

DevOps Simulation with a Rental Application Development

COMP3122 [Information Systems Development]
Project Report

SHOAIB Muhammad [18079999D]

Background

The game is designed to help the user learn various techniques and judge the best choices to make the project release in time. The player is asked to make different choices, as the key decision maker of the DevOps team, aiming to develop a vacation rental online marketplace platform. One of the similar applications existing in the market is Airbnb.

The user will be provided with different scenarios and dilemmas, and will have to choose between them. The user should, according to his/her understanding, choose the best possible option for every given scenario. Players' choices will affect some specific indicators, including lead time, change failure rate, MTTR, budget and stakeholder satisfaction. At the same time, players' choices may lead to different pathways/branches of the game (non-linear flow).

Learning Objectives

Through the game, the player can learn:

1. DevOps metrics. The definition of DevOps, the characteristic of DevOps and the metrics of the DevOps will be introduced.
2. How to handle different incidences such as when the player needs to make trade-offs between features and handle its consequences.
3. Software Deployment Pipeline. How features go through stages, how the DevOps metrics are improved with the pipeline.
4. Handling production incidence: Illustrate how DevOps team handles production defects and incidents.
5. Scalability, Availability and Reliability. Different strategies are adopted to meet the availability and scalability requirement of the different features and services.

Simulation Rules

We define the following attributes in the game:

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 0. 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.

Stakeholder Satisfaction: a score from 0-100. It shows the satisfaction of users, the higher the better. This attribute will increase if stakeholder's demands are met in time. If the stakeholder satisfaction reduces to 0, it means stakeholders lose their interests on the project, the game will be over. Initially it is 100.

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 = $[(100 - \text{Lead Time}) * 6] + [(100 - \text{Change Failure Rate}) * 3] + [(100 - \text{MTTR}) * 4] + (\text{Stakeholder Satisfaction} * 5) + (\text{Budget} * 5)$

Amongst all attributes, only Lead time, Stakeholder satisfaction and Remaining Budget will determine whether the player can successfully complete the project and win. And the Final score shows how well the player understands the use of DevOps metrics.

