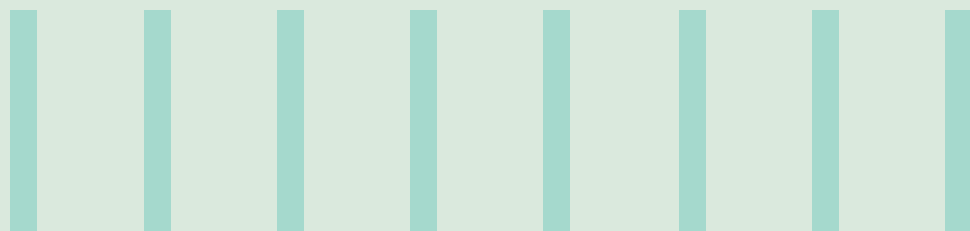




SCALING CONTINUOUS TESTING ACROSS THE ENTERPRISE



Abstract

Over the years, organizations have invested significantly in optimizing their testing processes to ensure continuous release of high-quality software. Today, this has become even more important owing to digital transformation. This paper examines some of the critical features of setting up a robust continuous testing practice across the enterprise. It considers the technology and process standpoints and provides guidelines to ensure a successful implementation from inception to delivery of high-quality software.

Introduction

Establishing a continuous testing (CT) practice is an organization-level change that cannot be executed with a 'big bang' approach. It should be conceptualized, implemented and allowed to mature at a program level. Once continuous testing gains inroads into various programs across the enterprise, it can then become the new norm for enterprise software delivery. To enable a seamless journey towards organization-wide continuous testing, the

gradual change must be sponsored and supported by senior IT leadership.

To begin with, enterprises must set up a program-level center of excellence (CoE) and staff it with people who have a deep understanding of automation. This team should be responsible for:

- Identifying the right set of tools for CT
- Building the right solutions and

practices that can be adopted by the program

- Integrating automated tests with the DevOps pipeline for continuous testing

These action points lay the foundation for establishing CT at a program level. They can subsequently be improved and aligned depending on the enterprise's needs.

```
input: Asus WMI hotkeys as /devices/platform/asus-nb-wmi/input/input10
EXTS: fs (dm-b): mounting ext3 file system using the ext3 subsystem
EXTS: fs (dm-b): mounted filesystem without journal. Opts: (no)
audit: type=1400 audit(1400709056.157:2): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/snapd/snap-confine" pid=930 comm="ap
armor_parser"
audit: type=1400 audit(1400709056.161:3): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/snapd/snap-confine/mount-namespace-
capture-helper" pid=930 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:4): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/sbin/cups-browsed" pid=932 comm="apparmo
r_parser"
audit: type=1400 audit(1400709056.161:5): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/sbin/dhclient" pid=920 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:6): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action
" pid=920 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:7): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=9
28 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:8): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid
920 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:9): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/lightdm/lightdm-guest-session" pid=9
27 comm="apparmor_parser"
audit: type=1400 audit(1400709056.161:10): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/lightdm/lightdm-guest-session/chro
mium" pid=927 comm="apparmor_parser"
USB 2-1.0: USB disconnect, device number 4
Bluetooth: GMP (Ethernet Emulation) ver 3.3
Bluetooth: GMP filters: protocol multicast
Bluetooth: GMP socket layer initialized
usb_labeled: (C) 2000-2004 Netfilter Core Team
Enabler V2.0 registered
nf_conntrack version 0.9.0 (14304 buckets, 45536 max)
nfnetlink: automatic filtering via arp/ndp/ipsetables has been deprecated. Update your scripts to load br_netfilter if you need this.
Netfilter messages via NETLINK v0.30.
ip netns protocol 0
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
asix 3-1:1.0 eth0: register 'asix' at usb-0000:00:14.0-1, ASIX AX88772 USB 2.0 Ethernet, 9c:ec:b0:07:df:b0
usbcore: registered new interface driver asix
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
asix 3-1:1.0 eth0: link down
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
asix 3-1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0xCPE1
audit: br_netfilter: callchain suppression
audit: type=1400 audit(1400709056.169:11): apparmor="DENIED" operation="open" profile="/usr/sbin/mysqld-akonadi:///usr/sbin/mysqld" name="/etc/mysql/my.cnf" fa
tality=1 back=144 comm="mysqld" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
initial 0000:02:00.0: AR_MSI bit toggled to disable radia.
USB 2-1: USB disconnect, device number 3
USB 3-1: USB disconnect, device number 2
USB 3-1: USB disconnect, device number 2
asix 3-1:1.0 eth0: unregister 'asix' usb-0000:00:14.0-1, ASIX AX88772 USB 2.0 Ethernet
tratarcraft=3
```


