Inductive Automation Software Development Lifecycle (SDLC)

inductive automation

(800) 266-7798

inductiveautomation.com

Overview

Inductive Automation is dedicated to providing our customers with the highest quality products and support. This document outlines information on Ignition versioning, release schedule, security/vulnerability fixes, and an overview of our Software Development Lifecycle (SDLC) including Quality Assurance (QA) processes. It includes company cybersecurity practices supporting secure development and Ignition resources supporting best practices for customer environments.

Software Updates

Ignition is an Industrial Application Platform that is typically downloaded and installed from a single file from the Inductive Automation website. The user manual covers verification that Ignition software is genuine. The Ignition platform provides a broad base capability, while available modules each provide additional functionality. Updates, including new features, bug fixes, and security patches, are all provided together in a given version, not as separate patch files. Inductive Automation sees this as the most viable way to keep Ignition up to date from the perspective of all stakeholders. Inductive Automation places top priority on remediating stability, performance, interoperability, and security bugs.

Ignition software undergoes constant improvement in features, quality, reliability, and security. All changes are tracked by tickets, which represent code changes in the form of features or fixes. To keep track of each iteration of our software and coordinate upgrades, version numbers are assigned to both the Ignition platform, as well as to each module. Version numbers are always three separate numbers separated by periods; for example, 8.1.31. The first number in this triplet is called the Major Version number. The second number is called the Platform Coordination number. The third number is called the Minor Version number. Minor version updates are always free. Changes are listed in the "new in this version" section of the online user manual and release notes associated with each download.

Ignition Software Evaluation

Inductive Automation has provided free public downloads of all production software builds since Ignition 7.3.0 (10/5/2011). The most current "Nightly" and "Release Candidates" builds are available for public evaluation. All software is fully functional, with a resettable two hour runtime trial, allowing for a thorough evaluation prior to purchase. https://inductiveautomation.com/downloads/

Release Schedule

Ignition follows a "release train" style of scheduled releases on a five-week cycle. This popular concept is quite simple: releases (e.g. 8.1.1, 8.1.2, etc.) go out on a regularly timed schedule with the fixes and improvements that are ready, just as a train leaves the station on a predetermined schedule with the passengers onboard. In software terms, this means each fix or improvement is developed and tested in isolation. When each change is finished and verified, it can be merged into the main branch of the software. The big advantage of this style is seen in the testing cycle. Since testing on all changes was already completed in isolation, the final release testing time is dramatically reduced.

Nightly and Release Candidate Builds

Ignition builds are made publicly available each night to include the available changes from that day. This is made possible by the automated build process and incremental testing strategy.

This nightly schedule replaces the concept of "Betas" entirely. The nightly upload is a great way for customers who need to try out a fix or feature in their environment to do so with the absolute minimum delay. The Early Access Forum channel offers nightly change logs and a space for discussion. Nightly builds provide such great testing environments because of the isolated change approvals that are completed with the release train strategy. However, it is still important to wait for official version releases before using a new version in a production environment.

Two weeks prior to a production release, we will lock (branch) that release and call it a Release Candidate, which has its own checklist. The nightly builds continue that very night as the nightly of the next version, as the team works on regression testing and analysis of the release candidate. Some customers use this version in their development or testing environments.

Release Checklists

Release checklists ensure that all actions from multiple teams, including comprehensive integration testing, were conducted prior to each production release and release candidate. The release management team, including senior developers and QA engineers attest that all required steps were properly completed on digital "release checklists" for scheduled and out of band releases.

Long-Term Support Program

Long-Term Support Version means an Ignition platform version announced by Inductive Automation to be given Long-Term Support (LTS). By convention, odd numbered major versions (e.g 7.9, 8.1, 8.3) are LTS. Platform versions that are noted as LTS will be supported for a period of five years from the date of their original release or two years past the date of a subsequent LTS version release, whichever is longer. Full details are provided under the support policy. https://inductiveautomation.com/support/policy/

Incident Management and Coordinated Vulnerability Disclosure

Vulnerability Reporting

Inductive Automation accepts and appreciates Ignition related vulnerability reporting at: security@inductiveautomation.com.