The Flow of Simulation

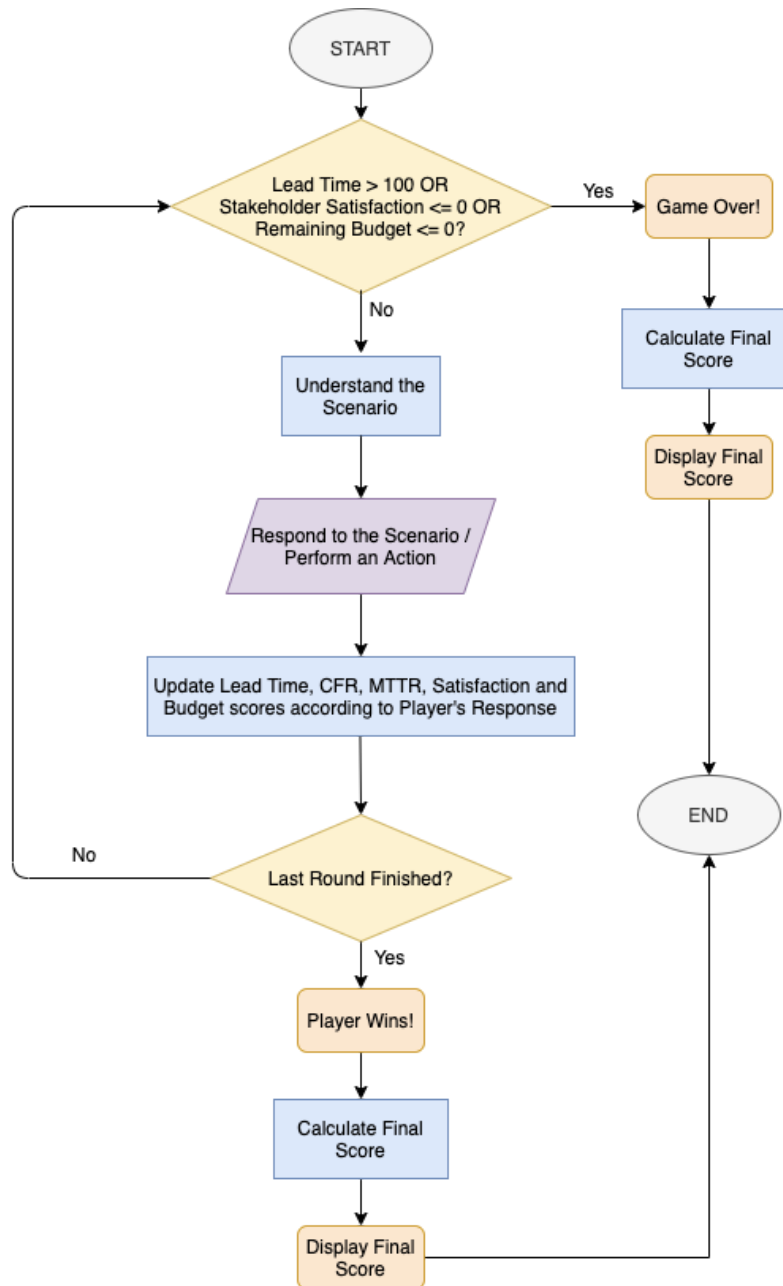


Figure 1. Simulation FlowChart

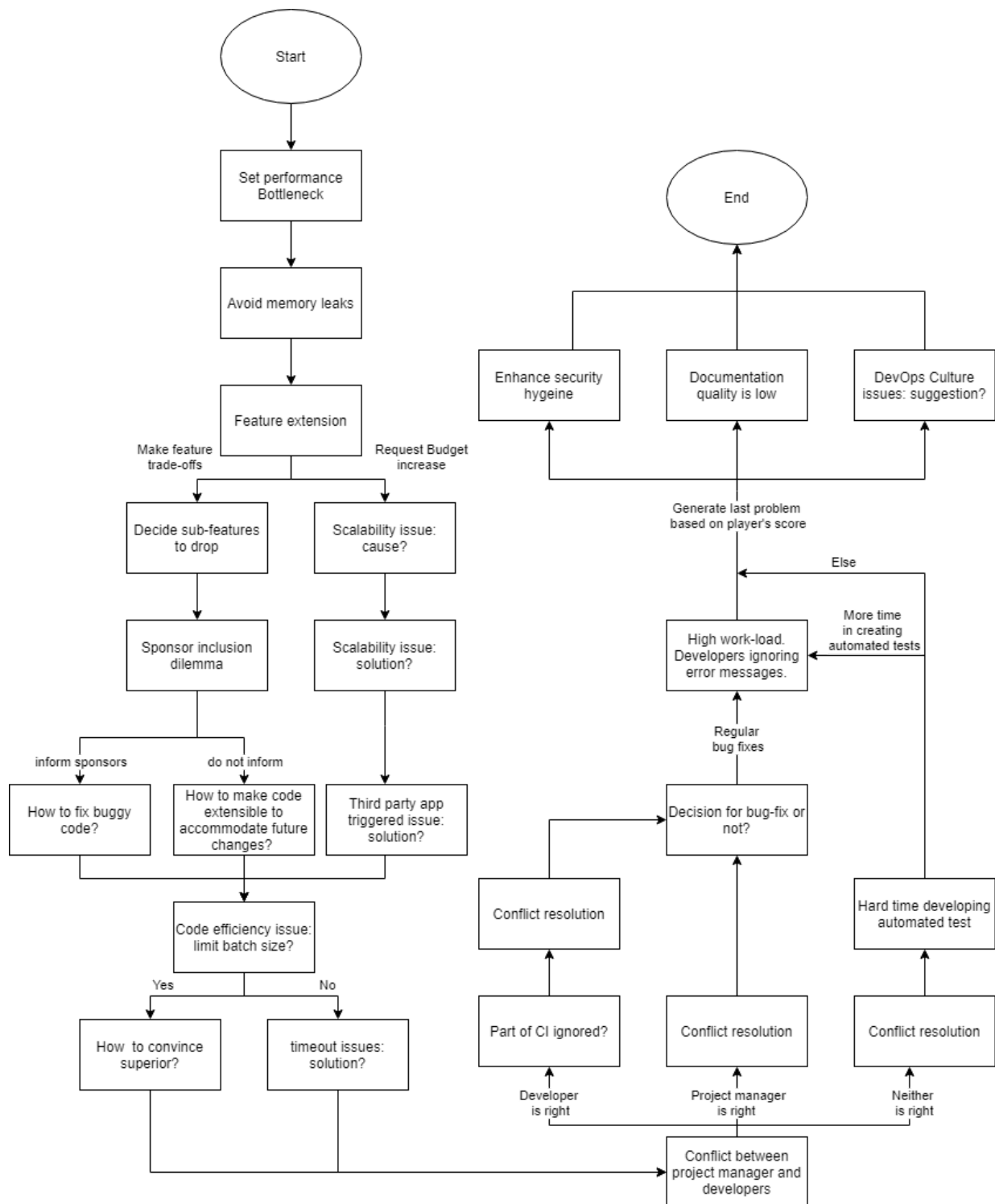
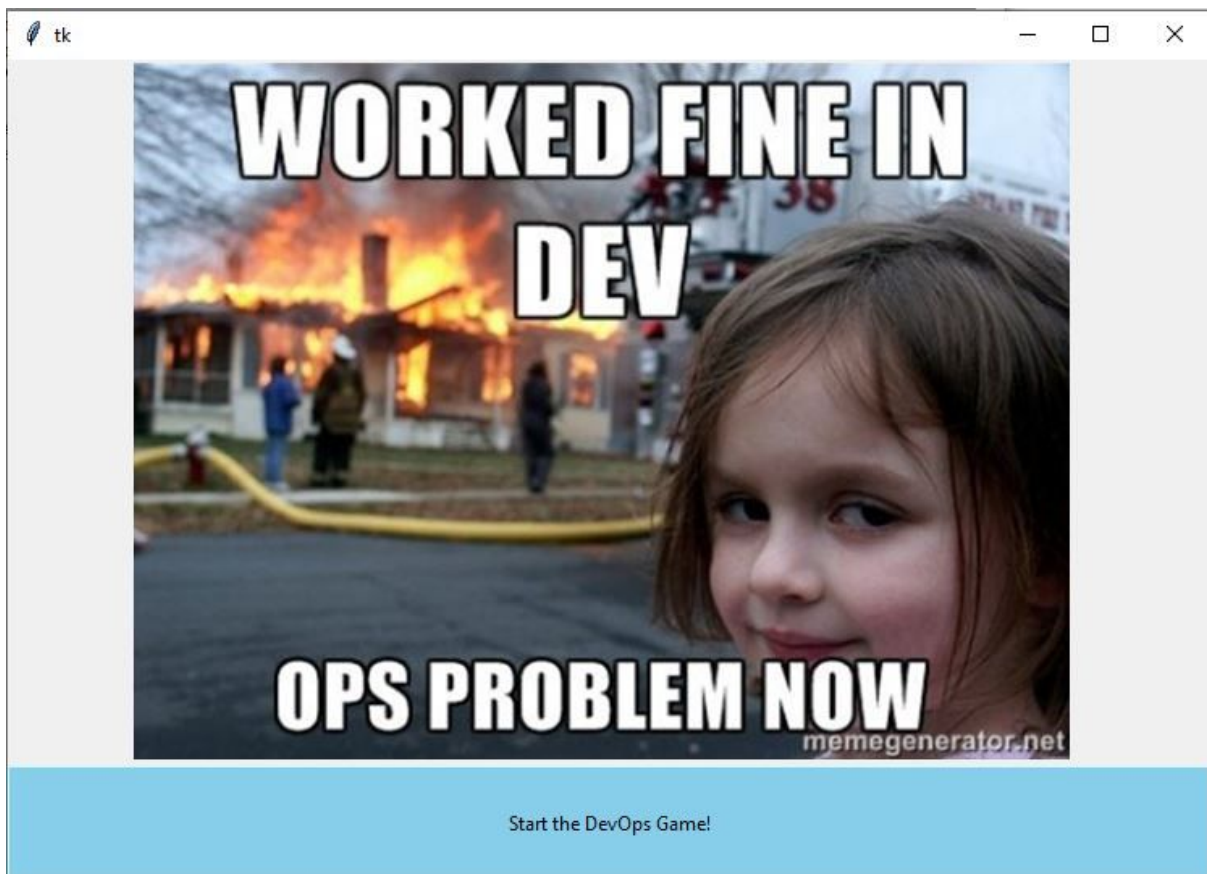
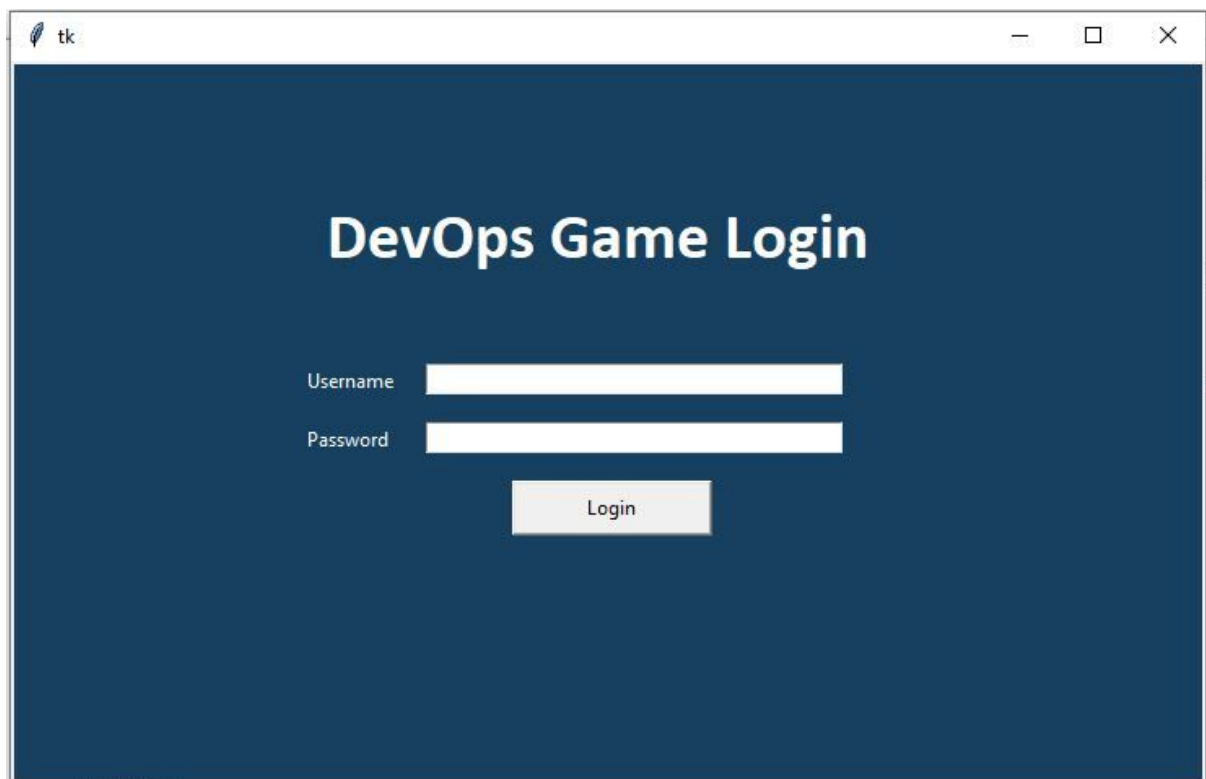