The journey

Once the CoE has been set up, enterprises must focus on expanding the practice of continuous testing within all program-level agile teams across the organization. This can be done by promoting the existing program-level CoE into an enterprise-level CoE with broader responsibilities.



Fig 1: Continuous testing journey of an enterprise

The primary goal of the enterprise-level CoE should be:

- To ensure CT adoption by all scrum teams
- To establish real-time reporting, metrics and measurement for faster adoption
- To identify areas with zero/low automation and provide support

Enterprises can also accelerate CT adoption by extensively showcasing the benefits realized from programs.

Challenges during enterprise adoption

Extending continuous testing operations across the enterprise can be daunting. Enterprises should be prepared to face resistance to change arising from higher operational costs and limited technical knowledge. Some of the key challenges of CT adoption and the ways to address them are listed below:

| Challenges during enterprise adoption | The way forward |
|--|--|
| Multiple automation solutions – Teams often devise their own ways to implement frameworks, leading to redundant code, confusion and wastage | During the initial assessment phase, enterprises can invest in developing a solution that is capable of porting with existing enterprise code |
| Low technical knowledge for automation – QA organizations often don't have the necessary skills to build and maintain automation scripts | Focus on talent enablement through brown-bag meetings, training workshops and support. Enterprises can also use script-less and model-based automation tools |
| Pushback from certain lines of business (LOBs) – Most enterprises use disparate technology stacks and execution methodologies. It is more challenging to adopt CT in some domains on legacy such as mainframe systems, PowerBuilder environments and batch jobs | Build a robust automation framework that supports different technology stacks for an improved usability experience. Define different goals for different technology stacks (like lower automation goals for legacy and higher automation targets for APIs) to ensure parity across teams during adoption |
| Limited funding – Many customers view continuous testing as an expense rather than an investment since it includes automation development as well as integration costs | Enable individual QA teams to adopt in-sprint automation, thereby transferring the high initial cost to individual projects. This reduces the overall enterprise CT cost to the expense of integrating automation with the DevOps pipeline |

Infosys approach on establishing CT

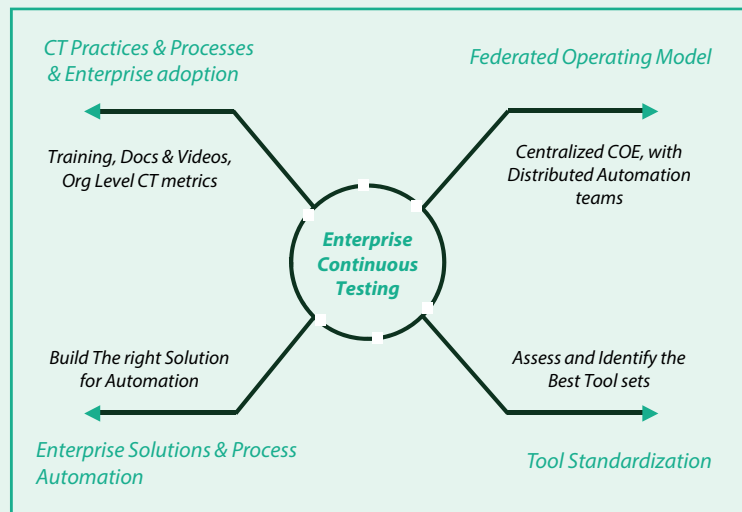


Fig 2: Key dimensions for establishing a successful CT practice

Based on our past experience across several engagements, Infosys has identified four main dimensions that are essential to establishing and operating a successful continuous testing practice across any enterprise. These dimensions are described below.

