

# How Cisco IT Uses Agile Development with Distributed Teams and Complex Projects

Continuous delivery improves quality, developer productivity, and the employee experience.

EXECUTIVE SUMMARY
<p><b>CHALLENGE</b></p> <ul style="list-style-type: none"> <li>• Adopt continuous delivery for large software development projects</li> <li>• Build teams with people in different locations</li> </ul>
<p><b>SOLUTION</b></p> <ul style="list-style-type: none"> <li>• Adopted scaled agile framework (SAFe) for Subscription Billing Platform</li> <li>• Used agile scrum framework for Android WebEx client</li> </ul>
<p><b>RESULTS</b></p> <ul style="list-style-type: none"> <li>• Subscription Billing Platform: decreased defect rejected ratio by 16 percent and normalization defects by 40 percent</li> <li>• WebEx client: decreased QA defects by 25 percent and sped up review of new features</li> </ul>
<p><b>LESSONS LEARNED</b></p> <ul style="list-style-type: none"> <li>• Build teams with best members from any location</li> <li>• Invest in test automation tools</li> <li>• Consider eliminating the program level of SAFe when developing loosely integrated features</li> </ul>
<p><b>NEXT STEPS</b></p> <ul style="list-style-type: none"> <li>• Form scrum teams to develop enterprise-to-enterprise capabilities</li> <li>• Embed quality control in development scrum teams</li> </ul>

## Background

Cisco IT constantly looks for new ways to go faster, simplify, and become more agile. Our guiding principle is to digitize every process and thing that we can. Digital IT leads to a better customer experience, higher productivity, and business growth.

Part of our digital IT strategy is agile development. The goal is to replace periodic major releases with continuous delivery of new features.

Like most IT organizations, Cisco had adopted agile software development techniques to some degree. We started with small teams working on straightforward projects. Teams that were large, distributed, or worked on complex projects continued to use the waterfall development method.

The Cisco Cloud and Software IT (CSIT) organization wanted to adopt agile development for all projects, even the largest and most complex. “Our goals are to speed up releases, increase productivity, and improve quality,” says Ashish Pandey, technical lead for the CSIT team.

This case study describes how we used agile development for two large projects: the Cisco® Subscription Billing Platform (SBP) and an Android WebEx® client.

## Subscription Billing Platform

### Challenge

Cisco began offering annuity billing for subscription-based services in 2012. Subscription services include Cisco WebEx, Unified Communications software, Prime Home, and Service Exchange Platform (SxP).

The SBP is complex software as a service (SaaS), including internal portals, a services layer, and integration with Oracle Business and Revenue Management (BRM). Originally we formed different teams for design, build, test, and deploy. The build team didn't start until the design team completed its work, the test team didn't start until the build team had completed its work, and so on.

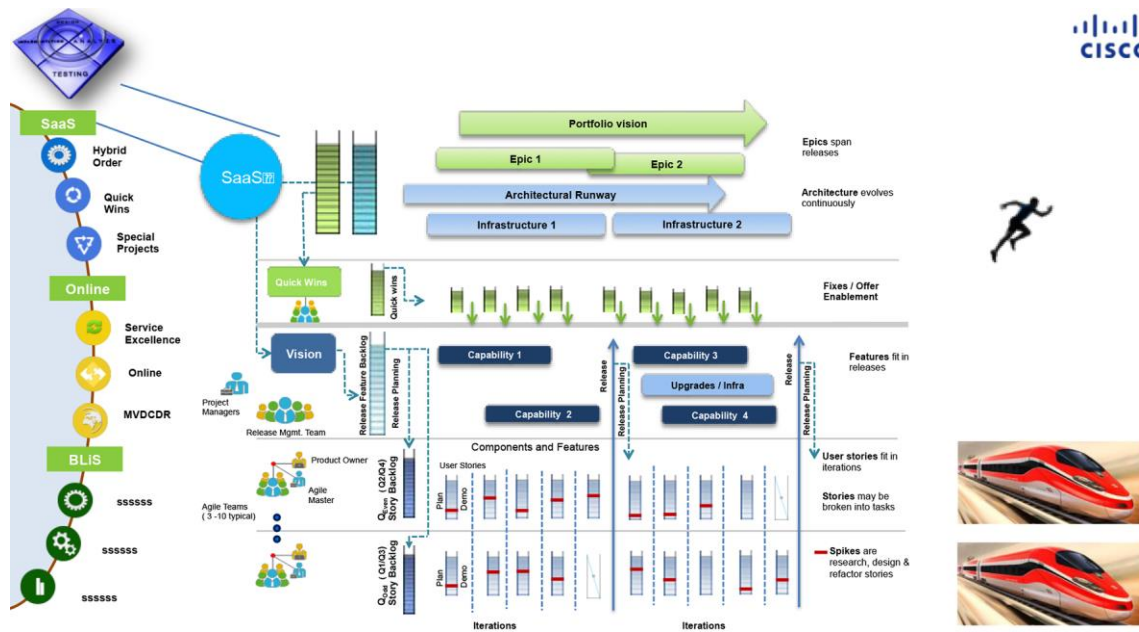
Over time we adopted agile practices for small functional enhancements, but larger enhancements continued to rely on the waterfall model. Maintaining separate tracks slowed us down, leading to:

- Long release cycles: more than three months
- Late closure on requirement documents
- Missed delivery dates
- Quality issues due to late integration cycles
- Long working hours and weekend work to make up for schedule slippage

## Solution

SBP development teams overcame these challenges by moving from waterfall development to the scaled agile framework, or SAFe (Figure 1).

**Figure 1.** Subscription Billing Platform Using the Scaled Agile Framework

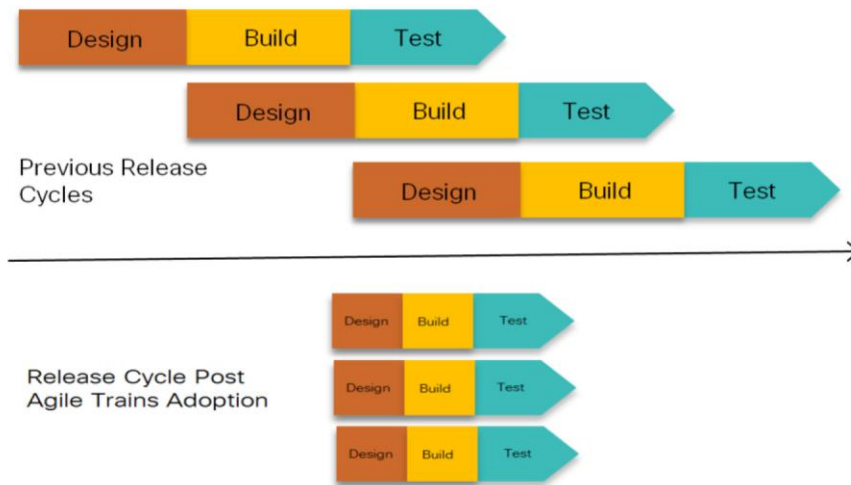


We began using agile release trains for SBP in 2015. An agile release train is a “team of teams.

We established three agile release trains: capabilities, defects and fixes, and projects. Capabilities and project teams are both core delivery teams. The difference is that project teams work on odd (development) and even (stable) releases, while capabilities teams work toward a bigger platform development.

The three agile release trains worked together to build and test small features within one SaaS component. They regularly delivered tested features to the system integration and testing team (Figure 2).

**Figure 2.** Continuous Delivery Shortened Release Cycles



During the release cycle, the delivery team met 15 minutes each day, entering action items on a Kanban board. A Kanban board is a work and workflow visualization tool that enables you to optimize the flow of your work. People working from home offices that day viewed the shared board on WebEx.

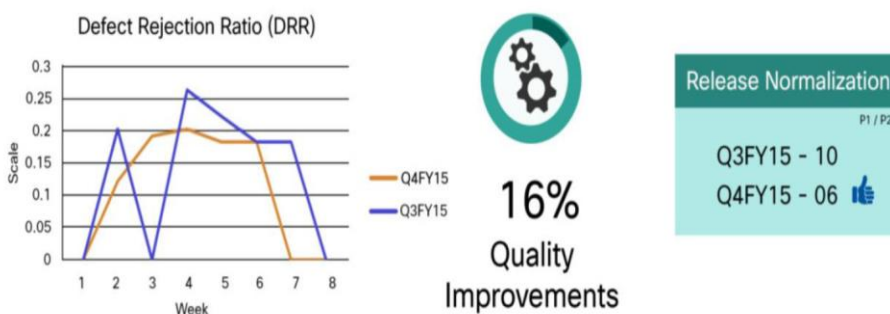
## Results

“Continuous delivery improved quality, increased productivity, and improved the employee experience,” says Pandey.