Figure 2. Order/Flow of Scenarios
(Each rectangle represents a scenario)

User Interface Design:



A Tkinter window titled 'tk' with standard window controls. The background is a solid dark blue. The text 'Create an account!' is centered at the top in a large, white, bold, sans-serif font. Below this, there are four labels and corresponding white input fields: 'Full Name', 'Username', 'Password', and 'Department'. At the bottom center, there is a white rectangular button with the text 'SignUp'. Below the button, the text 'Already have a Account?' is displayed in a smaller, white font.



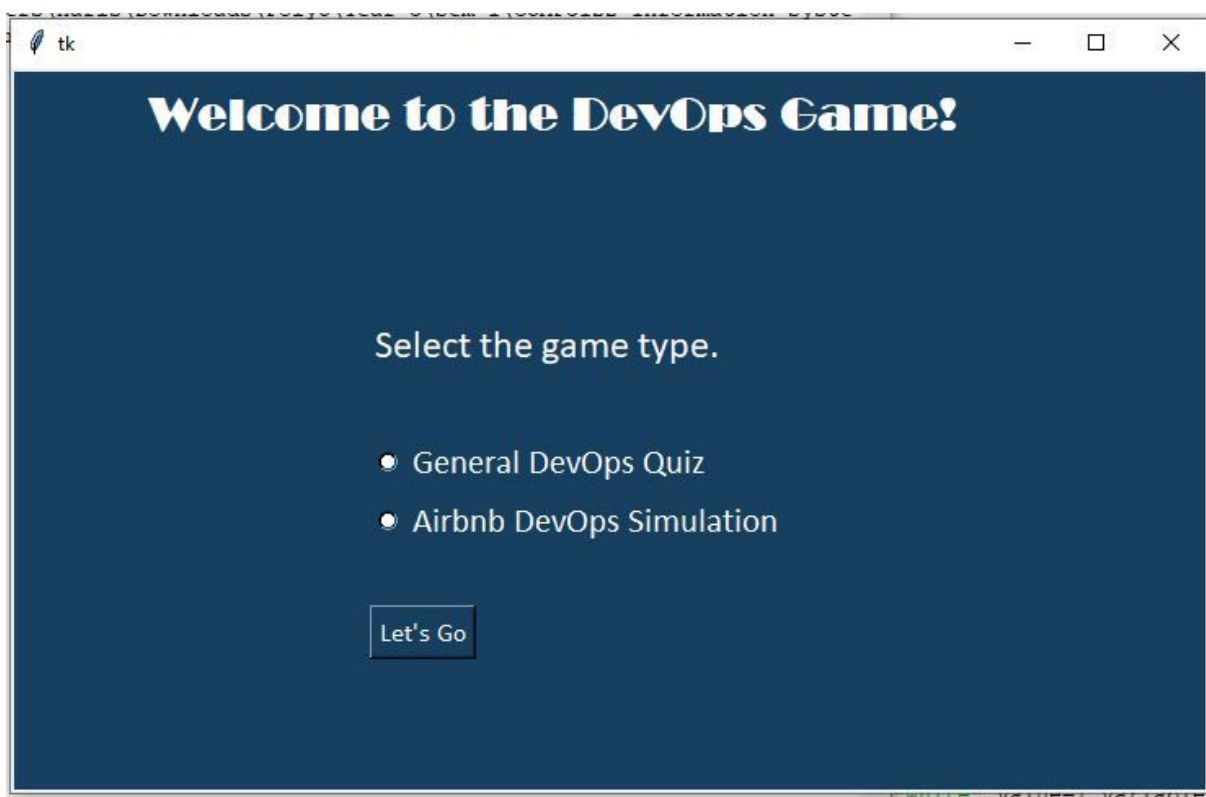
A screenshot of a Tkinter window titled 'tk' with a dark blue background. The window displays a login form for 'DevOps Game Login'. The form includes two white input fields for 'Username' and 'Password', and a 'Login' button below them.

