User Guide

Backlog Auditor & Export

Introduction

Backlog Auditor & Export is a Forge-powered Jira Cloud application designed to help project administrators and teams **systematically monitor and improve backlog quality**.

Backlogs often grow messy — with missing fields, outdated issues, or inconsistent data. Manually checking hundreds (or thousands) of issues is both time-consuming and error-prone. Backlog Auditor & Export solves this problem by:

- Allowing administrators to **define rule-based checks** for fields across issues.
- Automatically detecting **violations** that deviate from agreed standards.
- Presenting **actionable reports** that highlight trends, bottlenecks, and recurring issues.

Instead of asking “Why is our backlog full of unclear or outdated issues?”, Backlog Auditor & Export provides a **clear, visual, and repeatable way to enforce standards** and drive backlog discipline.

Prerequisites

Before installing or using Backlog Auditor & Export, ensure the following:

- **Jira Environment**: Only supported on **Jira Cloud**. (Jira Data Center and Server are not supported yet).

- **Installation**: Install via the Atlassian Marketplace or deploy through Forge.
- **Permissions:**
  - Project Administrators – Can configure rulesets.
  - Team Members – Can view audit reports if they have project access.
- **Access to Fields & Workflows:** For meaningful audits, the user must have visibility of project fields (system and custom) and workflows.
- **Browser:** Use an up-to-date browser (Chrome, Firefox, or Edge).

Ensure all fields that you want to audit are visible and not restricted by field configurations or screen schemes.

Application Structure

The app consists of two key areas, each serving a distinct purpose:

1. **Admin Page** – This is where administrators design and manage rulesets. Think of it as setting the “audit checklist” for a project.
2. **Project Page (Auditing)** – This is where the audits are executed and results are displayed in an easy-to-read format.

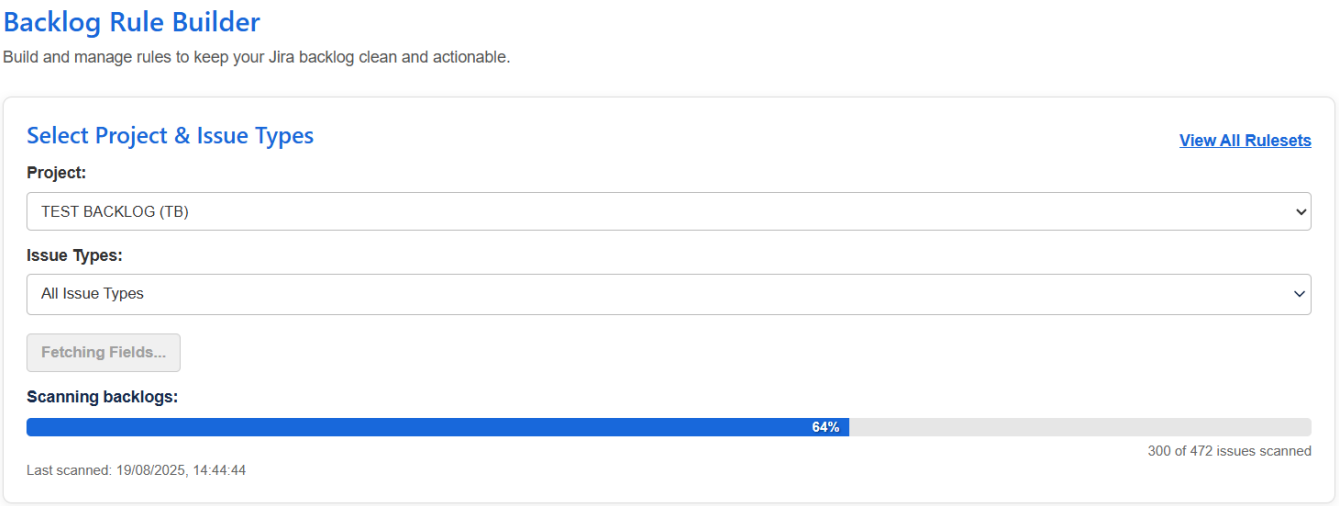
This separation ensures that **rule creation and audit execution** are distinct, reducing errors and keeping responsibilities clear.

App Flow

Admin Page – Building Rulesets

When you first configure the app:

1. **Select a Project** – Narrow scope to one project at a time.
2. **Select Issue Types** – Decide whether audits apply to Stories, Bugs, Epics, or custom issue types.
3. **Fetch Relevant Fields** – The app loads only the fields that apply to your selected project and issue types. On first use, it performs a live discovery from Jira; afterward it serves the results from a cached snapshot (shown with a Last scanned timestamp) for speed. Use Rescan to bypass the cache and fetch a fresh set directly from Jira.