### Improved Quality

We delivered the new release of SBP on time, with 100 percent of planned capabilities (Figure 3). Compared to releases that relied on the waterfall method, the defect rejected ratio (DRR) dropped by 16 percent. Critical and major defects decreased by 40 percent.

**Figure 3.** Continuous Delivery Improved Defect Removal Efficiency and Decreased Major Defects



We attribute the quality improvement to:

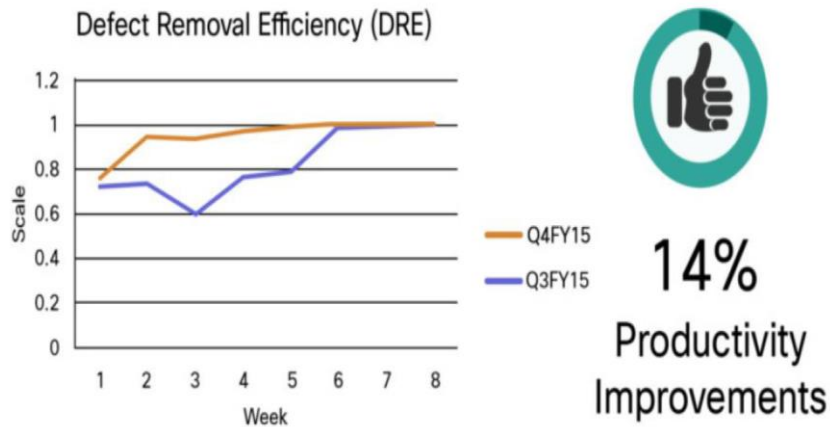
- Improving team collaboration and focus
- Enabling all team members to see current project status, promoting accountability
- Helping the three teams see beyond their own track to complete application and customer need
- Enabling teams to manage themselves

## Increased Productivity

Shifting to continuous delivery increased defect removal efficiency (DRE) by 14 percent (Figure 4). Reasons for the improvement include:

- Improving collaboration among developer teams in the US, China, and India
- Helping team members identify opportunities for improvement during daily scrum / triage calls

**Figure 4.** Continuous Delivery Increased Productivity



## Improved Employee Experience

Continuous delivery eliminated the need for after-hours and weekend work for analysis, design, and implementation. Employees appreciated having just one daily scrum / triage call instead of multiple calls. "Agile development also improved work satisfaction," Pandey says. "The release planning event helped team members understand how their work fit into the big picture for Cisco and our customers."

## WebEx App for Samsung Tablets

### Challenge

Beginning in January 2014, the WebEx Meetings app came pre-installed on Android tablets. Now people who purchase a tablet receive six months free access to WebEx Meeting Center. To prepare the app on time, CSIT developers had to work quickly. Requirements changed frequently, and developers had to expedite changes requested by the business unit.

### Solution

The CSIT team implemented an agile scrum framework with three sprints:

- Sprint 1 (3 weeks): Delivered to US and Canada
- Sprint 2 (3 weeks): Delivered to four European countries
- Sprint 3 (5 weeks): Delivered to seven Asian sites

During the planning phase, Cisco IT and other tracks worked together to gather requirements. They evaluated the readiness of environments, partners, and engineering and marketing teams. The development team used extreme programming practices, including test-driven development (TDD). With TDD, developers first write an automated test case for a new function. Next they produce the least amount of code needed to pass the test. Then they refine the code to make it simpler and easier to maintain.

