eBOOK

TRIGENT
OVERCOMING LIMITS

Adapting Agile and DevOps Principles for Business Agility

Table of Contents

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Did You Know?

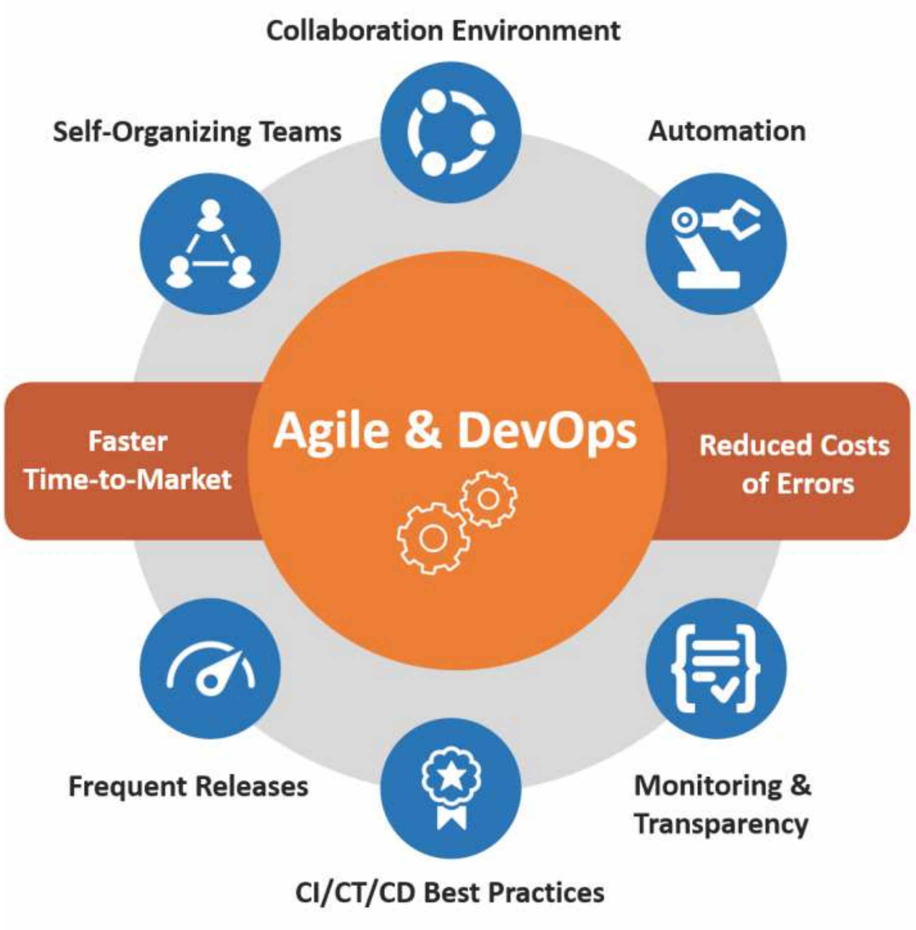
- High-performing IT organizations following DevOps practices experience 60X fewer service failures and recover 168X faster than their lower-performing peers
- DevOps helps organizations deploy 30X more frequently with 200X shorter lead times
- Technology that supports the DevOps toolchain is predicted to grow 21% worldwide
- By 2018, agile and iterative development will require 30% of resources for application testing engagements to be on-site or onshore
- About 70% of organizations in Asia-Pacific and Japan have started implementing DevOps

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Agile & DevOps Practices – An Overview

Speed to market is essential, when it comes to software development. Users will abandon an application and the poor customer experience can hurt a brand, making software testing integral to achieving speed and quality. However, the challenge is to enable rigorous testing that fits into the same sprint, while allowing stakeholders-from testers to business analysts-to align and remain flexible. This is a tall order that requires replacing the typically slow, manual and error-prone testing process with a powerful, model-based approach.

High performing IT organizations are therefore adopting the DevOps methodology. According to the State of DevOps Report, organizations following DevOps practices experience 60X fewer services failures and recover 168X faster than their lower-performing peers. To ensure agility, Gartner predicts that by the year 2020, 50% of customers will have dismantled test centers of excellence (TCoEs) as they shift to competency centers, agile methodologies, and DevOps to provide business agility.



© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Evolution of QA & Testing in an Agile & DevOps World - Why and How?

Most of us will acknowledge the fact that Waterfall was the methodology of choice and the role of QA varied from being an integral part of the team to a grudging necessity. In the Waterfall era, a single defect, could result in the failure of a solution. Waterfall meant that QA would not be involved in the development process until the necessity arose to actually test the product. Requirements, if they existed, were incomplete and QA had to put the puzzle together by making assumptions. Business stakeholders did not have to interact with QA and roadblocks resulted in a dysfunctional environment.

Agile has helped to overcome the challenges that Waterfall methodology posed. A significant difference between Waterfall and Agile is the participatory environment that Agile provides, balancing the desire for speed with the requirement for quality. This is accomplished by forming a small, focused team, including QA, that collectively looks at the requirements and determines what can be accomplished within a set period of time. Daily meetings, along with frequent informal discussions, ensures that everyone is on the same page. A product owner is available to respond to business questions raised by the team and a Scrum Master ensures that all roadblocks are removed so that the team can focus on deliverables. Typically, what is accomplished within that time frame is a distinct piece of functionality that is fully formed. To that end, the team as a whole is held responsible for ensuring a product that is as defect-free as possible. This approach ensures that QA must have an equal place at the table, since a defective deliverable means missed stories, and a failed sprint.

Now software delivery moves faster than ever, thanks to a combination of Agile practices and continuous delivery. Continuous delivery, an offshoot of Agile practices is about delivering more value to clients. In a continuous delivery method, the role of QA has evolved to include an expanded focus where QA is involved in all stages of project development including release management, release planning, server admin and devops, requirements analysis, performance testing, and security testing. CD demands a ‘shift-left’ approach to QA whereby testing is only one part of the QA process and plays an active role in discovery, design, requirements gathering, and estimation.

Pitfalls of transitioning from Waterfall to Agile

As more and more companies and teams transform to agile, the challenges become more diverse and affect how teams execute. Agile transformation poses challenges that span product architectures & modularization, execution velocity, timelines, release management, and the roles of product, project, and engineering managers. It is necessary to prepare teams for the change, identifying skill gaps and upskilling, revised responsibilities of team members, the roles of the team as part of enterprise agile, and the challenges of scaling agile activities.

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Importance of Adapting the Agile & DevOps Ecosystem

The key challenges that developers and testers face when they attempt to create better & faster software are:

- Ambiguous requirements
- Poor test case design and limited coverage
- Waiting for test data
- Unavailability of system components
- No automation

Rightsizing User Stories

Transitioning from waterfall or other traditional project management styles to an Agile/DevOps method, teams often feel the pain of structuring their work. User stories describe requirements for a software system to keep track of what it needs to do and the reason the requirement provides value. Teams need to pursue proven techniques to split epics and stories, so as to make them smaller and more manageable, while sticking with the scrum rule that each user story must deliver value. Fortunately, agile development uses four clear delivery approach to bring structure to any agile project: epics, user stories, versions, and sprints:

Epic	Story	Sprint	Version
Large body of work, contains stories	Smallest unit of work, also known as a task	Iteration where team does the work	The release of software to the customer

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

The Four Values of the Agile Manifesto

The Agile Manifesto is comprised of four foundational values and 12 supporting principles which lead the Agile approach to software development. Each Agile methodology applies the four values in different ways, but all of them rely on this to guide the development and delivery of high-quality, working software.

1

Individuals and Interactions Over Processes and Tools
The first value in the Agile Manifesto is "Individuals and interactions over processes and tools." Valuing people more highly than processes or tools is easy to understand because it is the people who respond to business needs and drive the development process.

2

Working Software Over Comprehensive Documentation
Agile does not eliminate documentation, but it streamlines it in a form that gives the developer what is needed to do the work without getting bogged down in minutiae. Agile documents requirements as user stories, which are sufficient for a software developer to begin the task of building a new function. Historically, enormous amounts of time were spent on documenting the product for development and ultimate delivery.