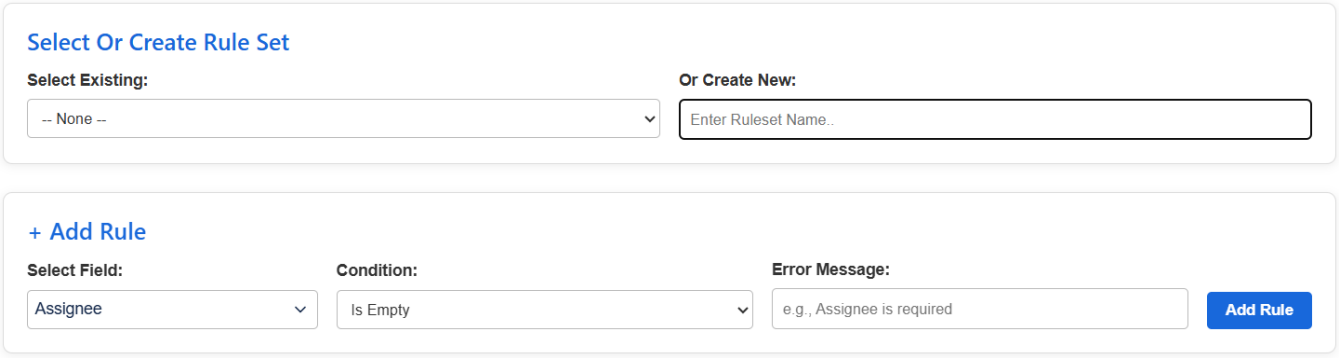


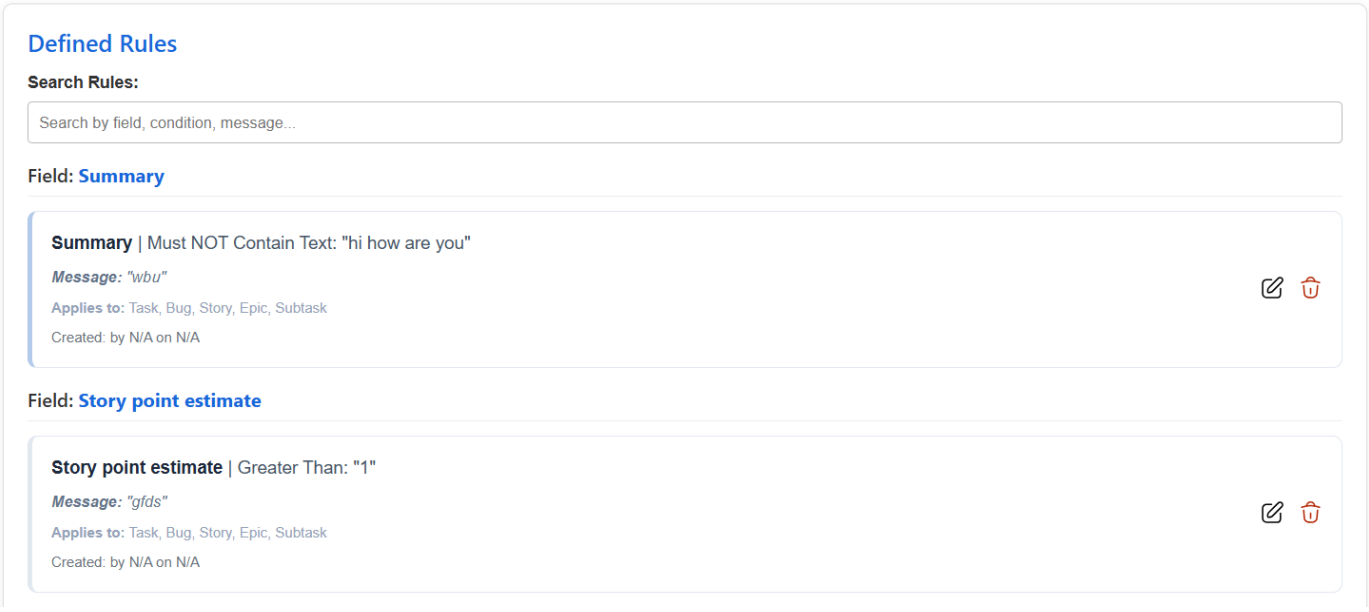
4. **Choose or Create a Ruleset** – Reuse a prior ruleset or start fresh.
5. **Add Rules – For each rule:**

- Pick a field.
- Select a condition (based on field type).

