



DO MORE WITH LESS IN SOFTWARE TESTING

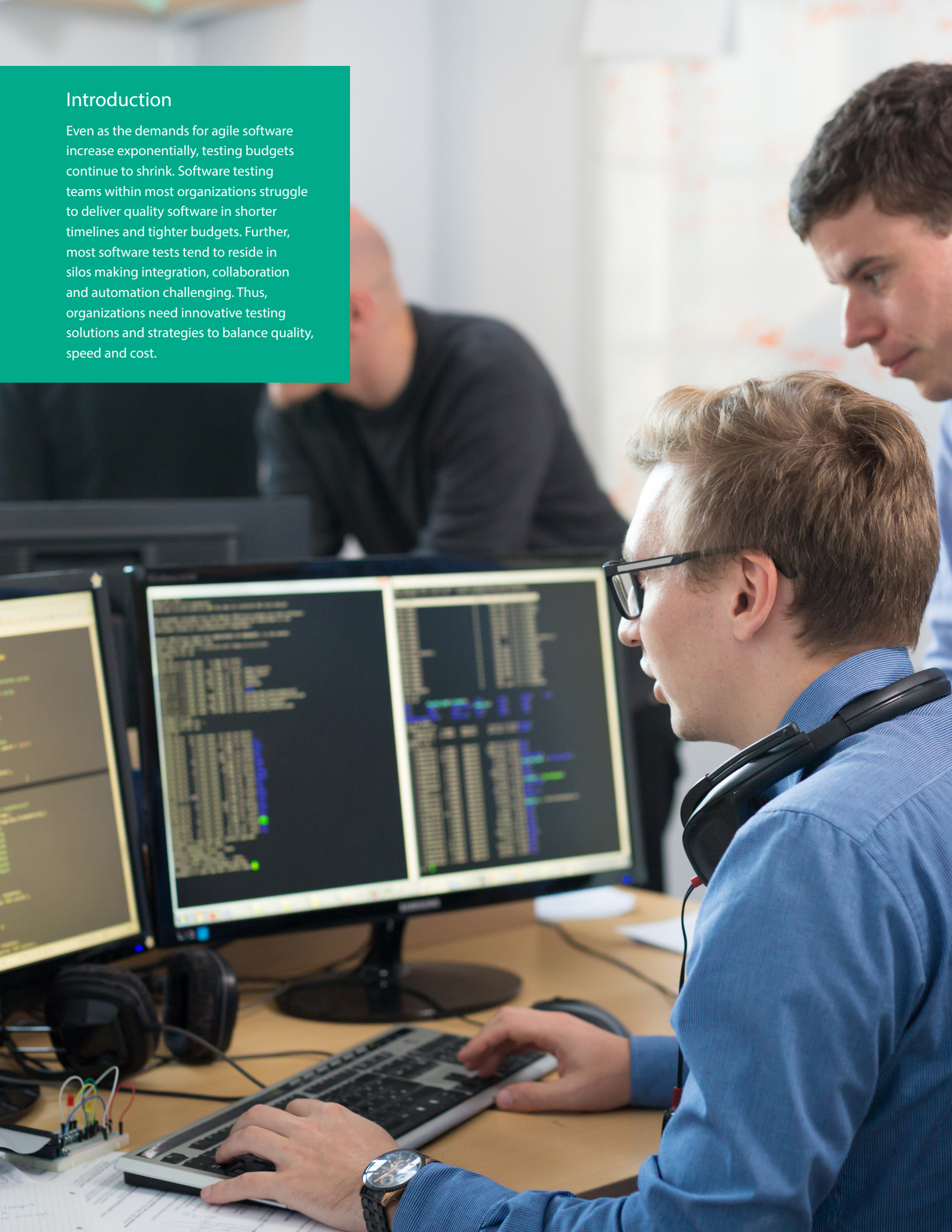
Indumathi Devi

Abstract

Faced with pressure to deliver more despite ever-shrinking budgets and shorter timelines, most companies struggle to balance the cost of innovation with business demands. Testing is a critical area where neither speed nor quality of output can be compromised as this leads to negative business impact. This paper explains some cost-effective strategies that enable testing organizations to improve efficiency within their testing teams while ensuring high-quality output.