3

Customer Collaboration Over Contract Negotiation
The Agile Manifesto describes a customer who is engaged and collaborates throughout the development process, making. Agile methods may include the customer at intervals for periodic demos, but a project could just as easily have an end-user as a daily part of the team and attending all meetings, ensuring the product meets the business needs of the customer. Negotiation is the period when the customer and the product manager work out the details of a delivery, with points along the way where the details may be renegotiated. Collaboration is a forward looking approach entirely.

4

Responding to Change Over Following a Plan
Traditional software development regarded change as an expense, so it was to be avoided. With Agile, the shortness of an iteration means priorities can be shifted from iteration to iteration and new features can be added into the next iteration. Agile's view is that changes always improve a project; changes provide additional value.

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

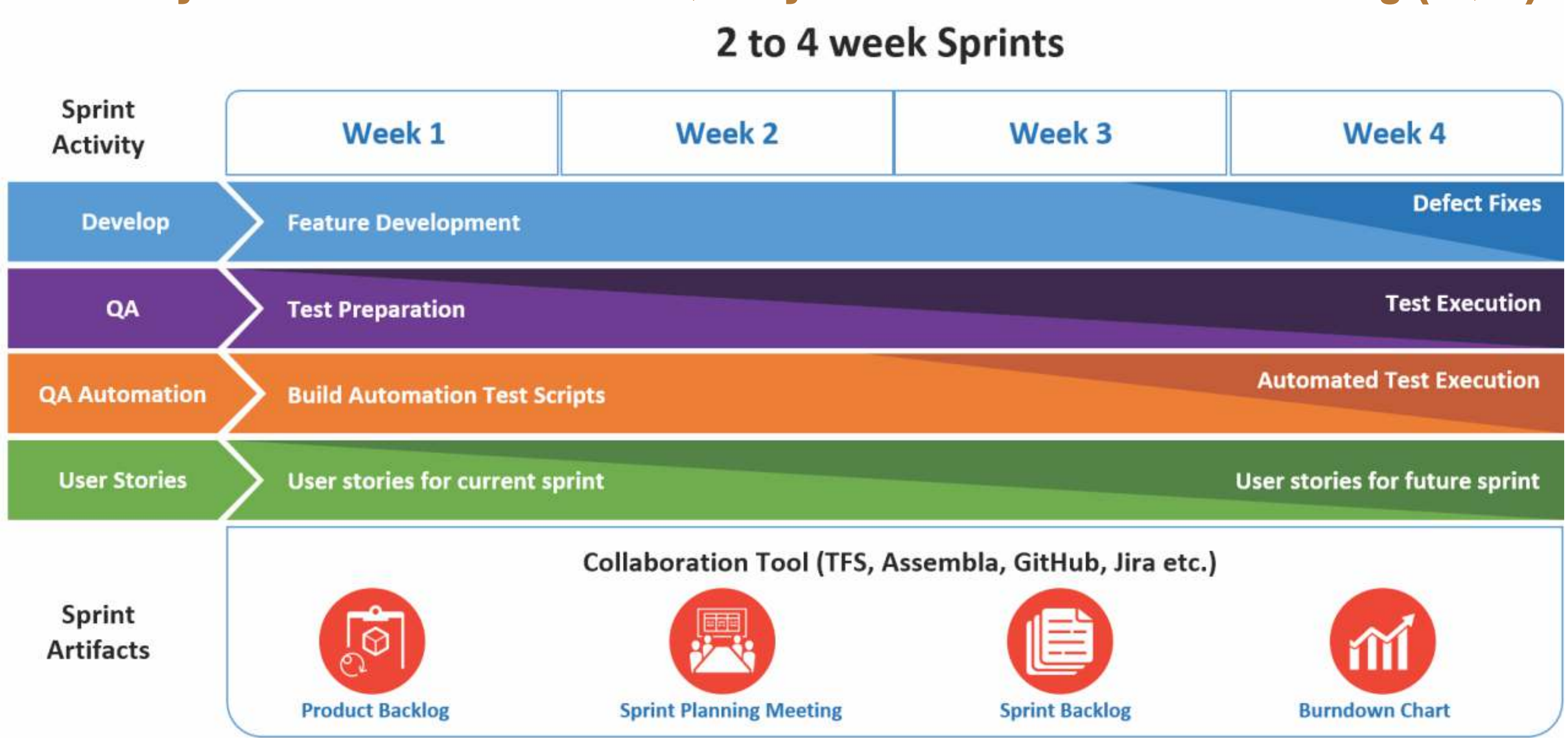
Why Trigent?