Inductive Automation PGP key: https://files.inductiveautomation.com/security/pgp-public-key.asc

Vulnerability Notification

Customers are notified by subscribing to "Trust Center Updates" on the Security and Trust Portal (https://security.inductiveautomation.com). Notifications are also sent to Ignition email lists and posted online as appropriate. If possible, mitigating recommendations are also provided.

Inductive Automation works with security researchers, the National Institute of Standards and Technology (NIST) and Mitre to disclose Ignition vulnerabilities to the National Vulnerability Database (NVD).

Incident Management

Incident management deals with unexpected events. It is covered by a cross divisional organizational program with playbooks for individual incidents. Coordinated vulnerability disclosure, which includes managing Ignition vulnerabilities throughout their lifecycle, is managed by playbook under the incident management program.

Priority Ignition Tickets

Critical issues or vulnerabilities are tracked as priority 1 or “Red” tickets. These typically either hold an upcoming release for remediation or expedite a new release, based on risk assessment and timing during the 5 week release cycle. Severity for Ignition related security issues is based on CVSS v3 score, with the assessment taking public knowledge, exploits in the wild, and intended configuration into account.

Third Party Testing

Inductive Automation hires annual penetration tests or external security reviews for the company environment. IA hires regular application penetration tests for each major Ignition version. Ignition 8.0 was tested in December 2019 (InGuardians) and Ignition 8.1 was tested in January 2022 (Coalfire Labs). A bug bounty is in progress for 2024. Most recently, the company underwent an external penetration test in Sept 2023 and an assumed breach (internal) penetration test in October 2022 (TrustedSEC). Inductive Automation scored well and quickly remediated significant risks. Inductive Automation also voluntarily participates in Pwn2own Miami in the targeted Industrial Control System (ICS) category annually as available since 2020.

Software Development Lifecycle

The Inductive Automation Software Development Lifecycle (SDLC) is certified under ISASecure and IEC 62443-4-1, which covers security management, requirements, design, implementation, verification and validation testing, defect and update management, and guidelines. The Inductive Automation Software Development Lifecycle (SDLC) is a coordinated effort based around people, processes, and technology that covers a single common code base. Agile software development processes are based on an iterative and incremental development methodology for managing product development and synchronized across the organization. Processes are informed by industry best practices such as the NIST Secure Software Development Framework (SSDF). Ignition is released under the Ignition, Ignition Edge, Ignition Maker Edition, and Ignition Cloud Edition product lines with builds supporting Windows, Linux, Mac, Docker containers, and more.

Ignition development is governed by a Product Council, overseeing five Product Groups consisting of approximately 20 Product Areas for a total of about 60 systems. Cross functional teams are assembled around complementary product areas and consist of 8-14 members including software developers, quality assurance engineers, and UI/UX design group members.

All work is strictly done via change control processes based around a change management ticket based system within a larger project management suite. Ignition is developed under short regular DevSecOps “sprint” cycles following accepted agile development methodologies.

Inductive Automation utilizes an independent QA department with a target of a one to one parity between Software Developers and QA Engineers. QA is involved in all stages of the SDLC from planning, design, implementation, testing, and ongoing support and has representation on all development teams. QA conducts a range of manual and automated testing modalities in support of stability, security, performance, and quality across the spectrum of Ignition functionality.

Single Software Build Process and Pipeline

Ignition, including all modules and target architectures, is only ever built on one well protected and monitored Continuous Integration, Continuous Delivery (CI/CD) pipeline. Multiple CI tools regularly perform static code analysis, inspecting code branches to detect bugs and security vulnerabilities, as well as measure the technical quality in terms of potential defects, vulnerabilities and maintenance risk. A separate tool provides software composition analysis, focusing on third party libraries through continuous vulnerability scans and assessments for security, licenses, and operational risks of dependencies. The tool generates an SPDX 2.2 (JSON) "Software Bill of Materials" (SBOM.txt) and third party license and copyright notices (notice.txt), that are bundled with each release (manual). Vulnerabilities are prioritized and managed with the project management system and release checklist.

