



Change & Release Management

e: info@redactxpert.ai

w: <https://www.redactxpert.ai>

Powered by Simpson Associates



Contents

1. Introduction.....	3
1.1 Purpose	3
1.2 Scope	3
2. Types of Change	4
3. Change Classification & Prioritisation	5
3.1 Prioritisation.....	5
3.2 Change Classification Matrix.....	5
3.2.1 Change Urgency Table	5
3.2.2 Change Risk Table	5
3.2.3 Change Impact Matrix	5
4. Change Approval	6
4.1 Standard Changes	6
4.2 Normal Changes	6
4.3 Emergency Changes	6
5. Release Management & CI/CD Process.....	7
5.1 Release Planning	7
5.1.1 Release Frequency	7
5.1.2 Scheduling.....	7
5.1.3 Testing Strategy	7
5.1.4 Rollback Plan.....	7
5.1.5 Documentation	7
5.2 Pre-Release Checks	7
5.3 Deployment.....	7
5.4 Versioning	8
5.4.1 Version Number Format	8
5.5 Post-Release Validation	8
6. Change & Release Communication	9
6.1 Release Notes.....	9
6.2 Customer Notifications	9
7. Post-Implementation Review (PIR)	9
8. Change Monitoring & Continual Improvement.....	10
8.1 Ongoing Monitoring.....	10
8.2 Continuous Service Improvement.....	10
9. Change Process Flow	11

1. Introduction

1.1 Purpose

This document outlines the change and release management process for the RedactXpert SaaS solution. This process ensures that changes to RedactXpert are:

- Introduced in a controlled, predictable manner
- Properly assessed for risk and impact
- Tested and approved before release
- Communicated appropriately to stakeholders
- Reversible if issues occur

1.2 Scope

This process applies to:

- Application code (frontend, backend, APIs)
- Infrastructure and platform changes
- Configuration changes
- Security-related changes
- Third-party dependency upgrades
- Data migrations
- Production deployments only

Out of scope

- Routine operational tasks with no customer impact (e.g. log reviews)
- Incident response activities (covered by Incident Management)
- Changes/releases made to any environments other than production

2. Types of Change

Type	Description	Examples	Change Recording
Standard	Pre-approved, low-risk, and frequently occurring changes with a pre-defined workflow following CI/CD processes.	• Minor UI tweaks • Non-breaking bug fixes • New features with low potential impact for users • Schema additions without the potential for data loss • Routine dependency updates	Release notes are required for each version. A pre-approved change record is required in the ITSM tool ahead of release.
Normal	Changes that require assessment, authorisation, and approval, usually those with a larger potential for customer or system impact.	• Changes to existing features • New features with high potential impact for users • Infrastructure updates • Schema changes with potential for data loss • Data migration work	Release notes are required for each version. A change record is required in the ITSM tool ahead of release.
Emergency	High priority changes that need to be implemented urgently to prevent or resolve major incidents, or to restore service.	• Security patches after a breach • High-priority bug fixes to restore service ("hot fixes") • Fix critical vulnerabilities	A change record is required in the ITSM tool. This can be documented retrospectively.

3. Change Classification & Prioritisation

3.1 Prioritisation

The priority of the change is determined based on its urgency and potential risk. This considers elements such as business needs, regulatory compliance, and security concerns.

3.2 Change Classification Matrix

Changes are classified and prioritised using the following tables:

3.2.1 Change Urgency Table

Urgency	Description
Critical	- Must be completed immediately or as soon as possible - The solution is unavailable without the change being implemented
High	- Change must be completed within business hours - Core features of the solution are unavailable
Medium	- Change can be completed outside of business hours - Small impact to user experience
Low	- Change can be completed outside of business hours - End user experience is not affected

3.2.2 Change Risk Table

Risk	Description
Critical	- Critical security implications - Low/no testability - Long recovery time (e.g. into business hours) or no recovery possible
High	- High security relevance - Low testability - Long recovery time (e.g. into business hours) or no recovery possible
Medium	- Medium security relevance - Medium testability - Medium recovery time
Low	- Low security relevance - High testability - Short recovery time

3.2.3 Change Impact Matrix

		Risk			
Urgency		Critical	High	Medium	Low
	Critical	Extensive	Extensive	Significant	Moderate
	High	Extensive	Extensive	Significant	Moderate
	Medium	Significant	Significant	Moderate	Minor
	Low	Moderate	Moderate	Minor	Minor

4. Change Approval

Changes are approved based on the type of change being conducted:

4.1 Standard Changes

Standard changes are pre-approved and do not require additional approvals outside of the CI/CD processes.