DevOps Game Login

Username

Password

Login



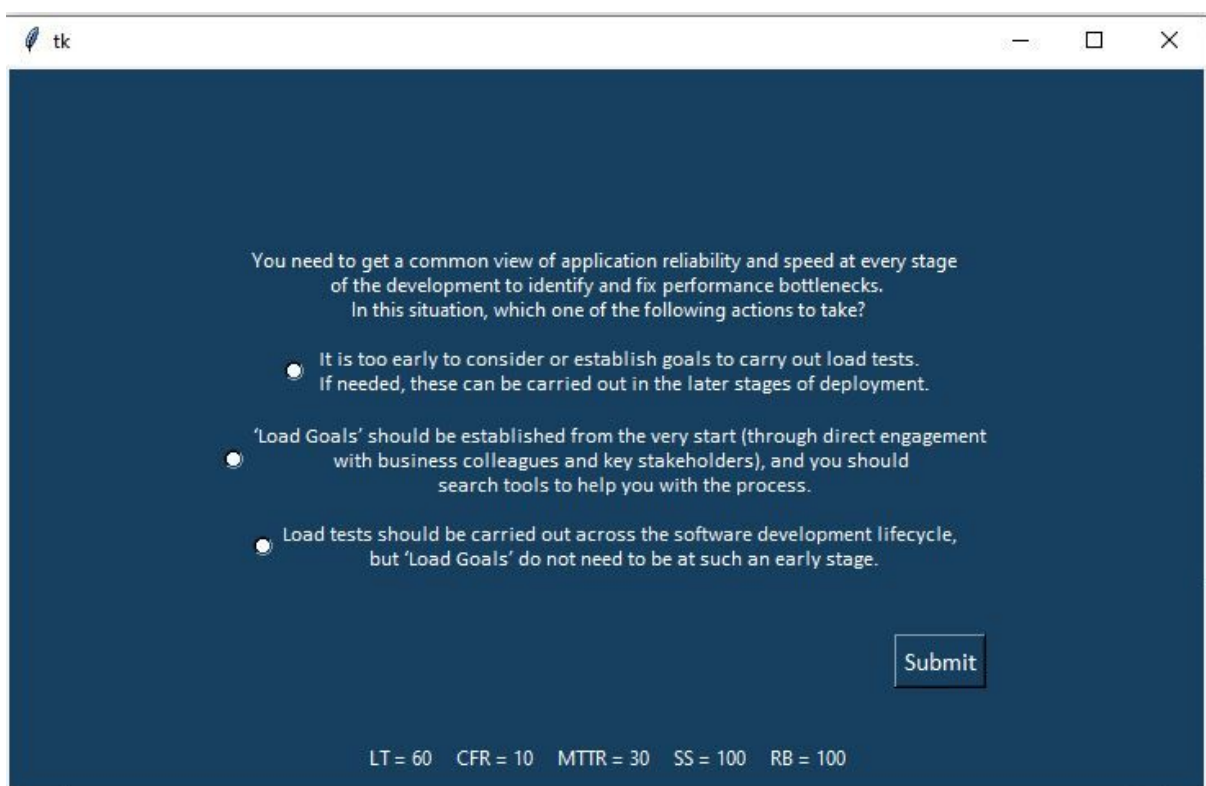
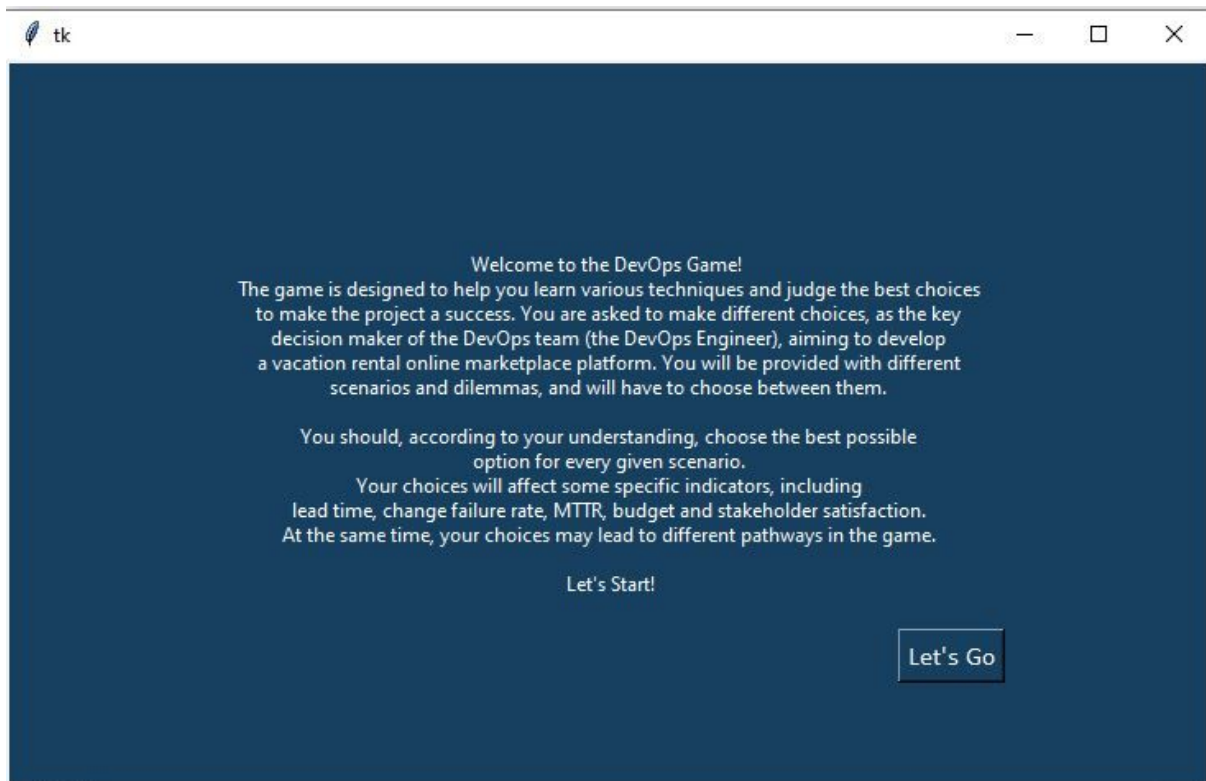
A screenshot of a Tkinter window titled 'tk' with a dark blue background. The window displays a welcome message and a selection screen for the game type. The text 'Welcome to the DevOps Game!' is at the top. Below it, the text 'Select the game type.' is followed by two radio button options: 'General DevOps Quiz' and 'Airbnb DevOps Simulation'. At the bottom, there is a 'Let's Go' button.

Welcome to the DevOps Game!

Select the game type.

