Seven ways to reduce waste and accelerate software delivery

How an IBM DevOps approach can help shorten software development cycles and increase customer value

IBM

Contents

Executive summary

Lean development is a culture; waste reduction is one of the results. Removing waste can improve operational efficiency, but more importantly, it can shorten development cycles and increase customer value.

Shorter cycles can improve innovation, competitiveness, and responsiveness in the marketplace. They can also provide a valuable opportunity for learning and continuous improvement for development teams.

Many businesses use IBM DevOps capabilities with lean development principles to successfully reduce waste. Forrester Consulting analyzed many IBM customers who use these solutions and found developers, testers, and project managers achieve time savings of 30 to 50 percent on communication and collaboration activities alone.

Introduction

Waste reduction techniques make it possible for development teams to spend more time doing productive work, reducing cycle times and increasing the amount of code that can be delivered in a set timeframe.

Waste in software development projects often results from seven common features of software development:

- Waiting
- Handoffs and task switching
- Motion
- Extra processes
- Extra features
- Partially completed work
- Defects

This paper will explore how an IBM DevOps approach can help address these seven types of waste. IBM DevOps includes two core tools:

- IBM® Rational® solution for Collaborative Lifecycle Management (CLM): a platform for a code creation pipeline
- IBM UrbanCode: a platform for a code delivery pipeline

Measuring waste and value chains

Value chains help develop an understanding of how long tasks take and how much waste is present in any aspect of work. Often, the following value chains have significant waste:

- Moving new ideas into development
- Getting sign-off on requirements
- Searching for project information
- Setting up a test infrastructure
- Deploying an application into test or production
- Fixing defects

The value chain in figure 1 shows how a task—in this case, fixing a defect—passes through a number of individuals or teams until it gets completed. At each point the task is either waiting to be worked on (waste, shown in red), or actions are being taken to move it towards completion (effort, shown in blue).

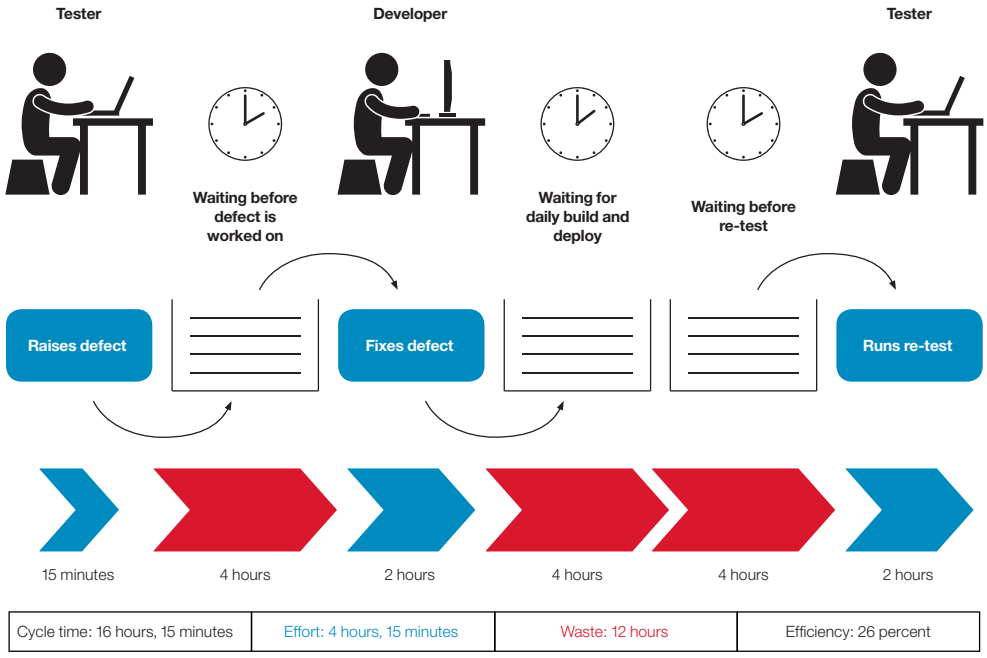

Figure 1. Example of a defect value chain

Over the length of the value chain, consider these measures:

- Cycle time = Effort + Waste
- Efficiency = Effort / Cycle time

Concentrating on reducing the average waste and effort across a value chain can make processes leaner and provide more value, more quickly. Reducing waste should be a higher priority than reducing effort, because it not only cuts the cycle time but also increases efficiency.

