Ethics of Canary Testing: Users as Guinea pigs

Fredrik Svanholm - svanhol@kth.se

May 30, 2022

1 Introduction

Canary testing or canary release, is a method of software deployment where a new version or feature is rolled out gradually to users and evaluated over time to mitigate negative effects and/or bugs [8]. Canary testing targets specific user groups and can thus like other testing methods that can compare different versions metrics, be very powerful in optimizing and improving software [5]. When developers are presented with the ability to target things toward specific users, certain ethical concerns present themselves and developers may run the risk of engaging in ethically questionable practices. I suggest that by keeping testing practices transparent with users to keep the team accountable for their testing practices along with trying to make sure the interests of the users are kept in mind and prioritized and giving users the autonomy of opting in or out of these canary tests, developers may be able to mitigate the risks of ethically questionable practices while still utilizing the flexibility and power of canary testing.

2 What is Canary Testing

Canary testing is a method used to release, test and evaluate new versions of software or a service in the field of Software Engineering. The name canary testing historically comes from coal miners bringing cages with canary birds into the coal mines. The birds acted as early warning signs for the miners and if the canary birds died it was a sign that lethal gasses were too high and that it was time to evacuate and leave the mines. [8] Similarly in software testing, canary tests are used to prevent large scale catastrophes and complete system failures when rolling out new updates.

The idea behind canary testing is similar to rolling deployment where specific servers in the live production environment are given a new update and the results of the new update are evaluated [9]. If the results seem to indicate that it works as intended, then the update is gradually rolled out to more nodes. But with canary testing, instead of deploying the update to specific servers, the new updates are instead rolled out to a specific subset of users of the application. User metrics of this user subset are then analyzed and if these metrics indicate that the update is working, or if it points towards an improvement, then the new update continues to roll to a larger subset of users with the goal of it eventually reaching all users [9]. For a visual representation of the gradual deployment see Figure 1.

An example of how developers could use canary testing is by using something called feature flags. A feature flag is simply a Boolean value that developers can conditionally enable based on some condition [6, 7]. With Feature flags developers could set features to be enabled by user age or other personally identifiable information that may make those specific users desirable to test the new features on. These flags along with the new code would be put into the production environment and live build of the app and user metrics would indicate which flags a user has enabled to differentiate between different users [8, 6].

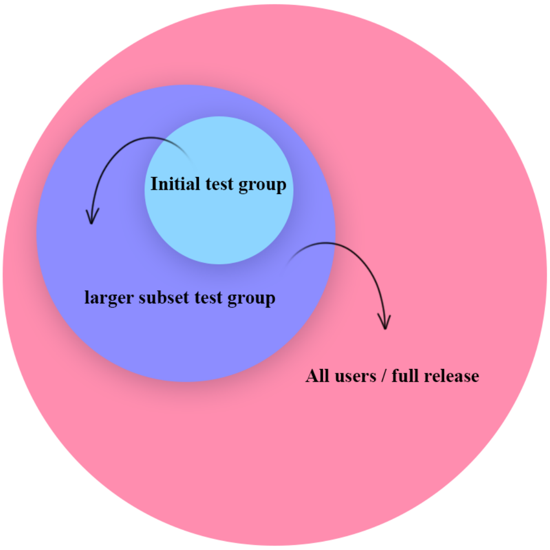

Figure 1: Gradual deployment to users with Canary testing

3 Benefits of Canary Testing

By using canary testing with a Gradual deployment of new updates developers can mitigate the risks of colossal total system failure and minimize the damage of uncaught errors. There are also other ways of gradually rolling out new updates other than canary testing such as Rolling Deployment and Blue/Green deployment [9].

Canary testing allows developers to gradually roll out and evaluate updates and or new features of a service to a small subset of the user group. By doing this developers can test the update in a live production environment and get real user test data while not risking the entire service going down [8]. Targeting specific user groups can have certain benefits over methods such as Rolling deployment since developers have control over which users will get a certain update. Developers can deploy new test builds with certain features that they think may be useful for a certain subset of users that are more likely to interact with a given new feature and thus get more test data per user.

