


COMP3122 Final Project

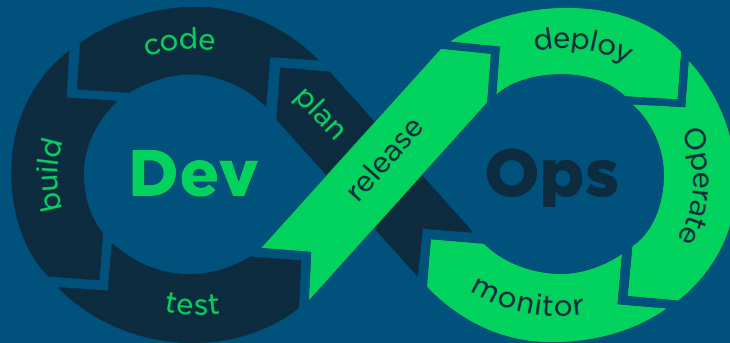


SHOAIB Muhammad [18079999D]



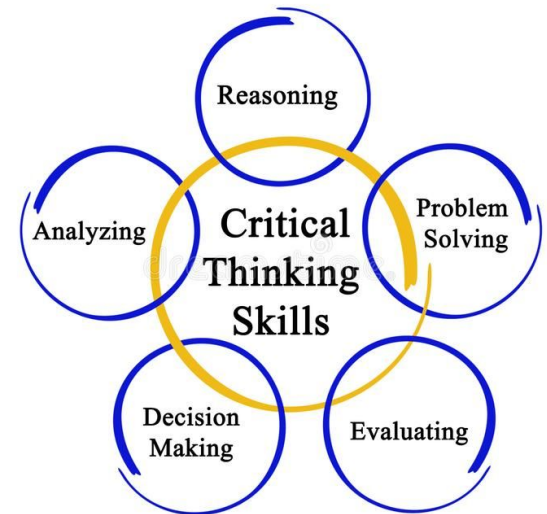
Introduction & Background

- DevOps Simulation for the development of an Airbnb-like system
- Player is a DevOps team member and the key-decision maker of the team.
- Focuses on key DevOps metrics
- Player has to handle different scenarios
- Judge which one is the best outcome



Learning Objectives

- Familiarise the player with DevOps metrics
- Handle dilemmas, trade-offs → Triggers critical thinking
- Judging which tradeoff or situation is the best
 - e.g. production incidences
 - system failure during testing
 - integration failure
 - scalability issues
- Focus on other DevOps metrics
 - Lead time
 - MTTR
 - Change Failure Rate



Simulation Rules

- **Lead time:** a score from 0-100, representing the estimated day of lead time. Initially, the lead time is 60, the player should try to lower this attribute, if the value is over or equal to 100, the player loses the game. The lower the lead time, the better.
- **Change Failure Rate:** a score from 0-100. This will increase as there is a problem with a deployment as a ratio of the total number of deployments. Initially, the change failure rate is 10. The lower the CFR, the better.
- **MTTR:** a score from 0-100. It means 'mean time to repair', it describes the average value of repair time when the product changes from failure state to working state. Initially, the score is 30. The lower the MTTR, the better.

Simulation Rules

- **Stakeholder Satisfaction:** a score from 0-100. It shows the satisfaction of the key stakeholders, the higher the better. If SS reaches 0, the player will lose the game.
- **Remaining Budget:** Initially, the remaining budget is 100, if it reaches 0, the game is over. The higher the remaining budget, the better.
- **Final score** = $[(100 - \text{Lead Time}) * 6] + [(100 - \text{Change Failure Rate}) * 3] + [(100 - \text{MTTR}) * 4] + (\text{Stakeholder Satisfaction} * 5) + (\text{Budget} * 5)$

Flow of the Simulation

- Player is provided with the scenario:
 - A major vacation rental online marketplace competitors has decided to shift its operations to an online booking system, and has outsourced the development of the system to your company.
 - Your company was in dire need of such a large-scale project. Now that your company has got it, you have to try your best to make the project a success.
 - You will employ DevOps, a set of practices that brings together software development and IT operations, to complete the project and make it a success for your company.