IBM DevOps capabilities can be configured to calculate the waste and effort at each stage of any value chain, and use these measurements to drive improvements.

The seven key causes of waste in software development

1. Waiting

Waiting for infrastructure

The time it takes for a team to provision machines and deploy software is often the cause of long periods of waiting.

Cloud technologies make it possible for teams to rapidly provision development, test and production infrastructure, potentially reducing waiting from days to minutes. Cloud strategies can also provide access to production-like environments earlier in the development and testing phases, which can drive defects out sooner and reduce deployment risk.

Waiting for applications to be deployed

Manual application deployment can create significant delays, especially when it depends on waiting for specialist team members to become available.

IBM DevOps capabilities can automate application deployment. One online retailer, for example, reduced the time for each deployment by 95 percent by using IBM DevOps.

Waiting for other teams

Teams often fail to prioritize work appropriately. Often there is no simple way for a team to understand which aspects of its work impact others.

IBM DevOps capabilities can raise the visibility of which tasks are being blocked by other tasks, helping teams self-organize to reduce waiting time.

Waiting for reviews to complete

Authors of requirements, design documents, code and software assets often have to wait for reviews to finish before they can move forward.

IBM DevOps capabilities can help by centrally assigning these artifacts to reviewers, who are then notified and reminded automatically. Moreover, the status of any review can be seen at any time.

2. Handoffs and task switching

A handoff occurs when one team member hands off a piece of work to another. Task switching is when an individual switches between different tasks.

Handoffs within project teams

Handoffs can introduce significant waste for several reasons:

- Re-work caused by the need to re-key information between tools or teams
- Wait time caused by team members being unaware that something is ready
- Lack of sufficient detail
- Outdated artifacts
- Changes that have not been successfully communicated to the team.

IBM DevOps capabilities help to reduce the handoff waste between development and operations by enabling teams to:

- Automate deployments
- Promote applications between environments
- Organize and manage large-scale releases.

With IBM DevOps capabilities, team members from any team can view information and follow changes in other teams, helping to save time. This transparency reduces the time wasted in handoffs, because team members can see the information they require without having to disturb others for status updates.

Handoffs between projects and later phases

As well as waste within project teams, there can also be waste between projects, departments, and third parties. Handoffs between teams can be wasteful:

- An application is passed from the development team to another team for deployment.
- An application is passed to a support team with insufficient documentation, so the support team must spend time figuring out how it works.
- An application needs to be upgraded, but the original requirements, design, tests, or even code can no longer be found.
- An application needs to meet requirements that have already been addressed when developing an earlier application, but the designs, tests and code of the earlier project can no longer be found.

A DevOps approach that includes software asset management can help solve these problems and reduce waste by storing useful groups of artifacts together with their context and relevant metadata. With all of the information in one place, it is easy to find development artifacts for systems in maintenance. Teams are able to quickly identify other projects that have solved similar problems and re-use their solutions to save time.

Task switching

Many personal productivity books emphasize that multitasking is not the fastest way to get jobs done.[1] Agile and Kanban development teams follow this tenet to minimize the number of things being worked on simultaneously.

Theoretically, therefore, the best approach is to avoid overcommitting teams. This will reduce their need to juggle too many different tasks, and enable them to undertake projects in sequence. In practice, however, this is not always possible.