- ☐ General DevOps Quiz
- ☐ Airbnb DevOps Simulation

Let's Go





Your Score: 258

Your Score Breakdown

Lead Time: 95

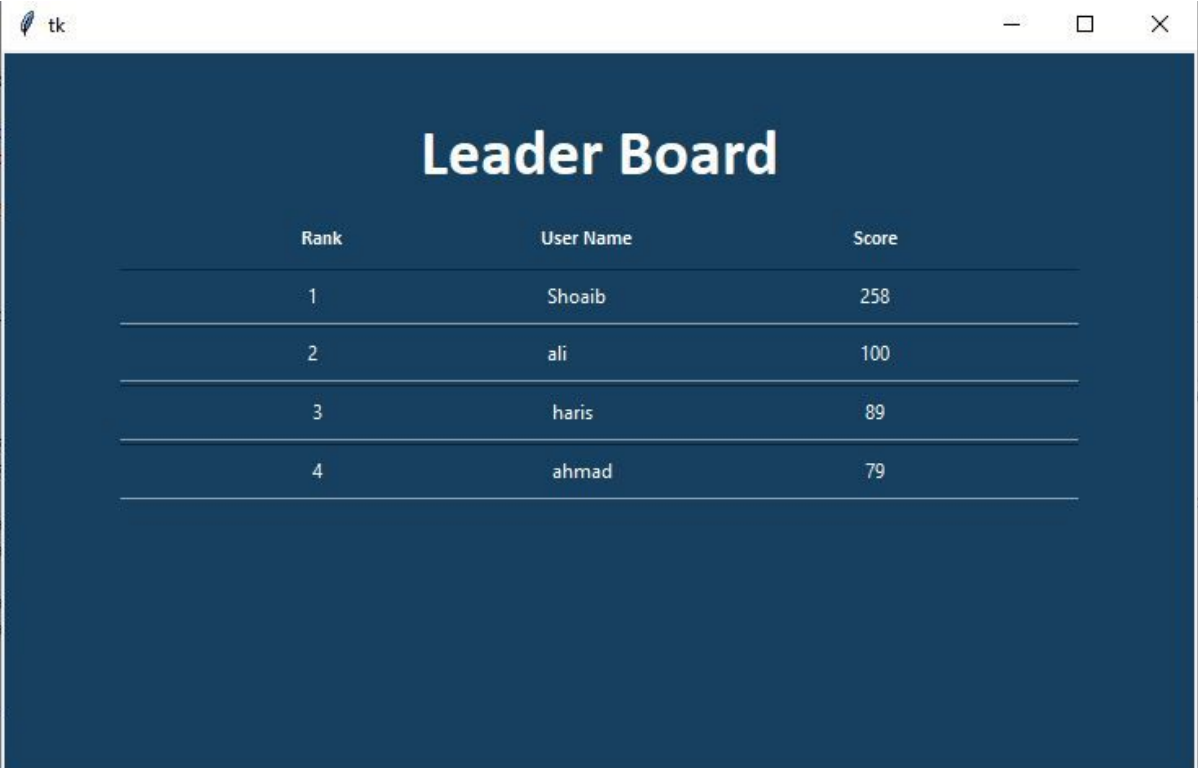
Change Faliure Rate: 89

Mean time to Repair: 95

Stakeholders' Satisfaction: 15

Remaining Budget: 20

Next



Leader Board

Rank	User Name	Score
1	Shoaib	258
2	ali	100
3	haris	89
4	ahmad	79

Simulation Design

Initially,

**Lead time = 60,
Change Failure Rate = 10,
MTTR = 30,
Stakeholder Satisfaction = 100,
Remaining Budget = 100.**

(These values will change when a user chooses an option.

The changes (increase or decrease is pre-set) for every option)

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.

THE CONSEQUENCES (ATTRIBUTE CHANGES) SHOWN WITH ALL OPTIONS ARE JUST FOR PROGRAMMER'S REFERENCE AND WILL NOT BE SHOWN TO USER WHILE HE IS PLAYING THE GAME.

This involves the number 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.

Please keep this in mind when handling the next two scenarios.

Ø: In such a situation, 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 of the following statements makes sense?

- (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.
Consequence: (Lead Time +5, MTTR +3, CFR +3)
- (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.
Consequence: (Lead Time -3, MTTR -2, Budget -4)
- (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.
Consequence: (Lead Time +3, MTTR +1, CFR +2, Budget -1)

Correct response: (b)

Load tests should be carried out across the software development lifecycle: APIs, components, microservices, and system-wide by using performance test metrics

translate into pass/fail results to automate the Continuous Delivery pipeline, and the 'Load Goals' need to be defined as early as possible!

Ø: What is the best way to ensure that the CPU stays low and you do not experience any memory leaks.

- a. To make sure you stand up to your load goals, you need a buffer; specifically, you need to make sure that your system can handle heavier loads; also, take your system to the limit and see when and how it fails.
(MTTR -2, Lead Time -5, Budget -1)
- b. A buffer is not needed as the 'Load Goals' that are set are pretty strict. Hence, just testing the system to its limits and seeing when and how it fails is enough.
(MTTR +4, CFR +6, Lead Time +3, Budget -2)

Correct response: (a)

Ø The project has started. Your team is using Git to collaborate on the project. You are noticing inconsistent repositories and you are not able to see the change history. What do you think is the issue here?

- (a) A developer is not integrating the contents of the feature branch to the master branch.
(MTTR +5, Lead Time +2, CFR +5)
- (b) A developer is using 'git rebase' instead of 'git merge'.
(MTTR -1, Lead Time -1)
- (c) A developer is using 'git merge' instead of 'git rebase'.
(MTTR +3, Lead Time +3, CFR +4)

Correct response: (b)

Prompt for (b): You are correct! You've correctly identified the issue at such an early stage. This is going to save you a lot of resources and time.

Prompt for (a) and (c): Oh! Wrong option. Since you couldn't identify this issue, it will cost you some time and resources.

Ø 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?

- (a) Tell the stakeholder that this feature cannot be implemented.
(Stakeholder Satisfaction -10)
- (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.
(Budget -10, Lead Time +3, CFR +5)
- (c) Ask for a raise in budget to make up for the extra resources required.
(Stakeholder Satisfaction -5, Lead Time -5)
- (d) As you fear that this may lead to dissatisfaction of the stakeholders, you go a little over budget to accommodate their needs
(Budget -30, Stakeholder Satisfaction +5)

Both (b) and (c) are viable options, but the user will have to handle different follow-up scenarios.