Design & Scoring of the Simulation

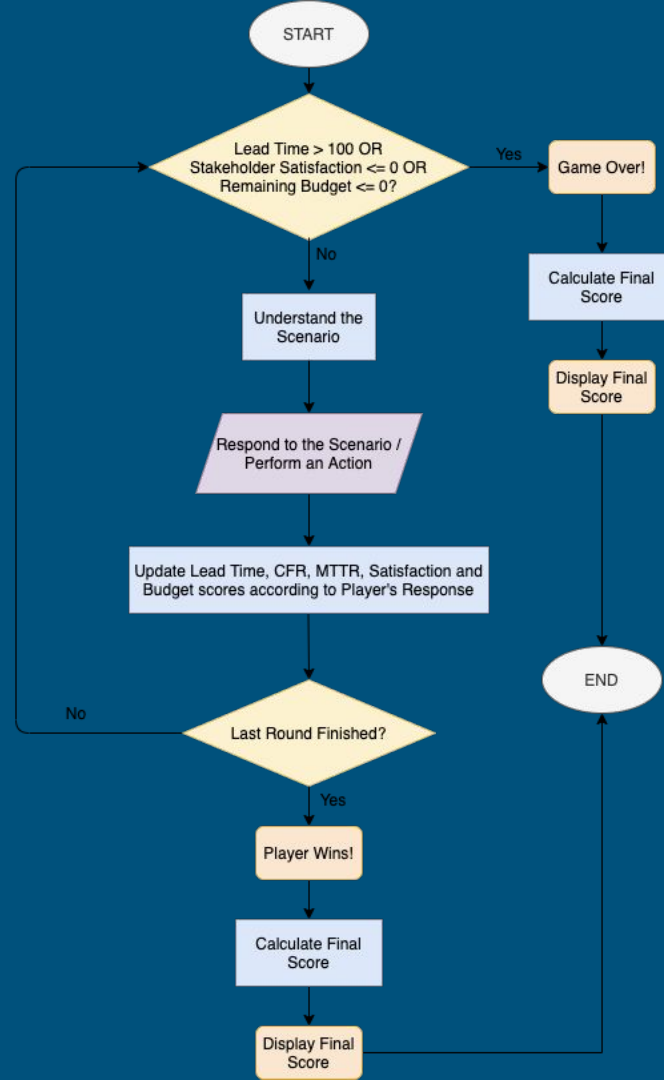
- Design: Non-linear, Influenced by user choices
- Score-based approach to calculating how well the player does
- Amongst all attributes only Lead Time, Stakeholder Satisfaction and Remaining Budget will determine whether the player can successfully complete the project and win.
- Final score shows how well the player understands the use of DevOps metrics.