If task switching is necessary, IBM DevOps capabilities can help. The waste caused by getting up to speed is minimized, because all artifacts (business processes, requirements, discussion forums, reviews, comments, tests, designs, code, plans, and even people) are linked and related. This arrangement provides context and shortens the time required to switch from one task to another.

IBM DevOps capabilities can allow developers to keep many code changes running at the same time, with minimal effort required to start or suspend work on any individual task. Instead of waiting until a particular artifact is available and ready to edit, the suspension function provides a way to minimize the waste from waiting and task switching.

3. Motion

Motion waste occurs when people spend time physically moving about—for example, when team members leave their desks to confer with business experts or attend unnecessary meetings.

IBM DevOps capabilities can reduce motion waste. By linking project artifacts and documentation, and enabling team members and stakeholders to view them easily (for example, via a web browser), there is less need for people to meet face-to-face to share information. This is particularly valuable for distributed development teams, where people work in different locations.

In interviews with IBM DevOps users, Forrester Consulting identified significant waste reduction resulting from teams being able to find the right information easily. One customer estimated that they saved around 70 percent of time previously spent on communication.

4. Extra processes

Extra processes are additional processes that do not deliver value to the customer—for example, producing documents that no one reads, manually updating plans and collecting metrics, organizing reviews that participants don't respond to, and other similar tasks.

To avoid waste from unnecessary processes, evaluate each process step in context using value chain analysis or a similar technique. Determine whether each step accelerates or delays the delivery of value for the customer. If it doesn't add value, consider whether that step can be eliminated or automated.

Removing extra processes, particularly through automation, can make it easier to implement smaller, more frequent releases, continuous integration, integration testing, test virtualization, and automated testing. These practices can also reduce waste, because they can reduce cycle times and enable more frequent feedback.

5. Extra features

Extra features are unnecessary features that take time and effort to design and develop, but deliver little value to project stakeholders.

IBM DevOps capabilities enable the business to collaborate around a set of high-level business ideas, prioritize them, manage the approval processes, and commit them to development. After the idea is in development, the business can view its progress across all the different teams involved.

This approach helps the business focus on the ideas that represent the best investment. It also minimizes the number of items that are developed, reducing both effort and waste. Meanwhile, the business retains the agility to respond to changing conditions.

6. Partially completed work

Waste from partially completed work is the amount of effort invested in features that are not yet implemented.

Using the waterfall development approach leads to a significant amount of partially completed work, because no working software exists until late in the development cycle. Yet even with an iterative approach, waste from partially completed work can still be significant.

If a team's work is spread across too many stories, testing might be left until very late in the iteration cycle. If defects are discovered when there is too little time to fix them, the iteration may end up not delivering any new functionality.

Therefore, within each iteration, the focus should be on finishing each story in turn, with the team working on as few stories as possible at any one time. This approach minimizes partially completed work and means that some completed code is delivered within each iteration or sprint.

Again, IBM DevOps capabilities can help. Agile planning taskboards can be used to help teams see where partially finished work exists, and close it off as quickly as possible. Collaborative development tools can also allow a greater number of developers to work on the same area of code.

7. Defects

Defects cause two kinds of waste: the waste in repairing the defect, and the waste caused by the software not working correctly. The IBM DevOps approach helps avoid this by improving the quality of requirements and reducing the time taken to find and fix defects—or in lean terms, improving the efficiency of the defect value chain.

The later defects are discovered, the more they cost to fix. To find defects earlier and reduce the waste associated with them, it becomes necessary to test earlier and earlier in the lifecycle. However, this can be difficult to achieve—particularly in integration testing—because of the complexity of setting up all the dependent applications.

DevOps techniques such as service virtualization can enable earlier integration testing by allowing applications or components to be tested against virtualized (i.e. simulated) versions of the services they depend on, instead of the real services themselves. This can be much quicker and less costly, while still ensuring that the application under test will integrate correctly.

Forrester Consulting reported on a large European bank that increased its project delivery capacity by 100 percent over three years (scaling from 40 to 80 projects completed annually) as a result of using IBM DevOps test virtualization and integration testing capabilities.

