asking us to imagine one scenario. Imagine our girlfriend or boyfriend asked us to cook a meal for him/her. But we are not so good in cooking. What should we do. Then some of us start to give him suggestion on what to do. Some say refer to recipe, some say google the recipe. Then, we should start to cook. Then he says what to do after cooked one meal? We say we should taste it. He agrees, and continue giving his explanation. He says we do not follow blindly to the recipe, just because we want to do the thing right. Instead, we will continually obtain fast feedback through tasting the food repeatedly during the entire preparation process, and with only one thing in mind – to impress the person. This is what we call “Doing the right thing, not doing the thing right”. He says the most important goals of software development is to deliver happiness. Delivering value to customer is the one and only desirable business outcome. He asked us to imagine the scenario again. Of all the things we would have done, from looking through recipe and continuously cooking and tasting, it is certainly we will cook something favorable to the person’s taste, and continuous improvement during the entire preparation process would be vital. It’s all about coordination, and learning. This is what we call, “processes, best practices, coordination, and learning”. He says he thinks that software development process is like black magic sometimes. Often, we find software development methodologies, processes and best practices are like and unopened Pandora box. Even though the effectiveness of some specific agile practices is not yet fully established, the agile approach is consistent with the sound principles grounded in management and organization theories.

After that, he continues the next topic, what is devops. Devops separated to 3 main things, the flow, feedback, then learning and experimentation. He compared it to cooking which is processes, best practices, coordination and learning. He also compares to theories, which is dynamic capabilities, coordination, double-loop learning. He continues to elaborates each thing. First is the flow. We should make our work visible, limit work in progress (WIP) and reduce batch sizes. We should also reduce the number of handoffs, continually identify and elevate constraints and eliminate hardship and waste. He then shows us how he uses Kanban board to make sure all those things in control and have a good flow. Next, the feedback. We must be able to see problems as they occur. All people in the team should then swarm and solve the problems to build new knowledges. We also should keep pushing the quality closer to the source. We also must enable optimizing for downstream work centers. He shows us some example tools like Jenkins. Last is learning and experimentation. Working as a team, we should enable organizational learning and a safety culture. We also should institutionalize the improvement of daily work and transform local discoveries into global improvements.

After that, he showed us example of how he handles his works. First, he setup the working environment using Docker. Then, he git checkout and create a separate branch. He git status to check the status. Then, he starts to do some tests and expecting the test to fail. Then, he starts his coding work. After amending the code, he does the testing again. After done, he commits his code and notify his colleagues to review his codes. He also showed us the process of deployment. First, he merged files into another branch. Then, he shows us that the Jenkins will detect changes and help to merge and deploy. He says that using continuous software delivery, we can increase confidence in production readiness. He also showed us deployment pipeline, which including code commit, build, acceptance test, performance test, manual test and production.

He concludes the talk by ask us imagine the cooking scenario again. He asked us, does the practice of devops equate to a successful software engineering team? Not necessary. The most important thing is that we know clearly what are we doing, then we can do things effectively. And most importantly, we should always know that processes and best practices are not our ultimate goal or business outcome. The most important thing is to deliver value to our customers (internal and external) is the one and only desirable business outcome. He says the key takeaway of this talks is “Win the war, not battles”. There are many battles in a war, we should focus on winning the war but not just winning only one of the battles. He also says that true continuous software engineering is more than adopting continuous deployment. These are merely techniques, but the ultimate goal is to take a holistic view of a software production entity, whether this be a single software organization or an ecosystem where different organizations together deliver a final product.

After his conclusion, the question and answer session started. First, a person asked that what happen when there is error within automation process. Mr. Lau answered that through process of automation, we also will receive feedback, hence will know in time when an error occurs and can fix it in time. He also asked that what happened if there is a problem or error in the unit testing code itself. He answered that when they wrote a testing code, they will show it to the team and let then read and review the codes, hence it will greatly reduce the chance of problems in unit testing codes. Another person asked an interesting question, how is his daily life as a programmer. After laughing, he disclaimed that his is only his life, not all programmer has a life like this. His company has a culture that does not encourages overtime, so he usually works 8 hours a day. And also, in the office, although he is manager, they maintain a flat hierarchy. He wakes up at 9 in the morning, then leave house at 10 a.m., reach office at 10.30 a.m. Then he will start looking through the chat channel, so that he will know what happen during his down time. Then he will sort those issue together with his colleagues. On Tuesday and Friday, he will work remotely from home. When some of them face problems, they will use Google Hangouts to check the code. At about 12 in the noon he will have his lunch. After lunch, he will peer review some codes, and read technical documents. Sometimes, he also will fix some customer issues. He will leave office at 6.30 pm. or 7 p.m. depends on this workload. He also says that when he starts working few months, he always has to wake up at midnight when their clients face technical issues. But after he and his teams practicing the good practices he shared us just now, things got better and now they face much less issues. Another person asked him what is the hardest technical skill that we should focus, which he answered statistics, which he think is very useful when we working. One girl asked him should we focus on one or two programming language, or learn as much as we can. He says that it’s very depending on situations. It should depend on what we want to work next times and our working environment next times. The girl also asked him how if we get a task, or a project that we are not so sure how to do, or lack of knowledge. Mr. Lau answered that the most important thing is we should never give up. We should try hard to figure out how to go from basic to nice understanding of the certain field.

I think this talk is quite good as I’d learnt quite some knowledge from it. I get to learn about agile and devops, and the process of developments. If I only get to pick only one topic of the session that interests me, that will be the part where he showed us how he works and handle his tasks. From that sessions, I get to know how a real developer works.