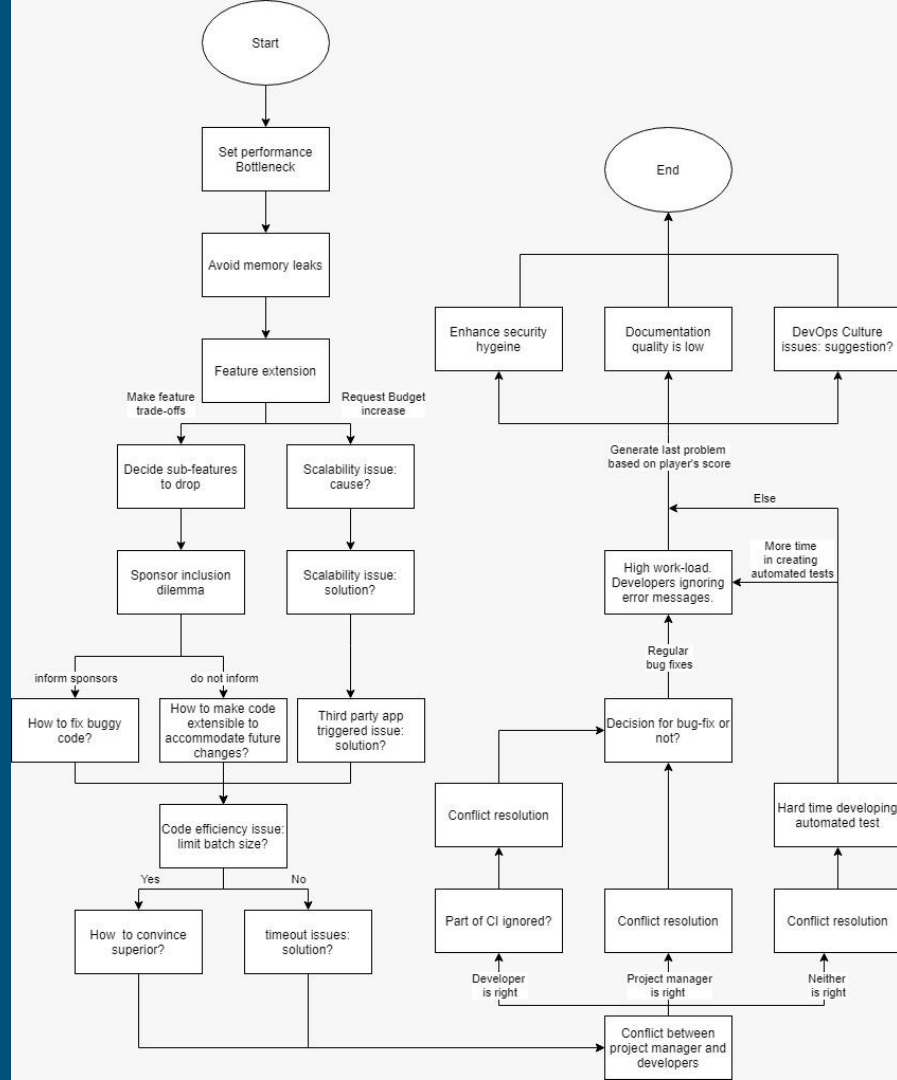
SIMULATION UI & DEMO

- Let's have a quick demo to show how the UI looks like and how a user can play the simulation/game.
 - Try to pay attention to the flow of the game during the demo.
-

Detailed Flow of the Simulation



Scenarios



Sample Qs That You Might Encounter While Playing The Simulation/Game

Sample Scenario

- The system that is to be developed will entertain a big audience, especially during the holiday season when a lot of people are trying to book accommodations at the same time and the server needs to deal with multiple requests.
- This involves the no. of visitors, traffic patterns, and features that users may access on your website. For instance, if your web has 10,000 concurrent users at its maximum, but 3,000 visit in the morning and that number gradually rises up to 10,000 in the afternoon—and these users mainly only search your homepage—you'll have to run a different test than if you've got 10,000 concurrent users on your site all day long, who continuously search several pages or access several features on your website.

Sample Scenario

You need to get a common view of application reliability and speed at every stage of the development, to identify and fix performance bottlenecks. In this situation, which one of the following actions should you take?

- (a) It is too early to consider or establish goals to carry out load tests. If needed, these can be carried out in the later stages of deployment.
- (b) 'Load Goals' should be established from the very start (through direct engagement with business colleagues and key stakeholders), and you should search tools to help you with the process.
- (c) Load tests should be carried out across the software development lifecycle, but 'Load Goals' do not need to be at such an early stage.

Sample Scenario

- The main features to be included in the system are:
 - (1) user should be able to post an ad
 - (2) user should be able to search for the accommodations
 - (3) user should be able to save favorite rental homes and places
 - (4) in-app messaging to the property owner
 - (5) ratings and reviews
 - (6) real-time updates and notifications.
- The project seemed to have an ample budget before the CEO requested to add a feature where you can connect with your friends and send them accommodation details in-app to plan a trip together.
- While this is an extension of feature (4) (which has already been developed by the programming team), the issue is that the extension still requires extra manpower and resources, so it probably is going to exceed the budget. What would you do in this situation?

Sample Scenario

- While this is an extension of feature (4) (which has already been developed by the programming team), the issue is that the extension still requires extra manpower and resources, so it probably is going to exceed the budget. What would you do in this situation?
 - (a) Tell the stakeholder that this feature cannot be implemented.
 - (b) You make a trade-off by not implementing some small components of a few features that may not be that important and implement this extra feature without asking for a raise in budget.
 - (c) Ask for a raise in budget to make up for the extra resources required.
 - (d) As you fear that this may lead to dissatisfaction of the stakeholders, you go a little over budget to accommodate their needs.
- Both (b) and (c) are viable options, but the user will have to handle different follow-up scenarios.

Project-related Q&A Based On Critical Thinking Aspects

What are the major decisions in the game/simulation?

Judge how much each option affects some metrics in a DevOps team:

- Lead time,
- CFR,
- MTTR,
- Stakeholder satisfaction, and
- Remaining budget.