Introduction

Even as the demands for agile software increase exponentially, testing budgets continue to shrink. Software testing teams within most organizations struggle to deliver quality software in shorter timelines and tighter budgets. Further, most software tests tend to reside in silos making integration, collaboration and automation challenging. Thus, organizations need innovative testing solutions and strategies to balance quality, speed and cost.



Agile testing solutions and strategies

1. Test with an end-user mindset

The job of testing goes beyond checking software against preset requirements or logging defects. It involves monitoring how the system behaves when it is actually being used by an end-user. A common complaint against testers is that they do not test software from the perspective of business and end-users. Effective risk-based testers are those who understand the system's end-users and deliver value-added testing services that ensure quality products and meet clients' expectations. To do this, testers must evaluate products

by undertaking real business user journeys across the system and test commonly used workflows in short testing windows. By mimicking real user journeys, such testers identify higher number of critical production defects.

2. Empower cross functional teams

Agile and DevOps methodologies in software testing are forcing teams to work together across the software development lifecycle (SDLC). Engendering test independence is not about separating testing from development as this can

lead to conflicts between developers and testers. Most teams have a polarized dynamic where testers search for defects and must prove how the program is erroneous while programmers defend their code and applications. Cross-functional teams eliminate such conflict by gathering members with different specializations who share accountability and work toward common goals. For instance, 'testers' are simply team members with testing as their primary specialization and 'programmers' are those within the team who specialize in coding.



This team structure encourages people to work with a collaborative mindset and sharpen their expertise. Such teams can be small, with as few as 4-8 members who are responsible for a single requirement or part of the product backlog. Cross-functional teams provide complete ownership and freedom to ensure high-quality output – an important step to realizing the potential of agile.

3. Automate the automation

Running an entire test suite manually is time-consuming, error-prone and, often, impossible. While some companies are yet to on-board agile and DevOps capabilities, others have already integrated

the practice of continuous integration (CI) and continuous delivery (CD) into their testing services. Irrespective of the level of DevOps maturity, CI/CD will provide only limited value if not paired with the right kind and degree of testing automation. Thus, organizations need a robust, scalable and maintainable test automation suite covering the areas of unit, API, functional, and performance testing. Automated testing saves effort, increases accuracy, improves test coverage, and reduces cycle time. To ensure automation success, organizations must focus on:

- Automating the right set of tests, particularly business-critical end-user journeys and frequently used workflows

- Integrating various components that may change continuously and need to be regressed frequently
- Automating data redundant tests
- Using the right set of automation tools and frameworks
- Moving beyond just the user interface and automating unit, APIs and non-functional tests
- Continuous maintenance and use of automated test suite

On its own, test automation is important. However, when automation is integrated with a CI/CD pipeline to run every time new code is pushed, the benefits in time, cost and quality are multiplied.

Conclusion

In a fast-paced and evolving digital world, companies want their IT partners to do more with less. When it comes to software testing, this places heavy pressure on software testers who are required to push high-quality code faster at lower cost. To streamline software testing, organizations need an approach where their testers adopt an end-user mindset by testing real user journeys and critical business transactions. They must also create efficient cross-functional teams that collaborate to achieve common goals and deliver value-added testing services. Finally, automating different layers of testing and practicing CI/CD will facilitate continuous testing and reduce time-to-market. These cost-effective strategies will help software testing professionals improve productivity and deliver more with less.



About the author

Indumathi Devi is a Project Manager at Infosys with over 15 years of experience in software testing. She has effectively executed multiple software testing projects. Working in different domains and technology stacks, she has assisted numerous clients in implementing robust software testing solutions for manual as well as automated testing.

For more information, contact askus@infosys.com

Infosys[®]
Navigate your next

© 2018 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.