Users and customers are demanding improved software solutions at a pace that is equal to many of the cloud services that they are familiar outside their work. This strong desire for frequent updates at increased speed has forced enterprises and software companies to adapt to Agile and DevOps. Many mobile and cloud development platforms (PaaS) makes Agile/DevOps methodology as the preferred way of working.

One of the top DevOps adoption drivers is the need to increase quality, followed by the need to improve the customer experience, reduce complexity, and reduce overall IT costs.

In order for DevOps/Agile to be successful, quality initiatives must be made an integral part of the DevOps/Agile methodology. Specifically, QA roles need to be embedded into the DevOps/Agile team from the get-go; QA automation must be aligned with build/deployment automation.

Go Beyond CI/CD to Continuous Quality Assurance /Automated Testing (CQAT)



2 to 4 week Sprints

Sprint Activity	Week 1	Week 2	Week 3	Week 4
Develop	Feature Development			Defect Fixes
QA	Test Preparation			Test Execution
QA Automation	Build Automation Test Scripts			Automated Test Execution
User Stories	User stories for current sprint			User stories for future sprint

Sprint Artifacts

Collaboration Tool (TFS, Assembla, GitHub, Jira etc.)

Product Backlog

Sprint Planning Meeting

Sprint Backlog

Burndown Chart

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Continuous Integration (CI) and Continuous Delivery (CD) practices strive to make small changes, integrate them frequently and ensure fast feedback. This results in fewer failures and faster recovery from failures. But mature teams go beyond CI/CD, to a Continuous QA through automated testing. Continuous QA and automated testing begin with a comprehensive framework that defines the process, scope, and tools. Trigent has built continuous QA and automated testing framework that works reliably and frictionless within agile/scrum methodology. While the focus is to test the incremental functionality during sprints, investments are made to automate a judicious amount of testing.

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

Agile/DevOps Testing Coverage

Functional Testing

All features and functions that are accessible to internal & external users, business partners, and customers are covered through functional testing. Our testers gain insights on writing best functional tests through multiple sources, including user stories, help desk or customer support tickets, and feedback from other members of the DevOps team.

API/Service Testing

Applications today are complex and provide API and service endpoints that are often used by other applications such as Mobile Apps, partner integrations, etc to complete a business transaction. Testing these APIs and service endpoints are best achieved through automated scripts. However, these require some level programming skills that may not be available with traditional test engineers. Trigent brings test engineers skilled in multiple scripting languages and with good knowledge of REST, SOAP, XML, Web Services, etc.

Non-functional Testing

Performance load and other non-functional tests that cover the entire application stack (code, database, storage, network) are automated, results tracked and compared from one deployment to next. This helps the operations team to be assured of performance levels, but also help the developers to identify and fix performance problems before the deployment.

In general, the mindset of our Agile/DevOps QA engineers is to “build quality in” – to catch defects early and fix them cheaply. This is often achieved by “shifting left”, establishing feedback loops closer to early stages of development to ensure predictable, high-quality software delivery.

© Trigent Software Inc. All trademarks, marked and unmarked, are the property of their respective owners.

TRIGENT

OVERCOMING LIMITS

About Trigent

Trigent is a CMM Level-4 technology solutions company with its US office at Southborough, MA, and India office at Bangalore. Trigent provides comprehensive solutions for business problems via outsourced software product and applications design, development and quality assurance. Trigent serves customers like Independent Software Vendors (ISVs), enterprises and SMBs in the High Tech, Healthcare, Education, Ecommerce and Manufacturing areas. Trigent's solutions help clients overcome budget, schedule and resource constraints.
To learn more about Trigent visit www.trigent.com

Our Headquarters

Trigent Software Inc.
2 Willow Street, Suite #201
Southborough, MA 01745
+1 (877) 387-4436
sales@trigent.com

Other Location

Trigent Software Ltd.
Khanija Bhavan, 1st Floor
#49, West Wing, Race Course Road
Bangalore - 560001, India
+91 (80) 2226-3000