1. Establish a federated operating model

A robust operating model is vital for any practice. Earlier models comprising functional testing teams and shared automation teams failed because these hamper functional teams from achieving in-sprint/progressive automation within 2-3-week sprint windows. Thus, the way

forward is to embrace automation as a culture and consider it as a business-as-usual process for anything that a tester does.

Care must be taken to ensure that generic and complex solutions are built and maintained by a centralized CT CoE team.

In their daily tasks, regular QA teams must adapt to artifacts developed by the CoE team. Through this federated method of operations, enterprises can simply deploy a few people with specialized skills to create CT artifacts, thereby supporting regular testers with relative ease.

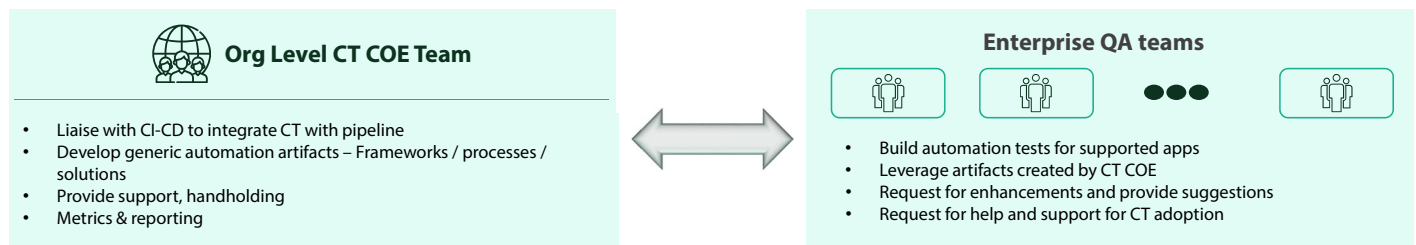
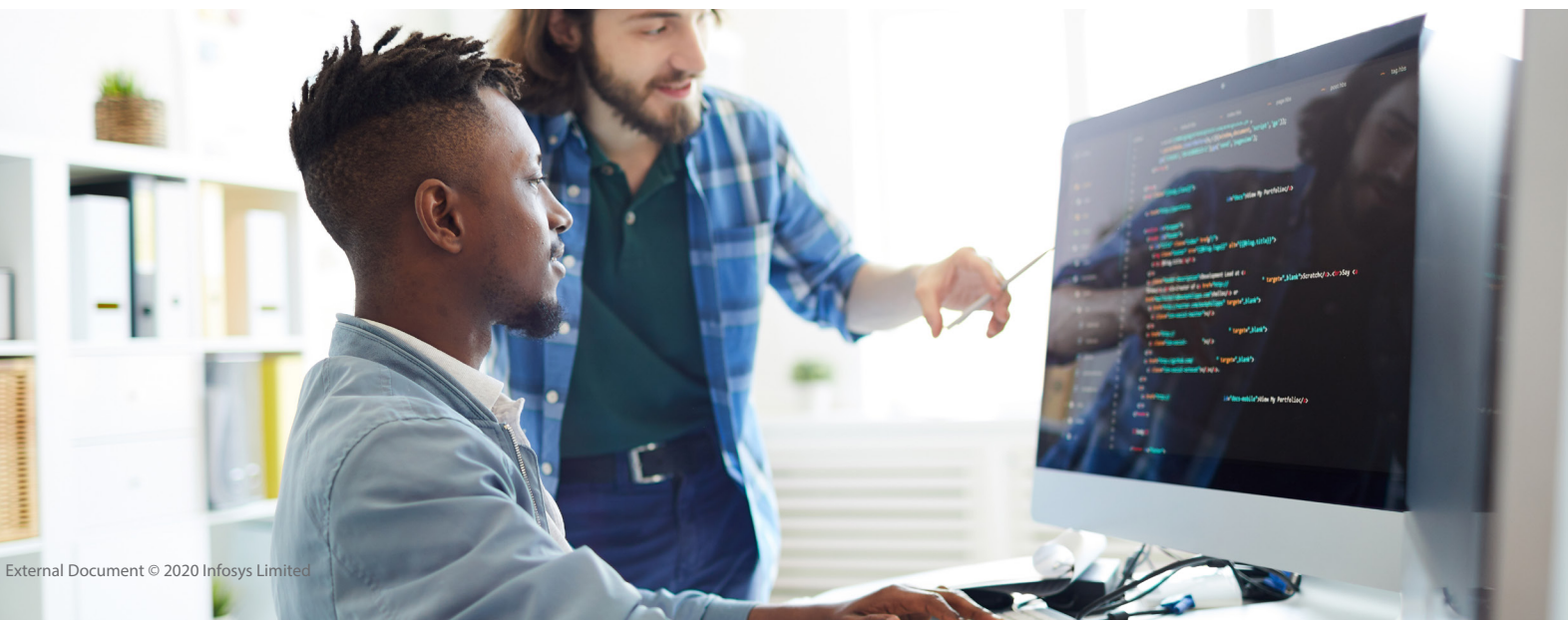