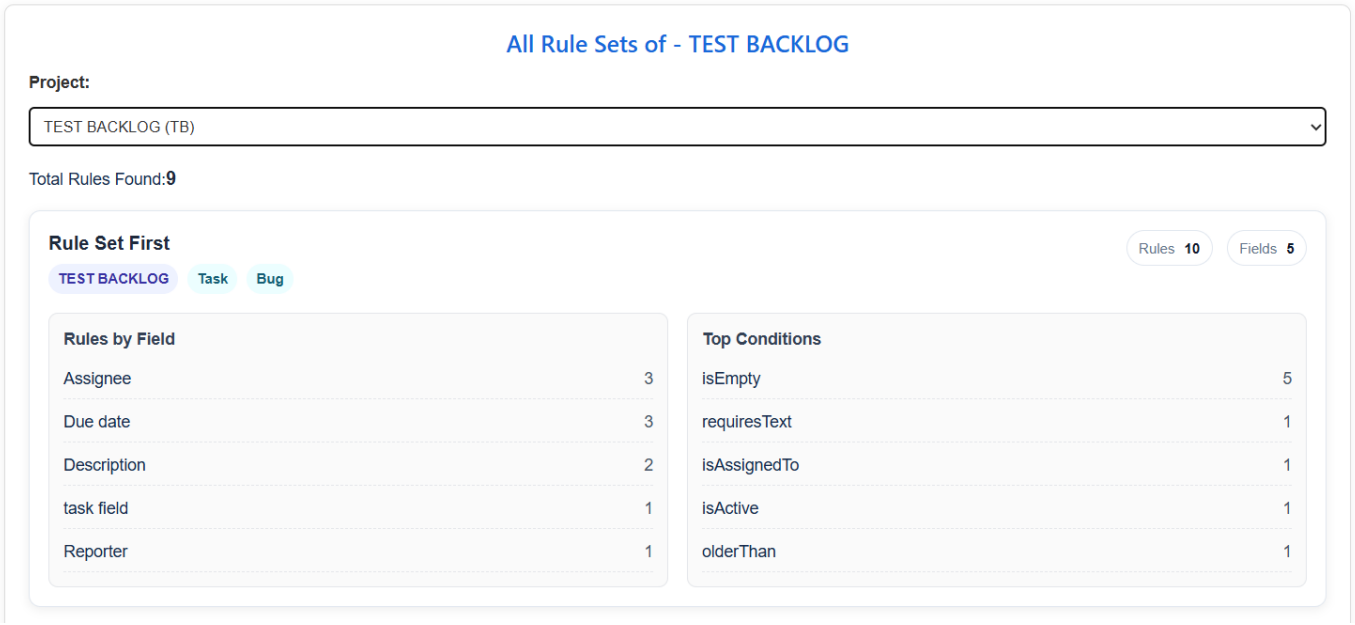
- Write a violation message (guidance for team members).
- Add the rule to the ruleset.

6. **Save Ruleset** – Once satisfied, store the ruleset for future audits.





7.. **View All Ruleset** – Quickly access a list of all existing rulesets for the selected project, making it easier to review, compare, or reuse them in future audits.