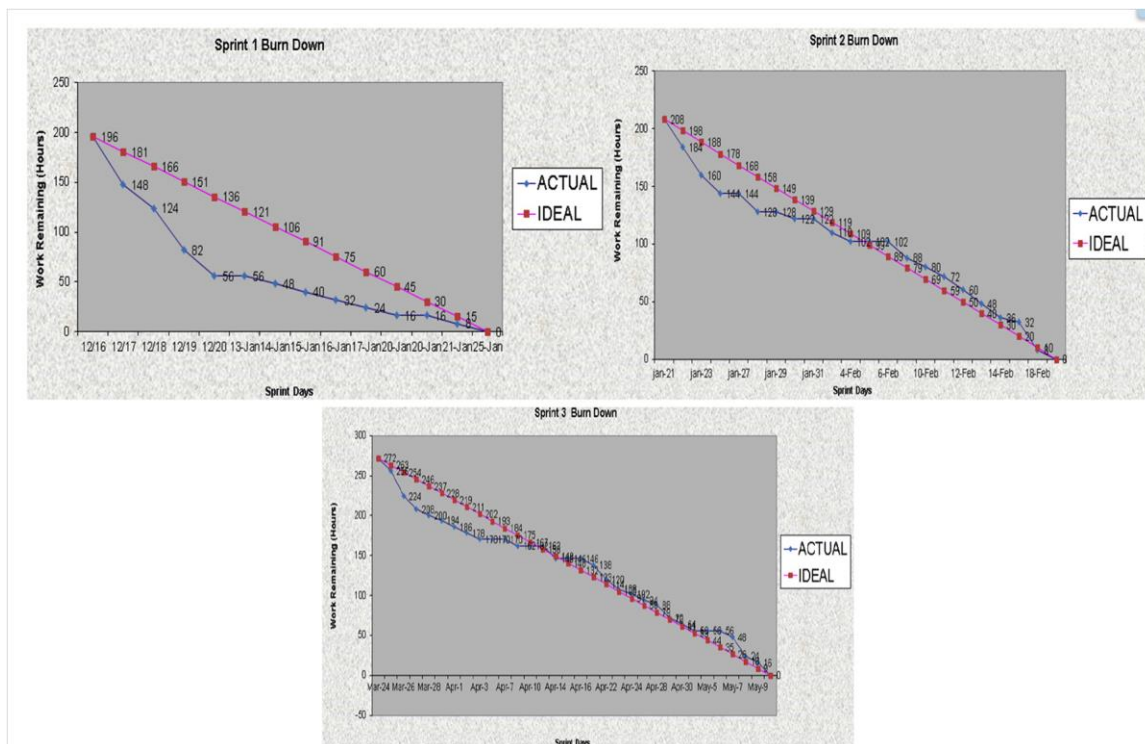
## Results

More than 35 million people purchased Samsung tablets preloaded with WebEx Meetings, increasing brand awareness.

We reduced quality assurance (QA) defects by 25 percent by using agile scrum development. In addition, the business group reviewed new features earlier in the cycle than they did with previous releases because developers checked code into a repository several times a day—the basis for continuous integration.

With each sprint we became more efficient, as shown in the burn-down charts in Figure 5. The charts plot outstanding work (vertical axis) against time (horizontal axis).

**Figure 5.** Each Sprint Was More Efficient than the Last



## Lessons Learned

- Carefully choose team members. People in CSIT initially were skeptical that agile release trains could work because the ideal team requires skills not available in one location. We built the teams with members in different countries. Most team members are dedicated to their team full time, while a few split their time between multiple teams.

- If possible, assemble teams with members in the same country. We found that developers in different global locations tended to operate as separate teams. The reasons are time zone differences, reporting structure, and cultural differences.
- Realize that incremental testing requires the right kinds of tools. We could not have conducted regression testing every two weeks without test automation tools. We still conduct some manual testing.
- It's possible to eliminate or scale back certain elements of SAFe when working on unintegrated or loosely integrated products, features, or components. Consider eliminating the program level.

## Next Steps

Our next steps include:

- Creating future teams to deliver enterprise-to-enterprise capabilities
- Integrating QA into development teams
- Scaling the scrum teams
- Accelerating delivery and feedback cycles.

## For More Information

To read How Cisco IT Developed a Self-Service Model for Build and Deploy, visit

<http://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco-on-cisco/cs-boit-02032016-self-service-model-for-build-and-deploy.pdf>

To read additional Cisco IT case studies about a variety of business solutions, visit Cisco on Cisco: Inside Cisco IT

<http://www.cisco.com/go/ciscoit>

## Note

This publication describes how Cisco has benefited from the deployment of its own products. Many factors may have contributed to the results and benefits described. Cisco does not guarantee comparable results elsewhere.

CISCO PROVIDES THIS PUBLICATION AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow disclaimer of express or implied warranties; therefore, this disclaimer may not apply to you.



Americas Headquarters  
Cisco Systems, Inc.  
San Jose, CA

Asia Pacific Headquarters  
Cisco Systems (USA) Pte. Ltd.  
Singapore

Europe Headquarters  
Cisco Systems International BV Amsterdam,  
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)