Specific. Actionable.
Threat Intelligence
Powered by Deception.

Product
Quality Assurance

COUNTERCRAFT PROPRIETARY INFORMATION. CONFIDENTIAL.

Table of Contents

Product Quality Guidelines

Introduction

At CounterCraft we use a central hub for quality tracking across the entire development lifecycle, while ensuring quality practices are embedded across all operational areas. The Quality area focuses on proactive and reactive quality initiatives. It encompasses quality planning, testing strategy, measurement, and continuous improvement. This area ensures that quality is built into the product from requirements through deployment, not just verified at the end.

OWNER: Head of Quality Assurance

STAKEHOLDERS: Development, Product Management, Customer Success, Operations, End-users, Design

ACTIVITIES:

- **Quality Measurement**: Collecting and analyzing quality metrics
- **Planning**: Establish quality strategies and standards
- **Release Validation**: Performing release testing and quality assurance
- **Process Improvement**: Enhancing quality-related processes for e2e, CD, CI

INPUTS:

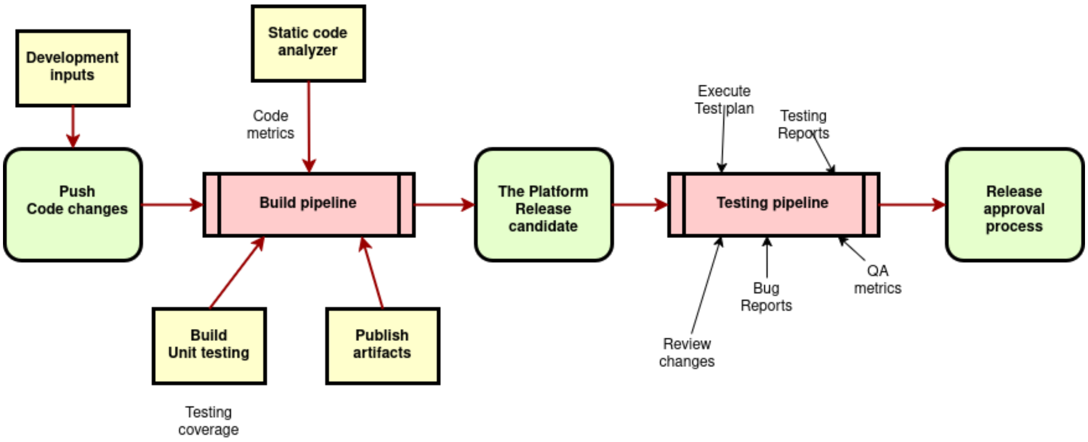
- Release Candidates
- Product Requirements
- User Stories
- Performance metrics
- User feedback

OUTPUTS:

- Observability and issue reports
- Test reports
- Quality metrics
- Manual Review
- Automated test and metrics review
- Release approval documentation