This concept of target group evaluation can be extended further to optimize and compare how changes of an update affect different target demographics, and through this gained knowledge optimize the app to adapt and better suit each user's needs and preferences. A deployment method that allows developers to compare different versions can be a powerful tool for user experience or behavioural optimization [5], especially in cases where users are unaware that they are being tested on, a so-called one-sided blind test [4].

4 Potential Ethical Concerns

As Helena Jeret-Mäe writes in her article *The 10 Commandments for ethical software testers* [3], it's very important to know which interests your testing serves. With this power of targeting user groups, developers may run the risk of potentially running tests on users that does not serve the interest of the user. With the ability to target and test releases on specific groups, investors and stakeholders in a service have the ability to push developers to experiment with features that target certain demographics to optimize profits, retention and/or other metrics. While canary testing can be used to improve the user experience and to quickly roll out updates while mitigating potential damage, it also has the potential to be used in an non-ethical manner against the interests of the users by analyzing specific target demographic user data with the intent of manipulating user behaviour to further company goals [5].

An example of how this user-targeted tests could hypothetically be used for morally bad practices is to single out and find users with behaviour and tendencies linked to being prone to gambling addictions and then with these users test slight variations of micro-transaction gambling features in games or other services, and through metrics steer the app, not in the direction that most users would enjoy but towards a path whereas many gambling prone users as possible spend as much money as possible.

Having an easy way of testing software without risking widespread failure could possibly open up for developers to be careless about the frequency of bugs due to pressure and potential rush jobs [2]. Developers may be rolling out a new test version to users without properly evaluating the changes in the service, how the update may affect the users, their ability to use the service and the reason behind the test.

It's the users that will be affected when performing canary tests and they are the ones running the risk of encountering issues when a new update rolls out. I think that users may be able to stand behind and accept running into potential bugs and changes with a service if they feel like it is done for the betterment of their experience and their long term benefit. But if users view a test or a new feature test as mostly or only being done in the self-interest of the company it may at best create resentment and negative views towards the company and at worst may be seen as ethically dubious or wrong by the user base.

When customers pay for a service they expect it to work and for it to be stable. I think that if users are not aware that they are being experimented on they may get frustrated when they realize that their version looks or behaves differently from their peers' versions or from what they find written about the service on the internet. I think a lack of transparency in how these tests are performed and how the user base is being used may lead to user mistrust and negative brand view.

5 Possible Mitigations

When faced with an ethical dilemma one of the best ways to approach it is to perform a risk analysis before deciding how to go forward [2]. Does the tests benefit stakeholders and investors by manipulating user behaviour or does the test provide a new feature set that may aid in improving user work efficiency and well being? Most companies are for-profit organisations and have the goal of making money. But if a test seems to be too heavily weighted towards serving the companies interest without giving much of any benefit to the user base, then rolling out canary tests might not be the right solution.

Developers may not always be in control over how canary tests are being used. They may be forced by upper management to do or perform certain test practices that may be viewed as ethically dubious. By performing risk analysis of a concerning situation when engaging in tests and possibly adjusting according to the analysis, it may be possible to mitigate negative reactions from users. I think that if a new version ready for canary testing is considered to be in a risk zone or potentially not beneficial for users or that it risks heavily impacting their experience negatively while not serving their interest, then adjusting the test version may be in order or it may be good to suggest another way of evaluating the new features to avoid engaging in ethically questionable practices.

I propose another way of mitigating risks of negatively impacting user relations; which is to give the user base some autonomy in their participation in the canary tests. Developers can incentivise users to participate in testing by allowing them to opt into special pre-release builds specifically for testing. An example of users having autonomy in their participation in canary tests can be found in how 343 Industries, the developers in charge of the game franchise Halo, has an opt-in play-testing option for players to test out new things with the knowledge of it possibly having bugs and issues called "The Halo Insider Program"[1]. Having an inner circle of testers who is opt-in allows users to have the option to opt-in or opt-out of testing new features. It allows users to evaluate if it's worth the risk of bugs to gain access to new features faster. I think some power user may find it worth it while some more casual users may not have a desire to get the newest things as fast as possible and values a stable experience higher. I believe that lending users some autonomy they will feel more in control over their experience and if a user participating in a canary test runs into issues they will have the autonomy of opting out and won't project negative thoughts towards the service provider.