Fig 3: Structure, roles and interactions in a federated operating model with centralized CoE



2. Standardize tools

The process of standardizing tools should be objective. It should first assess the existing enterprise tools, identify those that can be reused and procure new tools for CT. Here are some important considerations when evaluating any tool:

- Tool fitment – Ensure the tools are the right ones for the task they have to service
- Tool license and operational cost – This should fall within the client's budget
- Tool scalability – It should be capable of scaling to meet future organizational needs
- Tool integration – In a DevOps world, tools should be easily compatible with other tools and pipelines

- Tool support – There should be adequate online and professional support services for the tool

Currently, there are many tools available in the market that support the various stages of the software development lifecycle (SDLC). Enterprises must be careful to onboard the right set of tools that will aid software development and QA.



3. Build enterprise and process automation solutions

The automation solutions built to aid and enable CT are at the heart of any robust continuous testing practice. These solutions should encompass every phase

of the software testing lifecycle. Some critical automation artifacts that must be developed to sustain a robust CT practice are:

a) Automation frameworks – This is the basic building block that aids automation and enables CT across an enterprise, as shown below in Fig 4.

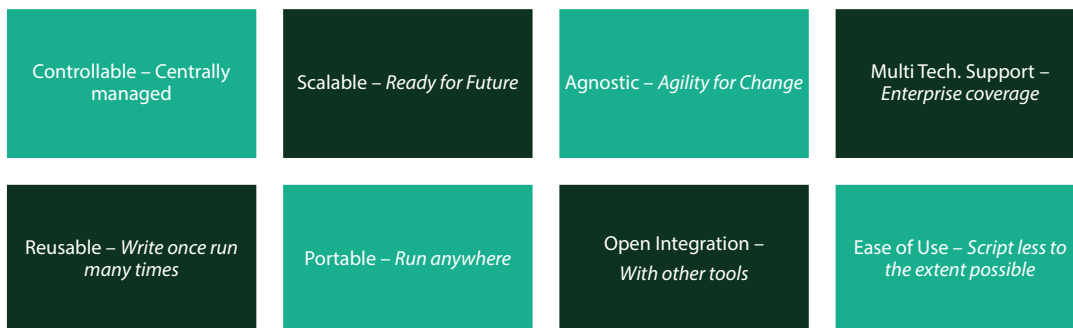


Fig 4: Characteristics of a robust automation framework

b) Distributed and parallel test execution approaches – Speedy test execution is critical to accelerate software delivery. As the volume of automated tests increases, enterprises should adopt distributed and parallel execution either by onboarding off-the-shelf tools or by building custom solutions as per enterprise requirements