Technical controls enforce strict adherence to a standard "Fork and Pull Request workflow", ensuring that no direct changes are ever made to the main repository and forming a core part of the quality process. Every single code change must be reviewed and is subject to significant checks. Each pull request (PR), which is a potential software change that must reference one or more tickets, triggers a review and testing process. The CI/CD pipeline creates a full Ignition build on a containerized image with the potential change. The build ensures the code changes properly compile, passes strict automatic code-style guidelines, and that all unit and module tests pass. Reliable human review with two-person integrity requires a qualified developer to digitally sign off on a code review for all changes. The build pipeline is responsible for code signing. Certificates are protected using industry accepted practices using external vaults. Automatic validation testing starts with automatic integration testing through multiple tools and frameworks on the candidate build.

The CI/CD pipeline continually builds, verifies, and tests software. Each night the pipeline uploads new versions to the website and performs steps to assure users the software is genuine. A complete change log is available online for each version of the software.

Security Activities

Security related activities and tickets are processed by security qualified individuals throughout the SDLC and across the organization. This includes threat modeling, attack surface analysis, fuzz testing, secure design reviews, and others. Secure coding standards and design practices are incorporated in training, implementation, code review, and other processes. Static code analysis rules are regularly reviewed by the security team and aligned with secure coding standards (e.g. OWASP, SANS top 25, cwe, etc). Input comes throughout the organization, such as via customer support or externally from security researchers. Security process outputs include other tickets to fix weaknesses, development priorities, risk information for decision makers, and documentation. Additionally, our development team works with our customer-facing engineering teams to produce technical documentation, such as the Ignition Security Hardening Guide, which describes recommended deployment patterns for maximum security.

Security Governance

Security governance is achieved through a cross divisional Security Advisory Group that is responsible for enterprise and product risk management, advising executive leadership, overseeing security policy and strategy. The Security Advisory Group is also involved with other areas, such as Incident Management and change control oversight.

Quality Assuance

QA oversees and performs a variety of automated and manual testing on quality, stability, performance, security, compatibility, and other important aspects throughout the SDLC. QA engineers are embedded in every agile development team and participate in all phases from requirement analysis, planning, through execution, release and maintenance.

QA runs multiple environments with a variety of hardware and software, maintains automated tests that run for every build, and performs tailored manual test plans for each production release version that focus on the significant changes by risk.

Verification testing, based on "Definition of Done" as a mandatory step for every ticket in the project management platform, occurs early and often. The CI/CD pipeline will not include code associated with a ticket in a build prior to QA attestation. QA also performs numerous validation and other tests including: load/stress testing, functional, integration, regression, security, and operational testing in dedicated environments with a variety of tools.

Validation testing is strictly enforced by the build process (CI/CD pipeline) for every code commit. Reliable human review with two-person integrity is required (SDLC). All module/unit tests must pass. Integration tests automatically run, and must pass, on a containerized instance with the candidate build.

Once the developers handle a ticket, it is handed off to the QA teams for testing. Each ticket is tested with a variety of test cases. All tests are documented. A ticket cannot be completed until passed through QA. This ensures quality through the development life cycle. The QA team also runs a series of automated tests and manual tests before a version is released to the public. The QA team has an automated test environment that includes testing the UI. A series of manual tests are run for items that can’t be tested automatically. This includes installing/upgrading on different OSs and hardware, special corner cases, etc. When a new issue is identified by tech support, it is added to the QA’s test environment.