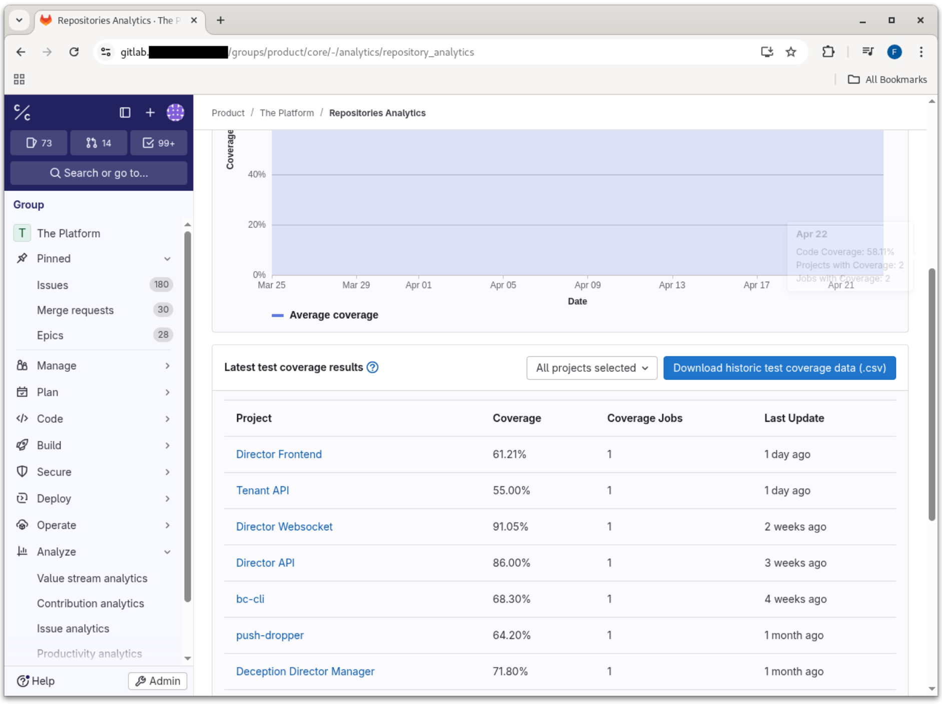
Development Process

After the Head of Development pushes code, the automated pipelines start to build code. Product development is a solution space for implementing high-quality, production-ready code. This area encompasses all production-bundled repositories and component-specific development. It follows iterative sprints for organized implementation and includes CI/CD practices and code reviews.



Code Quality & Coverage

The code is executed and all unit tests are triggered. From the unit testing we got the coverage metrics. The quality standards aim to get coverage over 80%, or at least 60%.



Apr 22
Code Coverage: 58.11%
Projects with Coverage: 2
Jobs with Coverage: 2

Latest test coverage results

Project	Coverage	Coverage Jobs	Last Update
Director Frontend	61.21%	1	1 day ago
Tenant API	55.00%	1	1 day ago
Director Websocket	91.05%	1	2 weeks ago
Director API	86.00%	1	3 weeks ago
bc-cli	68.30%	1	4 weeks ago
push-dropper	64.20%	1	1 month ago
Deception Director Manager	71.80%	1	1 month ago

Static code analyzer

Sonarqube is the tool mainly used for static code. The main rule is do not allow any new blocker, critical or major issues, aim for 0% duplications and increase coverage (or at least maintain it).

Old legacy repositories follow the rule to not increase issues, follow the standard style guide for coders and at least maintain coverage and duplicates.

After the build pipeline ends the artifacts are published and it is possible to create a release candidate for The Platform.

The testing pipeline focus on:

- Installation and update in real environments
- Raise all infrastructure pieces and let them play together (Deception Director, Tenan, Relay Nodes, Deception Hosts, Breadcrumbs, Rules, Integrations …)
- Perform an "End to end" testing using the web interface
- Perform complex operations via API
- Trigger attacks on controlled Deception Hosts