c) Test data automation – Testers spend a significant amount of time setting up test data. Solutions should be built to automate the manufacturing, cloning/masking,

mining, and management of test data from requisition to provisioning

d) Process and environment automation – This involves automating all test and defect management-related processes. Environment provisioning automation is essential to make the entire CT practice cost effective and manageable at an enterprise scale and some viable options are cloud-based or infrastructure virtualization solutions.

It is important to note that automation solutions are simply a critical subset of all the solutions that should be developed as a part of establishing a successful CT practice. Care must be taken to prioritize developing these solutions and weighing their benefits according to the needs of the enterprise.

4. Establish enterprise CT processes and drive adoption

Successful products are the sum total of their technologies, processes and best practices, which make products attractive and easy to adopt. Having explored the technical solutions and artifacts for CT, let us examine some of the critical CT processes and strategies for enterprise adoption.

a) CT processes – Robust processes act as a pointer for all who embark on the CT journey. The key processes to be established are:

- Day-wise activity tasks for helping teams adopt in-sprint automation, as shown in Fig 5

- CT metrics, measurement and reporting like overall automation coverage, in-sprint automation percentage, percentage of defects found from automation, execution time per run, code quality, and code coverage

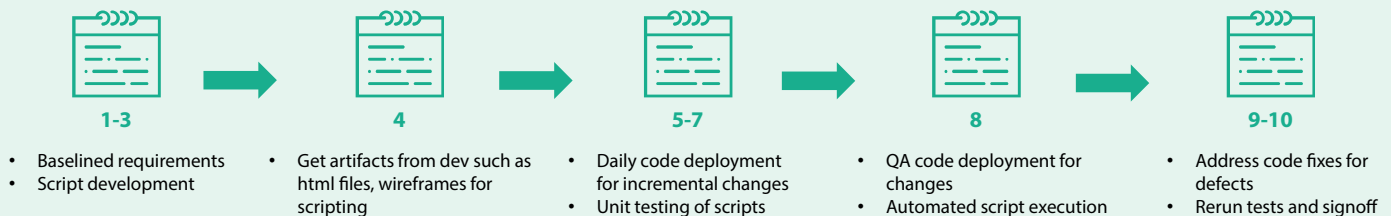


Fig 5: Daily tasks during a 10-day sprint