Change Control

Change control is strictly controlled at all levels through synchronized processes and tools with technical controls. Code changes are only done through a ticketing system. Tickets have a state flow, where processes ensure that multiple stakeholders attest to conducting required activities. Significant work must occur before any code is written. The Build Pipeline recognizes ticket state before processing any changes. Automation is employed throughout the workflow to synchronize decision making and changes sequenced by multiple teams and stakeholders.

Configuration Management

Configuration Management is achieved by traceability linkage between actions in all work products, such as multiple types of tickets, requirements, specifications, UI/UX mockups, source code repository commits, build activity, various automated testing, customer trouble tickets, QA activities, Release Checklists, documentation, public facing content, and other key products. Individual systems are all tied to a common Identity Provider supporting multi-factor authentication (MFA), device health checks, and more. Each system performs audit logs and products are versioned, enabling viewing or restoration of previous versions and visibility of changes. Automation is utilized throughout these steps to tie everything together. For example a custom “DevBot” links in source code pull requests, automated testing, two-person reliable human code review notes, and other content within the ticketing system and collaboration platform. It also provides custom notifications with links to the business messaging app for group notification, and is coordinated with Build Pipeline automation.

Root Cause Analysis (RCA) / Post-mortems

Regression post mortem events are triggered when a product problem passes process control gates. Senior leadership and stakeholders conduct a blameless post mortem in order to understand what failed and how the process can be improved.

The root cause analysis consists of first identifying the contributing factors that led to the regression. These are the situations, circumstances or conditions that increased the likelihood of the event. Next, these factors can be analyzed to determine the underlying process or systemic issue that led to the regression. Corrective actions focus on addressing these root causes.

While root cause analysis often focuses on the refinement of processes or introduction of new ones, these meetings also provide an educational opportunity for all team members. By thoroughly examining issues as a team, we learn from our mistakes and share domain knowledge.

Protecting the development environment

A dedicated cybersecurity division, separate from IT, monitors and scans the development environment. Inductive Automation adheres to the CIS 8 security framework. Access is protected by a zero trust strategy, heavy segmentation, Multi-factor authentication (MFA) on nearly everything, with a fully integrated Security Information and Event Management (SIEM) and endpoint security solution that includes application Whitelisting, endpoint detection and response, Next Generation (antivirus, firewall, web application firewall), and DNS/Web filtering based on threat feeds. The system is maintained in-house and monitored 24x7 by security operations center (SOC) analysts. Inductive Automation hosts numerous Ignition systems both internally and publicly that cybersecurity monitors and scans with multiple tools.

Ignition Related Resources

Security Hardening Guide

The Ignition Security Hardening Guide provides best practices for securing Ignition gateways and the associated architecture.
https://www.inductiveautomation.com/resources/article/ignition-security-hardening-guide

Training and Documentation

Inductive Automation offers in-person and virtual training courses. https://inductiveautomation.com/training/

Inductive University (IU) offers free, self-paced Ignition video training content. https://inductiveuniversity.com/

Ignition Documentation is available online. https://docs.inductiveautomation.com/

Online Demo

Inductive Automation offers a live online demo featuring a Perspective project hosted with a major Cloud Service Provider (CSP) in three geographic regions (US, Europe, and Australia), spanning multiple availability zones. The Online Demo runs production versions of Ignition behind CSP application load balancers and distributed networking technologies. It provides an example of an n-tiered architecture where devices, data sources, and users are segmented. https://demo.ia.io

Ignition Industry Specific Resources and Guides

Inductive Automation offers industry and technology specific guides that provide planning, implementation, and operating recommendations. For example, some regulated industries require asset owners to validate hardware and software within their processes. Inductive Automation believes that the best way to succeed in regulated industries is to understand and adhere to laws, regulation, guidance, adhere to industry best practices, and consider modern IT and OT technologies. Design, customization, integration, and implementation should be done based on customer requirements, risk assessment, corporate policy, and backed by subject matter expertise.

21 CFR part 11 and Pharmaceutical Best Practices With Ignition

NERC CIP Best Practices with Inductive Automation and Ignition