JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Risks of Manual Integration Testing in the Context Rapid Change

ARTICLE

By Steve Berczuk | November 22, 2010

SUMMARY

You have probably come across a situation like this: It's close to a release deadline. The QA team is testing. Developers as testing and fixing problems, and everyone is focused on getting the best product they can out the door on time. During this time, you may notice that someone on the QA team, while working late has found an interesting problem. Unfortunately, your team found and fixed the problem hours before.

You have probably come across a situation like this: It's close to a release deadline. The QA team is testing. Developers as testing and fixing problems, and everyone is focused on getting the best product they can out the door on time. During this time, you may notice that someone on the QA team, while working late has found an interesting problem. And they clearly spent a lot of time investigating the problem, identifying the expected results, and the details of why what's happening is wrong. If this is a data intensive application, there may well be SQL queries included to allow you to pinpoint the issue quickly. In short, an ideal problem report.

Except for one thing. Your team found and fixed the problem hours before. And the effort to find and document the problem could have been spent on something else.

It's hard to avoid this kind of overlap when you don't have a complete end-to end automated testing process. And it's probably impossible (for now, anyway) to create an automated test process that will completely replace exploratory testing. Any manual testing process has to balance the timeliness with the "flow" of the testers. If you update a deployment hourly, then you'll reduce the risk of redundant bug reports, but the testers will experience too many interruptions. If you have a paradigm where you code, deploy, and stop coding until you see the next set of issues, but that means time wasted, and

We use cookies to ensure you get the best experience on our website.
Read More
Accept

You're still early in adopting agile, you'll likely have a week or two at the end of a project where you'll have more effort put into manual testing, and that manual testing will find errors. So the simple thing to do is to query among the team before spending too much time documenting an issue.

Some of the options are:

- Searching the issue tracking system. This sound like a good idea, but sometimes it's hard to find the right query. And it's possible that something got fixed without and issue being generated.
- Asking. Yelling out a question, sending and email, or posting a message in a team chat room gives you the benefit of being able to be a bit vague in your query.

I tend to prefer the "ask the room" approach. But this isn't effective when you're working in different time zones, or at different hours. On the other hand, since software development is a collaborative activity, you'll need to address the question of how yo collaborate across time. (The simple answer is to try to avoid close coupling between activities that are likely to happen when the teams are on different schedules).

Situations like the one that I described can happen. But it's worth figuring out how often they happen, and if there are simple ways to reduce their incidence and impact. Good communication channels are importanty and sometimes the lower tech ones work better.

Topics: change management,

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

UPCOMING WEBINAR

We use cookies to ensure you get the best experience on our website.

LETS HANG!

FEATURED RESOURCES

SECURITY

Shifting Security Left: DevSecOps eGuide

White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

0 comments

YOU MAY ALSO LIKE

SECURITY

Agentic AI Opens Up New Security Flaws

Nahla Davies | August 27, 2025

We use cookies to ensure you get the best experience on our website.

SECURITY

Countering the Threat of Ransomware-as-a-Service (RaaS)

Nahla Davies | August 22, 2025

SECURITY

Building Trust in Blockchain: The Role of Smart Contract Testing

Ekaterina Egorova | November 18, 2024

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.