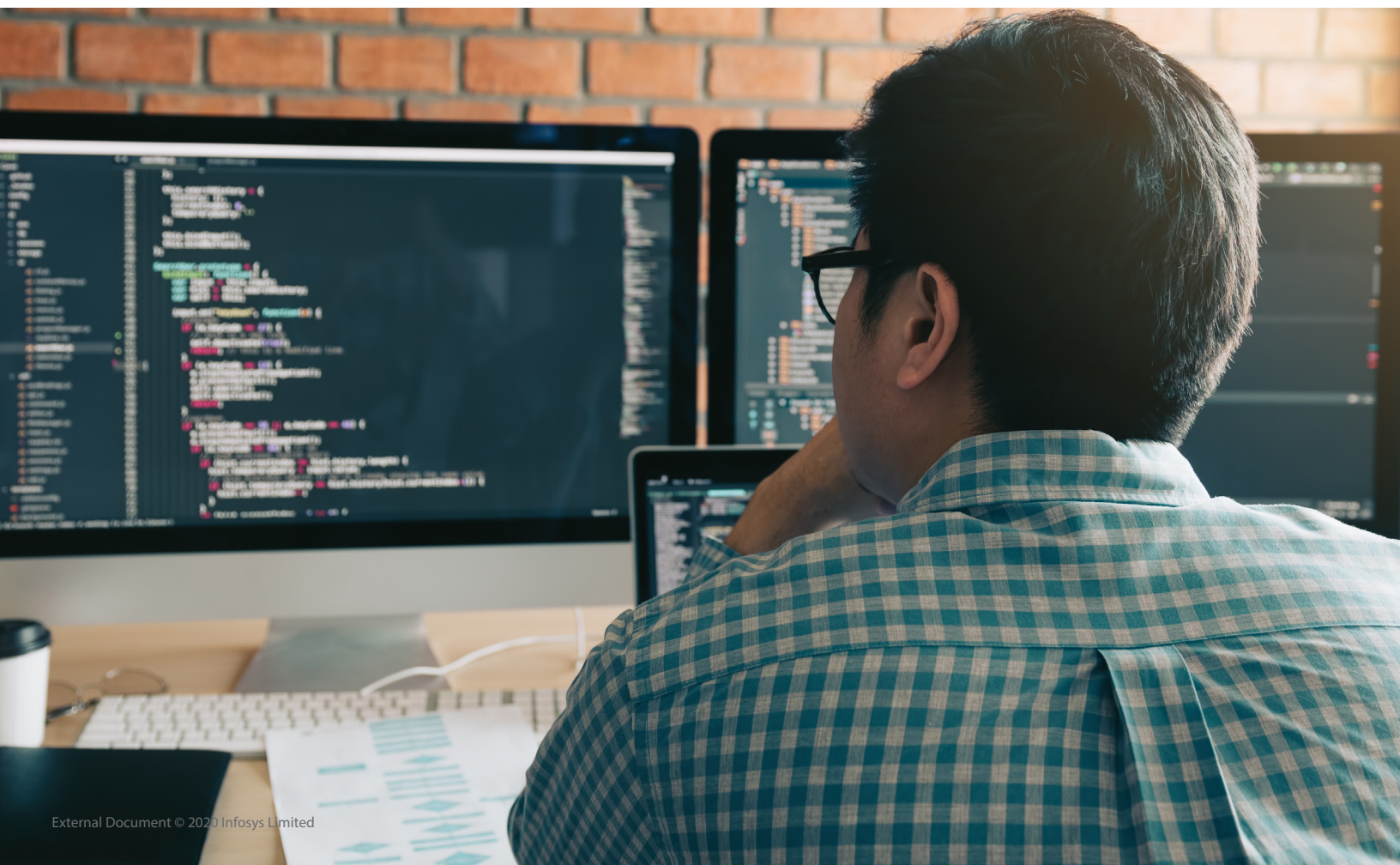
b) Adoption strategies – Implementing CT across the enterprise can be facilitated through the following ways:

- Providing organization-level branding like communications, mailers and workshops as well as talent enablement and support through brown-bag meetings,

demos, trainings, self-learning videos, playbooks, etc.

- Accelerating adoption through centralized CT metrics reporting that introduces healthy competition, faster adoption and early identification of problem areas

We, at Infosys, believe these dimensions can help enterprises set up and operate a mature continuous testing practice. Nevertheless, it is important to customize these according to the needs of each organization.



Conclusion

Establishing a continuous testing practice across the enterprise comes with its own challenges and complexities. These include insufficient knowledge of automation, limited funding, resistance to change, and disparate technologies. When developing a strategy for enterprise-wide adoption of CT, Infosys recommends paying attention to four critical dimensions. These include creating a federated operating model, standardizing tools, building automation solutions, and establishing CT processes. Through our approach, enterprises can benefit from a roadmap to successfully implement CT with a dedicated center of excellence. The successful adoption of continuous testing mandates changing how people work and embracing automation as part of the organizational culture and business as usual.

About the Author

Mohanish Mishra, Senior Project Manager – Infosys Limited

For more information, contact askus@infosys.com



© 2020 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.