If the user choses (b), he gets the following three questions:

Ø While considering the trade-offs, you have arrived at a dilemma. You must drop one of the following sub-features (because they are not explicitly stated in the requirements document) to implement the new feature that the CEO has asked you to. Which one would you choose?

- (a) The user may search the accommodation by the accommodation id.
(Stakeholder Satisfaction -10, Budget -5)
- (b) The user may see the promotions going on in the app but will not get notifications.
(Stakeholder Satisfaction -5, Budget -5)

Correct response: (b)

While compromising on (b), the user still has the ability to view the promotions in the app, whereas if we chose to drop the feature described in (a), the user will not have the ability to search the accommodation by its id. Hence, it is better to drop the feature described in (b).

Ø You may have noticed that we did not involve the sponsors of the project in making the decision to exclude this feature. Was it a good decision to not include the decision of sponsors?

- (a) The sponsors should be kept informed of such decisions, so it was not a very good practice.
(Stakeholder Satisfaction +5, Budget -2, Lead Time +2)
- (b) Since the feature was not explicitly stated in the requirements document, it was okay to play smart and exclude it even if the stakeholders were expecting to have it in the final deliverables.
(Stakeholder Satisfaction -10, Budget -1, Lead Time +1)

- (c) The team working in a DevOps environment already has a lot on their plate, so whether stakeholders are satisfied or not shouldn't bother them.

(Stakeholder Satisfaction -5, Lead Time +5)

Correct response: (a) and (b) are correct, follow-up questions will depend upon what user chooses.

If the user chooses (a), he gets the following question:

Ø As per your suggestion, the sponsors were informed of this decision. This led to serious dissatisfaction. Now, they are demanding that you implement both features. This has sparked anger amongst the developers. They have started to hastily code, are ignoring the error messages they get while coding and are deploying buggy code. What should you do in such a scenario?

- (a) Communicate the significance of not having bugs in the code in the first place and encourage the team members to perform their best. There is no need to have a developer dedicated to fixing bugs before deployment.
(MTTR +5, Lead Time +5, CFR +10, Stakeholder Satisfaction - 15)
- (b) Keep a team member on the call for every single day that the development process is being performed. Ask him/her to react to error notifications in the code and try to fix these bugs.
(MTTR -2, Lead Time, -2, CFR -1, Budget -3)
- (c) Even if it lowers work satisfaction, ask all the members to be responsible for the bugs in their own codes and fix them as soon as they encounter them.
(MTTR +3, Lead Time +4, CFR +7, Stakeholder Satisfaction - 10)

Correct response: (b)

If the user chooses (b), he gets the following question:

Ø Because you did not involve the key sponsors in this critical decision, your superior fears that they might be dissatisfied when they see that this feature is not implemented. He asks you to design the code such that this sub-feature, if necessary, can be implemented by extending the main feature. What approach would make the feature more extensible?

- (a) Visualize and understand critical details about your network and deployed resources. Integrate nodes with your own organizational-controlled cloud resources.
(Lead Time -2, MTTR -3, Budget -2)
- (b) Node integration with the organizational-controlled cloud resources is not essential to making the feature more extensible.
(Lead Time +2, MTTR +4, Budget -5)

Correct response: (a)

Prompt with Reason/Explanation:

Add extensibility and heightened control to your blockchain network by optionally integrating a node with native AWS and Azure services, all managed and configured within your own organizationally-controlled cloud suite. Services include key management, log streaming, backups and private data routing.

If the user chooses (c), he gets the following three questions:

Ø The stakeholder agrees to extend the budget. After the programming team deploys the extension of feature (4), you find that there are scalability issues as after the extension, the load tests keep failing. What caused this issue of scalability in the first place?

- (a) Lack of observability and forecasting, or
(Lead Time -2, MTTR -1)
- (b) Lack of a clear understanding of feature requirements?
(Lead Time +2, MTTR +3)

Correct response: (a)

Ø You suspect that the issue exists within software scaling. Your team has suggested these approaches to improve the scalability of the code:

1. using RUM (real-user monitoring)
2. user-facing metrics such as HTTP error rates
3. understanding system behavior during load tests

Which of these processes are relevant and should be carried to enhance the software scalability?

- (a) 1 and 2 only;
(Lead Time -1, MTTR -1, Budget -2)
- (b) 1 and 3 only;
(Lead Time -1, MTTR -2, Budget -2)
- (c) 2 and 3 only;
(Lead Time -2, MTTR -1, Budget -2)
- (d) 1, 2 and 3.
(Lead Time -2, MTTR -2, Budget -3)

Correct response: (d)

Ø After employing these techniques, it turned out that the scaling issues were encountered because a 3rd party software was involved in the development of the feature. What should be the next step in this situation?

- (a) Perform load tests again with larger load limits;
(Lead Time +2, CFR +3, MTTR +3, Budget -2)
- (b) Explore hardware scalability issues rather than software scalability issues.
(Lead Time -1, CFR -2, MTTR -2, Budget -3)

Correct response: (b)

After upgrading some hardware components, the issue was resolved.

Ø It has come to your attention that during the testing phase, several code efficiency issues have emerged. A staff member suggests limiting the batch size of the continuous DevOps deliveries. What would you do?

- (a) You will refute his suggestion as you think it is not the right approach to deal with the issue.
(Lead Time +3, MTTR +2, CFR +1, Budget -3, Stakeholder Satisfaction -15)
- (b) You think it is a good idea and you would suggest implementing it as soon as possible to make the process smoother.
(Lead Time -2, MTTR -1, CFR -1, Budget -1, Stakeholder Satisfaction +5)
- (c) You nod in agreement not to be rude, but you think that limiting the batch size of the continuous DevOps deliveries is not the way to go.
(Lead Time +3, MTTR +2, CFR +1, Budget -3, Stakeholder Satisfaction -10)

Correct response: (b)

If the user chooses (b):

Ø Seems like you are in favor of limiting the batch size. Smart move! Your superior asks you what would be the benefit of limiting the batch size of the continuous DevOps deliveries?

- (a) Potentially required rollbacks from your production systems will be less cumbersome.
(Lead Time -1, MTTR -1, CFR -1, Budget -2)
- (b) By continuously delivering in production, your team will have the constant pride of contributing your organizational mission.
(Lead Time -1, CFR -1, Budget -2)
- (c) You will be quicker to identify root causes of issues and resolve them.
(Lead Time -1, MTTR -1, Budget -2)
- (d) All above choices.
(Lead Time -3, MTTR -2, CFR -2, Budget -4)

Correct response: (d)

If the user chooses (a) or (c)

Ø Uh Oh! You should've chosen to limit the batch size! Since you didn't, a major has appeared at a later stage in testing. When a user applies multiple filters to search for an accommodation, the code times out. Now that you're very close to developing the application, you've got to make a careful decision on how to resolve this bug. What would you do?