Are there trade-offs or dilemmas between the different decisions/actions/options?

Multiple indicators introduce several types of combination

Examples:

- option A increases user satisfaction
- option B reduces the budget

How should the user make the decision and prioritization?

Focus on several indicators that can make him/her lose:

- Lead time
- Stakeholder satisfaction
- Remaining budget

Have you provided any context for making the decisions and prioritization?

- We design specific scenarios
- Present current value of indicators
- Present possible change of those indicators

References (Journals)

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.

Bou Ghantous, G., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. PACIS2017.

Davis, C., & Gaudin, O. (2015). Agile metrics in action : How to measure and improve team performance. Shelter Island, NY: Manning.

Dörnenburg, E. (2018). The path to DevOps. IEEE Software, 35(5), 71-75.

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). An exploratory study of devops extending the dimensions of devops with practices. ICSEA 2016, 104.

References

Mohammad, S. M. (2017). DevOps automation and Agile methodology. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.

Nader-Rezvani, N. (2019). An executive's guide to software quality in an agile organization (pp. 45-152). New York: APress / Springer Science+Business Media.

Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016, April). DevOps culture and its impact on cloud delivery and software development. In 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring) (pp. 1-6). IEEE.

References

Rossi, C., Shibley, E., Su, S., Beck, K., Savor, T., & Stumm, M. (2016). Continuous Deployment of Mobile Software at Facebook (Showcase). FSE 2016: Proceedings Of The 2016 24Th ACM SIGSOFT International Symposium On Foundations Of Software Engineering, 12-23. doi: <https://doi.org/10.1145/2950290.2994157>

Sánchez-Gordón, M., & Colomo-Palacios, R. (2018, October). Characterizing DevOps culture: a systematic literature review. In International Conference on Software Process Improvement and Capability Determination (pp. 3-15). Springer, Cham.

Schaefer, A., Reichenbach, M., & Fey, D. (2013). Continuous integration and automation for DevOps. In IAENG Transactions on Engineering Technologies (pp. 345-358). Springer, Dordrecht.

Snyder, B., & Curtis, B. (2018). Using Analytics to Guide Improvement during an Agile–DevOps Transformation. IEEE Software, 35(1), 78-83. doi: 10.1109/ms.2017.4541032

References (Websites)

<https://devops.com/devops-chat-continuous-delivery-at-airbnb/>

<https://techbeacon.com/app-dev-testing/how-airbnb-scaled-its-migration-continuous-delivery-spinnaker>

<https://medium.com/airbnb-engineering/production-secret-management-at-airbnb-ad230e1bc0f6>

https://www.infoq.cn/article/0OhXI_lxDkkJHsdM1HwS

<https://docs.microsoft.com/en-us/azure/devops/learn/>

References

<https://www.youtube.com/watch?v=esEFaY0FDKc>

<https://cloud.google.com/devops/>

<https://www.cnblogs.com/finer/p/12313928.html>

<https://www.tasktop.com/blog/5-best-metrics-youve-never-met/>

<http://blog.itpub.net/31547898/viewspace-2214005/>

<https://www.cnblogs.com/datapipeline2018/p/9994303.html>

<http://www.yunweipai.com/4469.html>

References

https://content.microfocus.com/reduce-defects-software-tb/business-impact-production-defects?lx=HhQqWV&utm_source=techbeacon&utm_medium=referral&utm_campaign=00172440

<https://victoriaecommerce.com/6-ways-reduce-app-deployment-time/>

<https://www.zend.com/blog/how-reduce-lead-time-continuous-delivery>

<https://stelligent.com/2020/04/14/reduce-cost-and-increase-efficiency-with-devops/>

<https://devops.com/devops-shift-left-avoid-failure/>

<https://stelligent.com/2018/12/21/measuring-devops-success-with-four-key-metrics/>

References

<https://inthecloud.withgoogle.com/state-of-devops-18/dl-cd.html>

<https://www.recipeforit.com/it-production-and-availability/improving-availability-where-to-start/>

<https://www.cablinginstall.com/home/article/16466101/strategies-for-increasing-system-availability>

<https://www.lce.com/Improving-Availability-Is-Much-More-Than-Maintenance-1167.html>

<https://www.infoq.com/news/2019/02/airbnb-deployment-pipelines/>



Q&A