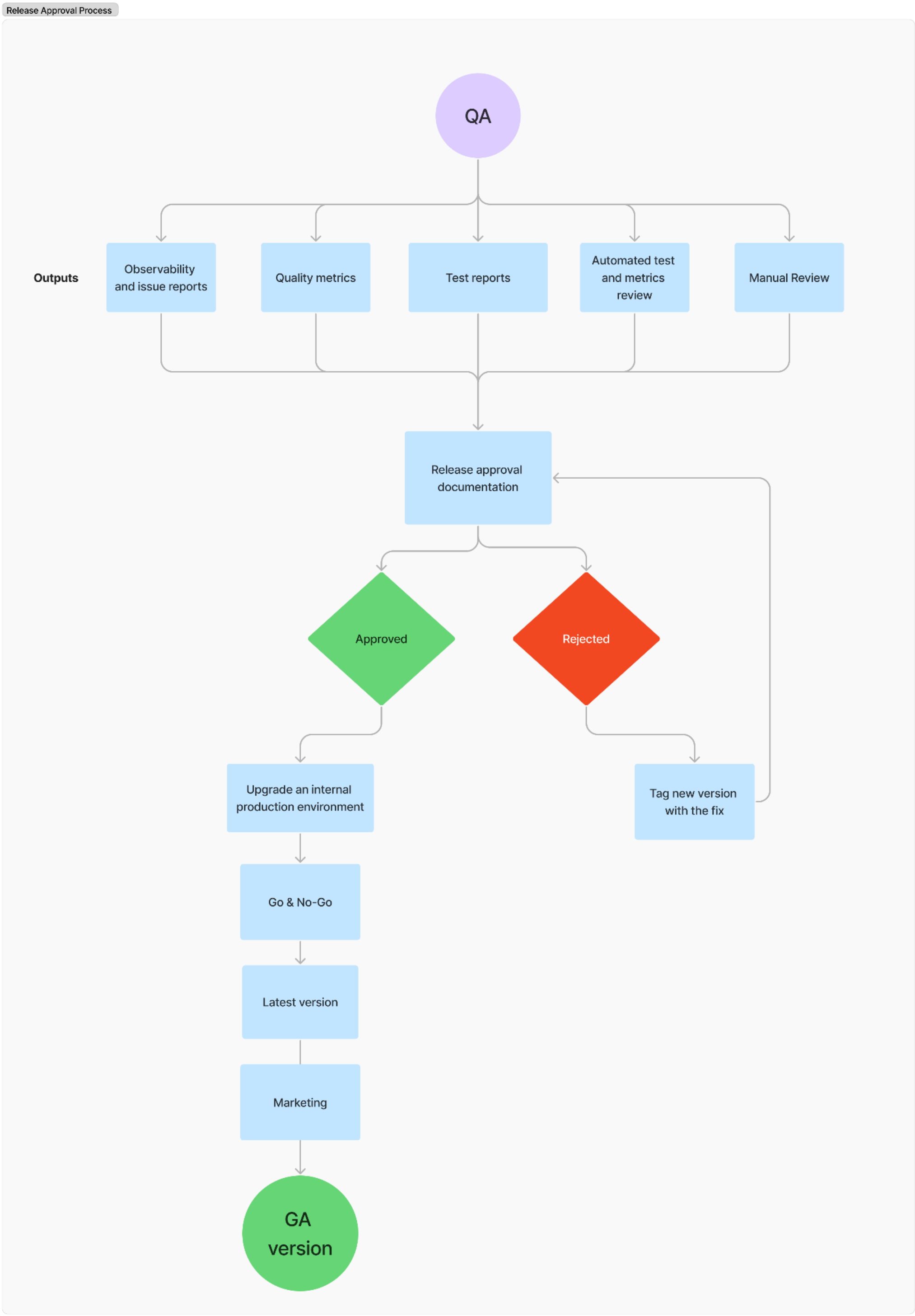
As this stage uses real machines in real cloud environments, small changes on resources, network connectivity or any external issue has a great impact on tests.

Our core principle is seamless installation and updates, along with robust infrastructure interoperability. We prioritize stable performance and full feature functionality, allowing for minor, non-critical errors.

Once this stage concludes, the release approval process moves forward.

Release Approval & Quality Check Process

At CounterCraft we use GitLab to manage and visualize its product release process, including a history of accepted and rejected versions.


Release Approval Process
QA
Outputs
Observability and issue reports
Quality metrics
Test reports
Automated test and metrics review
Manual Review
Release approval documentation
Approved
Rejected
Upgrade an internal production environment
Tag new version with the fix
Go & No-Go
Latest version
Marketing
GA version

- **Development**: This is the pre-release phase. It starts with direct builds from the development team. When this branch is stable enough the “freeze” branch is created. Version release: a tag is created inside the repository and an official bundle is created.
- **QA**: QA process for testing the new version:
  - Install from scratch and updates from previous releases
  - Manual review
  - Automated test and metrics review
  - At the end the release pass or is rejected
- **Early adopters:** The candidate version is deployed in early adopters environment for one week:- If non blocker issues are found, the version is considered accepted- If any minor issue is found but considered a known issue, it will be fixed in a future release and the version is accepted
- **Product Management:**
  - Go/No-Go phase. Accept or reject the product version.
  - Latest version (GA): The latest product version is updated to the new one for customers to download it.
  - Go To Market plan: Communicate the improvements from this release to end customers, and create all the necessary documentation.
- **Marketing**: Storytelling blog post with new release/features* Videos of key features* Use cases* Webinar (if great change/feature)