- (a) Dedicate a senior developer to work on fixing this bug.
(Lead Time -3, MTTR -2, CFR -2, Budget -3)
- (b) Take all developers on board and ask for their suggestion.
(Lead Time -2, MTTR -1, CFR -1, Stakeholder Satisfaction +5)
- (c) Ignore this bug for now and put it into the backlog.
(Lead Time +5, MTTR +4, CFR +4, Budget -4)

Correct response(s): (a) and (b)

Prompt if (a) and (b): Hurrah! You made a wise decision. The bug has been fixed.

Prompt if (c): No! You should've gone with anything except this option. MTTR has increased. Lead time has increased. Stakeholder satisfaction has decreased.

Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests. Among several key benefits, continuous testing helps with quick error detection and resolution.

While implementing Continuous integration, the team has run into conflict due to difference in roles and priorities of individuals making CI counterproductive.

The project manager, while doing his job is focused on meeting the deadline for a new feature that was planned to be launched soon putting pressure on software developers. Developers, however, think slowing down to fix minor bugs during the development process is slowing them down.

Ø. This scenario displays the lack of understanding from the developers and project manager. Which member of the team is right in this scenario?

- a) Software developers. It is better to deploy the feature without bug fixes and resolve them later.
(Lead Time +1, MTTR +2, CFR 0, Budget 0, SS -5)
- b) Project Manager. The development team needs to work harder to maintain the quality of the program.
(Lead Time +1, MTTR -1, CFR -1, Budget 0, SS-5)
- c) neither. As the project manager needs to meet the deadline but the development process may also be affected due to the continuous bug fixing.
(Lead Time 0, MTTR -1, CFR -1, Budget 0, SS 0)
(c is correct. All lead to different chains of questions.)

Beginning of case C and B but after 1 Q in a

Ø. Which of the following is the best way to resolve this conflict and keep the development process running?

- a) Call a team meeting and make them understand the benefits of Continuous integration.
(Lead Time +2, MTTR 0, CFR 0, Budget 0, SS +5)
- b) Highlight some incentive developers can gain from putting in more effort.
(Lead Time -3, MTTR -4, CFR -4, Budget -5, SS +10)
- c) Both a and b.
(Lead Time -5, MTTR -4, CFR -4, Budget -5, SS +10)
(C is the correct answer.)

In case of option a

Ø. The software developers are partially correct in their concern. What part of continuous integration are they missing?

a) Features cannot be deployed with minor bugs as it may cause merge problems.

(Lead Time +2, MTTR +2, CFR +2, Budget 0, SS 0)

b) Deployment is possible but minor bugs may lead to severe system crashes making it difficult to debug.

(Lead Time -5, MTTR -5, CFR -5, Budget 0, SS 0)

Prompt in a

Deployment cannot be restricted by minor bugs. It may slow down future debugging processes because of accumulating issues leading to severe issues.

Followed by Q mentioned above.

In case of b

Q mentioned above

In case of option b

Q mentioned above

Project managers may be partially correct about the need to keep up with minor bug fixes. Adapting to Continuous integration can be difficult for many organizations due to the organizational culture and structure. And it needs to be a gradual process, if imposed onto the employees may restrict the organizations potential. DevOps teams need to understand the benefits of CI to not just the organization but themselves.

Ø. You need to help the team decide between delaying bug fixes and feature deployment with minor bugs. What would you choose considering that each option may slow down or speed up the development process?

a) Insist on regular bug fixes by developers.

(Lead Time +4, MTTR -5, CFR -5, Budget 0, SS -10)

b) Add the bugs to the backlog for future resolution and deploy the feature.

(Lead Time +5, MTTR +5, CFR +10, Budget 0, SS +2)

c) pass the issue to another DevOps engineer.

(Lead Time +2, MTTR +2, CFR +5, Budget -5, SS -10)

Lead option a to the last question in case of c

In case of option c.

Q mentioned above

Adapting to Continuous integration can be difficult for many organizations due to the organizational culture and structure. And it needs to be a gradual process, if imposed onto the employees may restrict the organizations potential. DevOps teams need to understand the benefits of CI to not just the organization but themselves.

A good CI setup speeds up your workflow and encourages the team to push every change without being afraid of breaking anything. There are more benefits to it than just working with a better software release process. Continuous Integration brings great business benefits as well.

Ø. Now that you understood the conflict and understood the benefits of CI. Despite the benefits of CI, adapting to CI is not a piece of cake. Your development team is having a hard time keeping up with writing automated test cases. You have to make a critical decision and each option has its own downsides. What would you choose?

- a) Step back to traditional manual testing and delay automated testing.
(Lead Time +5, MTTR +5, CFR +10, Budget 0, SS +5)
- b) Provide some incentive to the developer's team to work harder on automated test cases.
(Lead Time +2, MTTR -5, CFR -5, Budget -7, SS +5)
- c) Hire third party developers/freelancers to create automated test cases.
(Lead Time +2, MTTR -5, CFR 0, Budget -10, SS -5)

In case of option a

Your MTTR just increased. Automated tests help in the long run by reducing the mean time to repair during continuous integration. It makes it easier to find bugs without manually testing all parts of code every time a batch is deployed.

In case of option b

As you chose to provide monetary incentive to the developers, your remaining budget has decreased. Although, Automated tests help in the long run by reducing the mean time to repair during continuous integration. It makes it easier to find bugs without manually testing all parts of code every time a batch is deployed.

Ø. Increased workload and divided attention increase the chance of errors. Testers raised an issue with exceptions that were ignored in the development process. Developers have been ignoring error messages due to the increased number of notifications from multiple submissions. As a devOps engineer, which of the following decisions would you like to make?

- a) Only allow necessary notifications to the concerned members of the development team and avoid unnecessary pop ups.
(Lead Time -2, MTTR -5, CFR -5, Budget 0, SS +5)
- b) Assign a person in charge for reacting to CI notifications and the whole team may take turns.
(Lead Time -2, MTTR -5, CFR -5, Budget +5, SS +5)
- c) Request developers to not mute notifications and keep an eye on daily notifications otherwise they may face serious consequences.
(Lead Time -2, MTTR -2, CFR -2, Budget 0, SS -10)
- d) Both a and b.
(Lead Time -4, MTTR -5, CFR -5, Budget +5, SS +5)

You have been entrusted with a lot of important responsibilities throughout the process. Now, since we're almost about to finish developing the platform, you have to perform one last task.

The program will generate a random number from 1 to 3. The following scenario would be based upon the random number generated.

