

Platform Release

[← Back to Core Processes](#)

Purpose

To ship Unicis Platform releases consistently, with clear quality gates, predictable communication, and no surprises for customers or the community.

Owner

Peter Zlatev — CTO

Trigger

This process starts when the team agrees a set of features/fixes is ready to ship — either at the end of a planned sprint or when a critical fix requires an unplanned release.

Steps

1. **Release candidate scoped** — Peter reviews open PRs and issues in GitHub. Agrees with Predrag on what is in scope for the release. Creates a GitHub milestone for the release version (e.g., v0.X.0). *\\Tool: GitHub*
2. **Feature freeze** — All in-scope PRs must be submitted. No new features added after feature freeze. Ongoing work continues in feature branches. *\\Tool: GitHub*
3. **Code review & merge** — All in-scope PRs reviewed and merged to main. At least one reviewer per PR (Peter or designated reviewer). *\\Tool: GitHub*
4. **QA pass** — Predrag and Peter run through the QA checklist for the release:
 - Core framework flows (NIS2, GDPR, ISO 27001) work end-to-end
 - SoA export (XLSX, PDF, HTML) renders correctly
 - REST API (at /api-docs) returns correct responses
 - Svix webhooks fire on key events
 - Auth flows (login, SSO if applicable) work correctly
 - No console errors or broken UI in Chromium and Firefox
 - Generate SBOM (syft) and Bearer scan (Code security scanning tool (SAST))

Tool: Unicis staging environment

1. **Release notes drafted** — Predrag drafts release notes in Markdown (Sveltia CMS). Structure: summary paragraph, What's New, Bug Fixes, Known Issues, Upgrade Notes. *\\Tool: Sveltia CMS / unicis.tech/admin*
2. **Tag and release** — Peter creates the GitHub release tag (vX.X.X), publishes the GitHub Release with the release notes, and triggers the deployment pipeline. *\\Tool: GitHub, CI/CD pipeline*
3. **Deployment verified** — Peter confirms the new version is live on the SaaS instance. Checks status.unicis.tech shows no incidents. Verifies Grafana metrics are nominal. *\\Tool: status.unicis.tech, Grafana*
4. **Release blog post published** — Predrag publishes the release blog post on unicis.tech. Post follows the

[writing style guide](#). Includes screenshots, changelog summary, and upgrade guide link. \\Tool: *Sveltia CMS*

5. **Community announcement** — Predrag posts the release announcement to: LinkedIn, Discord #announcements, Matrix #community, and GitHub Discussions. Links to the blog post. \\Tool: *LinkedIn, Discord, Matrix, GitHub*
6. **Mautic newsletter (if major release)** — For minor/patch releases, community posts are sufficient. For major releases (new framework, new module), Predrag sends a release email to the Mautic newsletter list. \\Tool: *Mautic*

Output

The process is complete when:

- The GitHub release is tagged and published
- The SaaS instance is running the new version with no active incidents
- The blog post is live on unicis.tech
- Community announcements are posted
- Grafana shows nominal metrics 30 minutes post-deployment

Tools

- **GitHub** — source control, PRs, milestones, release tags
- **CI/CD pipeline** — automated build and deployment (GitLab CI or GitHub Actions)
- **Unicis staging environment** — QA testing
- **Grafana / Prometheus** — post-deployment monitoring
- **status.unicis.tech** — public status page
- **Sveltia CMS** — blog post authoring
- **Mautic** — newsletter for major releases
- **Discord / Matrix / LinkedIn** — community announcements

Notes / Exceptions

- **Hotfixes:** For critical security or data integrity fixes, skip feature freeze and QA checklist (except auth and data flows). Ship immediately, post-mortem within 48h.
- **EU Project-linked releases:** If a release includes deliverables tied to OSC RAT, VIC, or another EU project, notify Alexander before publishing so he can update project documentation.
- **Breaking changes:** Any breaking change to the REST API or database schema requires a migration guide in the release notes and a minimum 2-week notice in GitHub Discussions before shipping.

Revision History

Date	Change	Updated by
2026-05-25	Initial version	Predrag Tasevski

[← Back to Core Processes](#)

[processes](#), [release](#), [engineering](#), [product](#)

From:

<https://handbook.unicis.tech/> - **Unicis Handbook**

Permanent link:

https://handbook.unicis.tech/pub/processes/platform_release

Last update: **25.05.2026 06:08**