I think it can also be a good practice to be transparent about the tests and the metrics used in the canary tests. That way developers can be scrutinized and held accountable for their testing practices by their users and if some practices are ethically questionable the pressure from a user base may be helpful if developers wish to change things in their testing practices when discussing it with upper management.

Alternatively one could introduce the ability for users to opt-out of canary tests rather than to opt-in. A key aspect in making ethically sound canary tests is to give users some agency. If users are negatively affected by a test that they have no control over, they will likely project their frustration towards the company, provider and or developers of the service. It's important to make the users feel like they are in control of their user experience and if they feel like they are negatively impacted by canary tests they can opt-out. Prioritizing user agency means that users can do something about their frustration if they do not approve of a company's testing, practices or how they are affected by it and thus instead of projecting their negative feelings towards the developers they can do something about it.

6 Conclusion

Canary testing allows developers to quickly iterate and test new features without affecting the entire user base. [8] But with the power of targeted testing on subgroups of users, developers run the risk of engaging in ethically questionable practices if canary testing is not approached and handled with caution. I suggest three main things developers can do to try and run ethically sound canary tests:

- Run a risk analysis. Evaluate who's interests the test serves and how the tests can affect the users.
- Be transparent about how the tests are carried out and what methods and metrics are used to evaluate the tests.
- Give users the autonomy to opt-in or -out of testing.

If not careful, canary tests, like other testing methods can be used to pray on user behavioural analysis to optimize for profit, retention and engagement due to the user-centred nature of canary testing. Developers must make efforts to evaluate who's interests a test serves and how it may affects it's users.

Even the best developers can sometimes make bad decisions and not realize that they might be using the users in an non-ethical way. I therefor think that developers should make efforts to disclose how they run their tests and what metrics are evaluated to keep themselves accountable to the users in case they are blinded by their own actions. I furthermore think that developers should also prioritize user agency in their choice of participation in canary tests. I believe that doing all of these things will aid developers in creating ethically sound canary tests and allow developers to avoid engaging in ethically dubious practices when utilizing the power of canary testing.

References

[1] 343 Industries. 2022. Halo Insider Program. Retrieved May 15, 2022 from https://www.halowaypoint.com/halo-insider.

[2] Brian Berenbach, M. B. Professional and Ethical Dilemmas in Software Engineering. Retrieved May 30, 2022 from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=4755159.

[3] Jeret-Mäe, H. The 10 Commandments for ethical software testers. Retrieved May 30, 2022 from https://nortal.com/blog/10-commandments-ethical-software-testers/.

[4] Kuhn, G. 2020. What is Blind Testing in Market Research? Retrieved May 16, 2022 from https://www.driveresearch.com/market-research-company-blog/what-is-blind-testing-in-market-research-customer-experience-cx-syracuse/.

[5] Optimizley. A/B testing. Retrieved May 30, 2022 from https://www.optimizely.com/optimization-glossary/ab-testing/.

[6] Optimizley. Canary Testing. Retrieved May 28, 2022 from https://www.optimizely.com/optimization-glossary/canary-testing/.

[7] Optimizley. Feature Flags. Retrieved May 28, 2022 from https://www.optimizely.com/optimization-glossary/feature-flags/.

[8] Team LaunchDarkly. 2020. What Is Canary Testing? A Detailed Explanation. Retrieved May 15, 2022 from https://launchdarkly.com/blog/what-is-canary-testing-a-detailed-explanation/.

[9] Tozzi, C. 2020. When to use canary vs. blue/green vs. rolling deployment. Retrieved May 15, 2022 from https://www.techtarget.com/searchitoperations/answer/When-to-use-canary-vs-blue-green-vs-rolling-deployment.