If the number generated is 1, user encounters the following scenario:

Ø Due to inadequate automation of configuration checks and vulnerability scanning, the DevOps output has been shown to have a lack of security hygiene while handling privileged inputs (API access tokens and passwords). What should we do in this scenario?

(a) Cross-functional collaboration is the key to effectively integrating security into the entire DevOps lifecycle. Hence, it is not too late to implement DevSecOps instead of DevOps.

(Lead Time +5, MTTR -2, CFR +3, Budget -3, SS -10)

(b) Carry out penetration testing and ask security teams to identify vulnerabilities. Then, they should work closely with the developers to fix these issues.

(Lead Time +3, MTTR -4, CFR -1, Budget -3, SS +5)

(c) Limit privilege access rights to reduce the avenues for attack, even if it means to restrict some of the essential access required by users to perform certain actions or use certain features on the website.

(Lead Time +3, MTTR -2, CFR -2, Budget +2, SS +5)

Correct response: (b)

If the number generated is 2, user encounters the following scenario:

Ø Documentation is created in a waterfall fashion, rarely updated, highly prone to human error and lacking consistency. The DevOps team, while trying to code, test and deploy features was just too busy to finish documentation that inevitably was abandoned altogether. How do you plan to deal with this situation?

(a) Dedicate a developer to review the documentation. Ask the developer to review and formulate the documentation in a detailed manner even if there is a need to hire extra help.

(Lead Time +2, MTTR 0, CFR 0, Budget -3, SS +5)

(b) Ignore the documentation as it is not that important. The timely delivery of the system is necessary and should be focused upon. Documentation quality is not the main deliverable so is not that critical.

(Lead Time +2, MTTR +2, CFR +4, Budget 0, SS 0)

(c) Dedicate a developer to review the documentation. If it lacks in quality, ask him/her to replace the detailed documentation with condensed demos that are critical for the understanding of the functionalities of the system.

(Lead Time +3, MTTR 0, CFR 0, Budget -5, SS +5)

Correct response: (c)

If the number generated is 3, user encounters the following scenario:

Ø During the project, we were faced with several technical issues and bugs. Another thing that was very prominent was the lack of collaboration between teams, which led to a rather impeding and thwarting Agile

environment. What would you suggest, for future teams, to reduce the impact of this issue?

- (a) Design audit-based control frameworks to improve quality, assurance, security, compliance, and risk mitigation via checklists and audits of activity.
(Lead Time +3, MTTR -1, CFR +2, Budget -2, SS -5)
- (b) Implement organizational changes at smaller units for better understanding of agile development, helping them build trust with more autonomous control frameworks, to make collaboration easier.
(Lead Time +1, MTTR -3, CFR -2, Budget 0, SS +5)

If the user successfully reaches this point without running low in any of the scoring attributes, he/she is successful in launching the project and the total score is displayed.

Further Development

- Add more metrics that determine the win/lose.
- More reasonable numerical design.
- Larger dataset to make better estimates of how every choice should affect the scoring attributes.
- More scenarios and more questions related to real world DevOps development.

Critical Thinking

1. What are the major decisions in the game/simulation?

Judge how much each option affects some metrics(lead time, CFR, MTTR, stakeholder satisfaction and remaining budget) in a DevOps team.

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

Yes, multiple indicators introduce several types of combination, these combinations have different advantages and disadvantages, so there are dilemmas. For example, option A increases user satisfaction on a small scale, while option B reduces the budget. Any wrong move may lead to failure, and a wise move might help a player to get a higher score.

3. How should the user make the decision and prioritization?

Though we introduce the final score in the game, it does not directly influence the game round. The player needs to focus on several indicators that can make him/her lose, in specific, lead time, stakeholder satisfaction and remaining budget. Players should try to avoid these indicators reaching their limits.

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

Yes, for each question, we design a specific scenario, the current value and possible change of those indicators are also presented to players. With the information, we believe that a player would think calmly and try to make the best choice for himself/herself.

5. What are the good practices and challenges related to DevOps/Modern Software Development that you want to show in your game/simulation?

These practices and challenges are shown in the report. For example, we want to add choices to help players to understand metrics of development and how to make proper choices. The main challenge is to develop the project within some limits.

6. Describe the “research questions”, “investigation questions” or “hypothesis” related to DevOps/Modern Software Development that you have proposed or developed related to the project.

1. What are DevOps metrics?
2. How these metrics influence development?
3. Software Deployment Pipeline
4. How to handle production incidence?
5. What is scalability, availability and reliability, and how they influence development?
6. Hypothesis: You always have to make hard decisions in development.

7. Discuss, with specific examples, the approach(es) you have adopted to investigate your proposed questions, collect data to confirm your understanding or test the hypothesis.

For example, I am in charge of collecting questions, I need to know what is wrong with the current DevOps development process through course slides, website pages and journals. Then I have to check the details of DevOps to illustrate how this method solves problems in different scenarios.

8. What are the major results and findings from your study/investigation?

To develop a project with specific limits and metrics is a complicated issue. For example, you want to reduce lead time to keep the project on schedule, one of the possible consequences is budget overrun. The trade-off is hard to decide. When we consider the real world scenario, lead time is not the only aspect that we need to focus on. Budget and stakeholder satisfaction are also two important criteria. Therefore when we design the choices, we try to show the influence on at least two aspects.

9. How would you integrate the findings and communicate your results/understanding in the final project deliverables (the simulation/game)?

We first separated the project into several aspects, we worked out a draft structure of the whole game, then we designed the sample questions, based on these questions, we developed more questions to check different aspects of players knowledge about DevOps. Having the structure and questions, we reflect what we missed and try to fix them.

The draft structure and sample questions help us improve our design continuously. Eventually, they correspond to each other and create a whole project.

10. Related to the project requirements and deliverables, what are the major challenges in the project? How would you overcome the challenge?

I think the major challenges are the game design, including game flow, numerical design and scenario design. We need to work out a proper structure, put questions into a scenario and set the change of values reasonable. We first found questions in the real world, then thought about scenarios that can fit our questions. After that, we set the value change and test. Eventually, we got a reasonable game flow.

11. Comment on your overall learning experience of the project. Do you think the project can help you to develop the ability to "learn to learn"? Explain your answer with relevant examples.

We believe that this project helps us to attain the ability to "learn to learn", we design a simulation to simulate real world scenarios. We need to research different choices. In traditional teaching, we learn correct choices and incorrect choices. However, real world is more complex, sometimes you only get two bad choices, you have to choose an option with less damage.

References

Academic 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.

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.

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/00hXl_lxDkkJHsdM1HwS

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

<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>

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/>

<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/>