Conclusion

This paper explores various ways that IBM DevOps approaches can help teams become leaner and smarter. The DevOps capabilities described in this paper include those provided by IBM Rational Asset Manager, IBM Rational DOORS® Next Generation, IBM Rational Focal Point™, IBM Rational Quality Manager, IBM Rational Team Concert™ and IBM UrbanCode.

Using IBM DevOps capabilities helps to remove waste and automate manual processes, enabling software to be delivered more efficiently, and with reduced cycle times. As an example, Figure 2 illustrates how DevOps capabilities can optimize the defect value chain shown at the start of this paper.

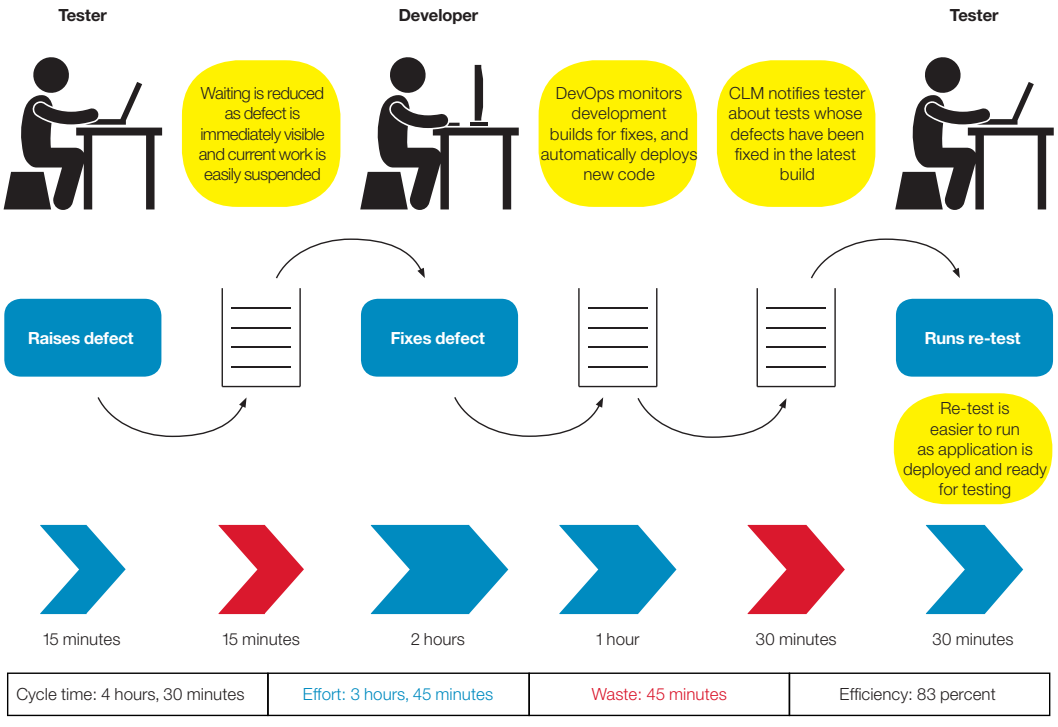

Figure 2. Example of a defect value chain optimized by IBM DevOps capabilities

In addition, IBM DevOps capabilities can help businesses steer their development processes more effectively. By increasing visibility of all aspects of a development project for all stakeholders, DevOps helps them to look at the end-to-end picture, to measure and monitor, and to continuously improve their software delivery.

For more information

For an in-depth discussion of how to shorten software delivery cycles by finding and reducing waste, please read this paper: ibm.co/1nnHZ73

To learn more about how IBM DevOps can help create leaner software development practices, please visit: **ibm.com**/devops

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
September 2014

[1] For example, Getting Things Done (David Allen, 2002) and Do It Tomorrow (Mark Forster, 2006)

Please Recycle

RAW14367-USEN-00