4.2 Normal Changes

Normal changes are reviewed by the product and technical teams, and approval is required from the technical lead to implement.

4.3 Emergency Changes

Emergency changes are reviewed by technical teams and approval may be recorded retrospectively to minimise downtime and prevent major incidents.

5. Release Management & CI/CD Process

5.1 Release Planning

5.1.1 Release Frequency

Releases are planned to occur in line with development of new features and bug fixes. There is no regularly recurring release schedule with a fixed time window (e.g. the first Wednesday of every month), however new releases occur roughly once a month. In some cases, releases may be more frequent if there are security vulnerabilities or critical fixes that need to occur.

5.1.2 Scheduling

Releases will be performed outside of business hours wherever possible. If there is a need to perform a change inside business hours then we will communicate this in advance, except for emergency changes.

5.1.3 Testing Strategy

Changes should have a clear testing strategy in place to verify the changes following release.

5.1.4 Rollback Plan

A rollback strategy should be in place for all changes. The default rollback strategy for Standard Changes following the CI/CD process is to utilise the CI/CD pipelines to deploy the previous known-working version.

The rollback strategy for a normal change should be planned and documented as part of the change record

5.1.5 Documentation

New features, bug fixes, security patches etc. are documented as part of release notes for a new version of the application.

- Standard changes should have a (pre-approved) change record produced in the ITSM tool.
- Normal changes should have a change record produced in the ITSM tool.
- Emergency changes should have a change record produced in the ITSM tool (this can be done retrospectively).

5.2 Pre-Release Checks

Before each release, the following checks are completed (where applicable):

- Code is reviewed by members of the technical development team
- Automated tests are executed and must pass for code to be merged
- Testing is conducted (e.g. functional testing)

5.3 Deployment

- Standard code and infrastructure changes are deployed via automated CI/CD pipelines where possible to ensure repeatable, resilient deployments. These include stages for each environment.
- Deployments to the production environment include a step for approval by a technical lead (this is always required). Note that deployments to other environments are not considered changes and fall into the standard software development lifecycle.

5.4 Versioning

- The application code is versioned to be able to track changes over time.
- Standard code releases performed using the CI/CD processes will receive a new version.
- The version number is available in the application for reference.
- Changes that do not alter the application code will not receive a new version.

5.4.1 Version Number Format

Application version numbers have the following format:

YYYYMMDD.R

- **YYYYMMDD** indicates the date of the version (specifically, the date the build artifact was produced in the CI/CD pipeline). For example, if the build artifact was produced on the 15th of January 2026, this would be 20260115.
- **R** indicates the revision number for the version. This is auto incremented by the CI/CD process to ensure unique version numbers, even if multiple build artifacts are created on the same day.

5.5 Post-Release Validation

Following a release:

- Validation that the version showing in the application is the version that has been deployed (if release is performed via the CI/CD processes).
- Testing is performed to ensure the changes have been implemented successfully, and core functionality is working as expected.
- The change is monitored during and after implementation to detect any potential problems. Any issues are resolved according to the rollback plan or through additional adjustments.
- Any required customer-impact checks are completed.

Releases are only considered successful once post-release validation passes.

6. Change & Release Communication

6.1 Release Notes

Release notes are produced for all new versions of the application. This includes details of new features that have been introduced, bug fixes, and security patches etc.

6.2 Customer Notifications

Some changes will be communicated to customers where there is a direct impact on that customer's setup, or if there is a large potential for disruption. This would be for changes that are classified as having an Extensive impact.

In general, performed outside of business hours following the CI/CD processes will not be communicated to customers, and instead details will be available via the release notes.

7. Post-Implementation Review (PIR)

For any changes that are unsuccessful, a Post-Implementation Review (PIR) meeting is conducted. This assesses:

- The root cause of any issues
- Lessons learnt from the change
- Effectiveness of the testing and rollback plans
- Next steps for the change

8. Change Monitoring & Continual Improvement

8.1 Ongoing Monitoring

Regular monitoring of a change's impact on services is conducted to ensure long term stability and performance. This is conducted using the monitoring tools and systems in place within the solution to check for errors, availability and performance etc.

8.2 Continuous Service Improvement

Lessons learned from each change are used to refine the Change & Release Management Process, improve efficiency and reduce risks in future changes.

9. Change Process Flow

