

# Continuous Delivery Software Release Methodology

## CONTENTS

Itron Software Release Methodology.....	2	Change Control .....	4
What is Continuous Delivery? .....	2	Automated Deployment.....	4
Continuous Delivery Benefits.....	2	Development .....	4
Agile Software Development Methodology.....	3	Test.....	4
Microsoft Azure DevOps Services .....	3	Staging.....	4
Test-driven Development (TDD).....	3	Production 1 .....	4
Incremental Iterative Changes .....	3	Production 2.....	4
Continuous Integration.....	3	Customer Testing.....	4
Required Peer Code Review.....	3	Service Level Agreements .....	4
		Conclusion.....	4

## ITRON SOFTWARE RELEASE METHODOLOGY

To provide the best value for our customers at a competitive cost and increase the velocity of innovation, Itron Software and Services continues to move towards a Software as a Service (SaaS), **Continuous Delivery** methodology. Instead of customers purchasing software and incurring the cost of the infrastructure and personnel to run it, they purchase a monthly subscription to the services and applications they need, as they need it.

Instead of worrying about upgrades and versions, users have confidence that they are running the latest and greatest versions, and they know they have the support of development teams who are focused on ensuring their system is running at optimal performance.

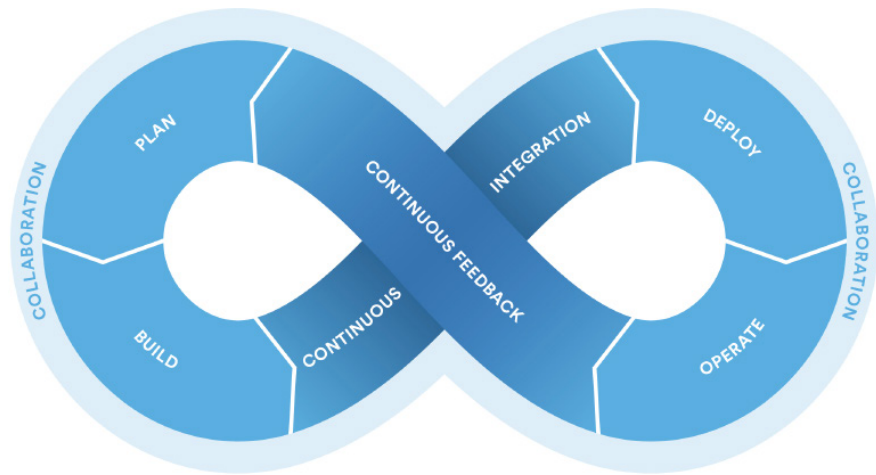
Instead of waiting months for new features that might miss the mark, Itron speeds up innovation and delivers new features into users' hands early and often. In doing so, we are able to collect valuable user feedback as we iterate on new functionality to ensure it provides promised value, with high quality. Itron Analytics successfully uses this model, releasing to production for all customer systems, every three weeks.

Every release adheres to strict metrics before it is deployed to production. These metrics include:

- » 98% of all automated and manual tests for regressions and new features must pass
- » 0 Priority 1 or Priority 2 defects are open
- » Approval by product management, development management and managed services

## WHAT IS CONTINUOUS DELIVERY?

Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time. It aims to build, test and release software faster and more frequently. The approach reduces the cost, time and risk of delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment process is important for continuous delivery.



DevOps Continuous Delivery Process

Continuous delivery is enabled through the deployment pipeline. The purpose of the deployment pipeline has three components: visibility, feedback and continually deploy.

**Visibility** – All aspects of the delivery system including building, deploying, testing and releasing, are visible to every member of the team to promote collaboration.

**Feedback** – Team members learn of problems as soon as possible when they occur so that they can fix them as quickly as possible.

**Continually deploy** – Through a fully automated process, you can deploy and release any version of the software to any environment.

## CONTINUOUS DELIVERY BENEFITS

**Accelerated Time to Market** – CD lets organizations deliver the business value inherent in new software releases to customers more quickly. This capability helps us stay ahead of the competition.

**Building the Right Product** – Frequent releases let the development teams obtain user feedback more quickly. This allows them work on only useful features. If they find that a feature isn't useful, they spend no further effort on it. This helps them build the right product.

**Improved Productivity and Efficiency** – Considerable time savings for developers, testers, operations engineers and so on, through automation.

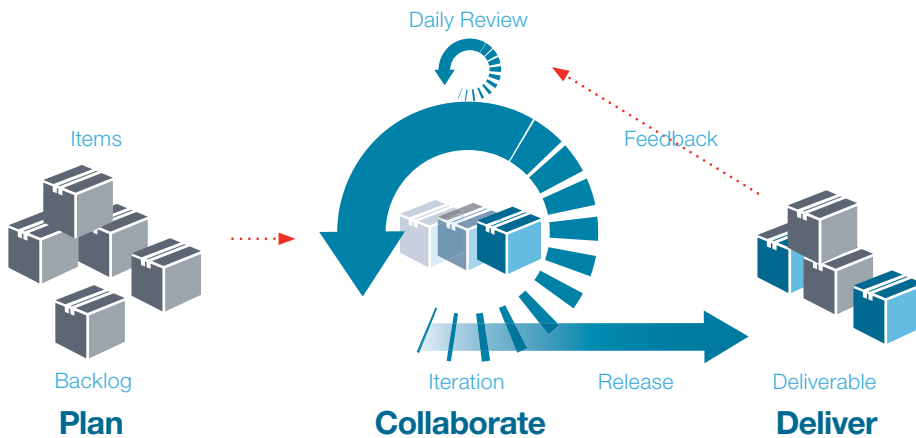
**Reliable Releases** – The risks associated with a release are significantly decreased, and the release process is more reliable. With CD, the deployment process and scripts are tested repeatedly before deploying to production. So, most errors in the deployment process and scripts have already been discovered. When the release frequency increases, the number of code changes in each release decreases. As a result, finding and fixing problems that occur is easier, and the length of time for which the error causes an impact is reduced.

**Improved Product Quality** – The number of open bugs and production incidents is decreased significantly.

**Improved Customer Satisfaction** – A higher level of customer satisfaction is achieved.

## AGILE SOFTWARE DEVELOPMENT METHODOLOGY

Agile software development is a set of principles for software development in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, prompt delivery and continuous improvement, and it encourages rapid and flexible response to change.



Agile Project Management: Iteration

Itron software development teams use Agile methodology to plan and deliver high-quality software in a short continuous delivery model, with focus on early user feedback and continuous improvement. Each team owns the services or applications they provide from planning, design, implementation, deployment, documentation and support—working closely with stakeholders such as sales, marketing and support services to external users.

Itron uses specific tools and processes to ensure high quality, innovative software is released to production every three weeks or even less to add more value to its customers using Agile methodology to support continuous deployment.

## MICROSOFT AZURE DEVOPS SERVICES

Azure DevOps Services (ADS) is Microsoft's modern Application Lifecycle Management system that provides teams with tools to plan, collaborate and deliver software. Itron uses ADS for:

- » Requirements planning and tracking
- » Release planning
- » Work scheduling and tracking
- » Test case development, execution and traceability

- » Defect and issue tracking
- » Deployment automation
- » Cross-team collaboration
- » Change control management
- » Peer code reviews, or "Pull request" review
- » Software programming

The tools ADS provides enables Itron to successfully implement an Agile methodology and continuous delivery, speeding up the velocity of innovation while improving quality.

## TEST-DRIVEN DEVELOPMENT (TDD)

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: Requirements are turned into very specific test cases, then the software is improved to pass only the new tests. Alternative software development processes can allow software to be added that is not proven to meet requirements.

Itron software engineers take ownership for collaborating with the product management and test resources on their team to first write automated unit tests that are run with every build deployed to any environment. All unit test failures are expected to be resolved immediately before that code can be merged into the release branch, stopping it from ever getting out in the first place.

A main component of the defect resolution process is that unit tests and test cases are added to regression testing each time a defect is identified in production and resolved.

## INCREMENTAL ITERATIVE CHANGES

To provide more visibility and feedback on new features, with CD, Itron deploys only very small changes, incrementally over time. Because only 1-10% of a new feature is put into production at a time, any issues that are found in production are resolved quickly and have limited impact to end users. With other software delivery models, huge software changes create the need for significant user training and advance upgrade planning to ensure changes are accepted. With smaller, iterative changes, users can more easily adapt to changes and provide early feedback to the team; reducing or eliminating the pain of upgrades.

## CONTINUOUS INTEGRATION

Continuous integration is a software development practice where developers regularly merge their code changes into a central repository, and automated builds and tests are executed. These automated tests that are run with every build identify and inform developers of any defects created by their code. Developers can resolve those defects early in the development process.

## REQUIRED PEER CODE REVIEW

All code changes that are integrated into the release branch require peer approval. A developer's code changes must be reviewed and approved by another developer on the team before it can be integrated. Code changes require team collaboration on the rework to ensure defects are prevented even before they are ever integrated and caught by a test.

## CHANGE CONTROL

Developers use change control tools in ADS to ensure code changes happen in a structured way. Code changes are tracked by the person who made the change, the work item associated with the change, the reason for the change and when to assist with continuous improvement and prevent defects.

## AUTOMATED DEPLOYMENT

Itron utilizes automated deployment mechanisms and scripting. When applied within the development, testing, staging and production environments that mimic production environments, we ensure solid, reproducible deployments throughout the entire pipeline. Itron also uses a concept known as “canary releases” to reduce the impact of issues found during deployment. If a canary release fails deployment at any step, other environments continue working on their current build until the issue is resolved.



Deployment Pipeline Diagram

### Development

The Development environment is where software engineers implement new automated unit tests and functionality within a given sprint. Code changes are peer reviewed prior to being allowed to integrate into the codeline and automated tests run on all new builds.

### Test

The second environment in the deployment pipeline is the Test environment. It is updated automatically with new builds multiple times per day within a given sprint. Unit tests are run with every deployment where failures are fixed immediately. Developers and testers also execute additional automated and manual testing of new functionality and targeted regression test cases beyond existing unit tests. Bugs identified are tracked through resolution and prioritized. Priority 1 & 2 bugs are required to be resolved and

verified; and 98% of all test cases must pass before deployment can proceed to any production environment in the pipeline.

### Staging

In the Staging environment, builds are deployed upon approval once the development team, in collaboration with product management, agree the release is stable and near “production ready”. This environment is where Itron internal stakeholders, such as Itron Customer Support and Product Management can start previewing new functionality and report any issues found. Bugs reported are prioritized and potentially required to be resolved prior to release to production. This is also the environment used to demonstrate new features to internal or external Itron stakeholders prior to production release for initial stakeholder feedback.

### Production 1

After the final release candidate build has been successfully deployed to the Staging environment with 0 priority bugs and 98% pass on all test cases, then approval is given to deploy the release to the first production environment. Any deployment failures are resolved by the team as soon as possible 24 hours per day, 7 days per week.

### Production 2

After the release build has been verified as successfully deployed to Production 1, then the build is deployed to all remaining production environments. Any deployment failures are resolved by the team as soon as possible 24 hours per day, 7 days per week. Deployment to production systems is done with zero down time when possible, but if not, always occurs during off-work hours (6PM to 5AM, weekdays) to minimize any maintenance down time of the applications.

## CUSTOMER TESTING

Because new releases can be deployed at any time, customers are not able to test before released to production. However, upon request in certain circumstances, customers can review any test cases and even request that specific test cases are included in automated tests to ensure adequate test coverage of customer-specific scenarios.

The changes introduced into releases are very small. Therefore, if any defects are identified by a user, they can be reported to the team, where it becomes the development team’s top priority to investigate, fix and add needed automated unit tests to the regression testing to ensure it remains resolved.

## SERVICE LEVEL AGREEMENTS

Service Level Agreements (SLA) are the customer’s primary measure of product quality. If an Itron service or application does not adhere to their SLAs, customers can demand recourse, including financial compensation. We take very seriously the trust that our customers have in us to deliver high quality software and services. For this reason, we believe we should compensate our customers if we don’t meet their expectations.

An example SLA is, “Service is available 99% of the time, excluding a specified maintenance window of Sundays from 9PM-12AM pacific time.”

## CONCLUSION

Continuous Delivery is a proven methodology that increases the speed of innovation and the quality and reliability of software applications and services in an efficient, cost-effective manner. Through the combination of SLAs, Agile Methodology, Automated Testing and Deployment, and Test-driven Development, Itron believes we deliver new value that’s reliable and trouble-free for our customers.



Join us in creating a more **resourceful world**.  
To learn more visit **itron.com**

### CORPORATE HQ

2111 North Molter Road  
Liberty Lake, WA 99019 USA

**Phone:** 1.800.635.5461

**Fax:** 1.509.891.3355