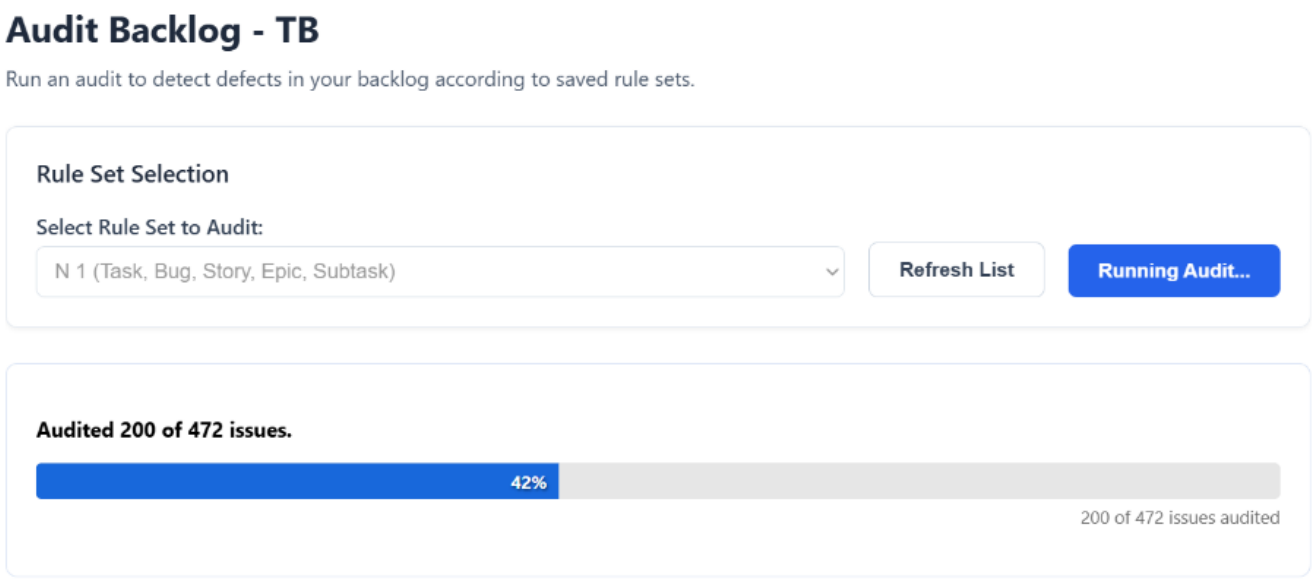


Think of the Admin Page as designing the “rules of cleanliness” for your backlog. Once created, these rules can be reused and refined over time.

Project Page – Running Audits

When running an audit:

1. **Select a Ruleset** – Choose from the rulesets available for that project.
2. **Run Audit** – The app will automatically check each issue against your ruleset.



3. **Review Results** – Results are presented in a structured, visual, and easy-to-interpret format:

- **Summary Cards** – Quick insights (e.g., total audited, total violations, most common violations).
- **Charts** – See defects by assignee, issue type, condition, and field.
- **Defects Trend Chart** – Track progress over time. For example, you can compare last sprint's audit to this sprint to see if data quality improved.
- **Detailed Table** – Get a row-by-row breakdown of issues and what failed.

Understanding the Audit Report

Backlog Auditor & Export's reporting is designed to be both executive-friendly and action-oriented.

Summary Cards

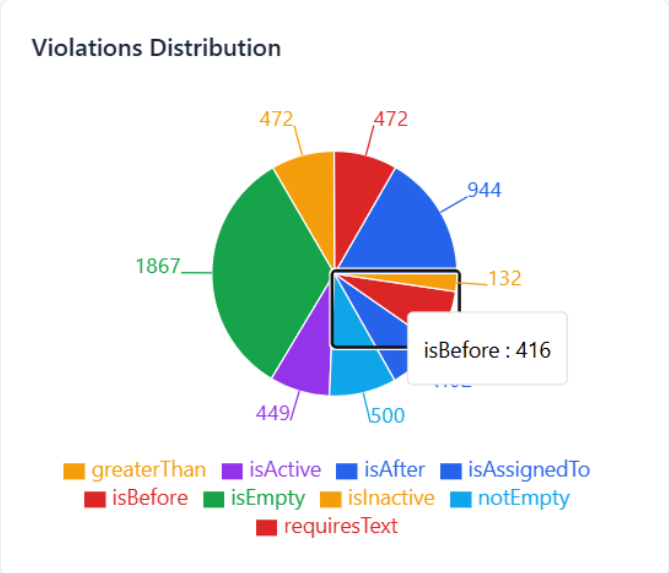
At-a-glance insights:

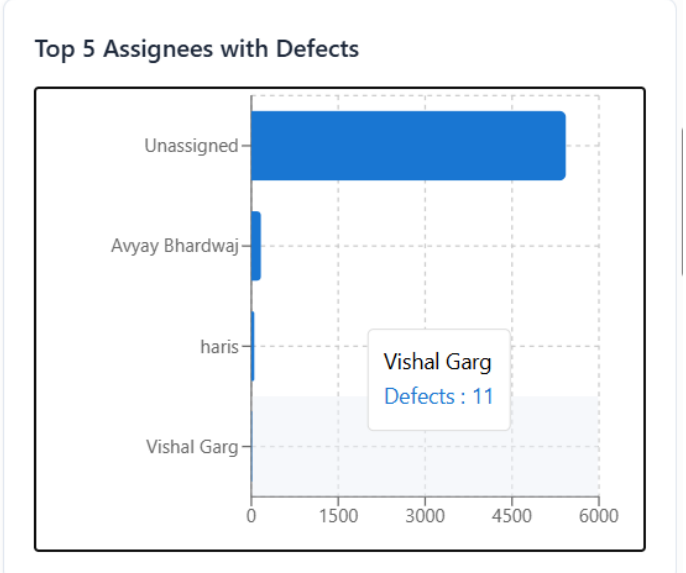
- Total Audited – How many issues were checked.
- Total Violations – Number of rule breaks.
- Top Violated Condition – The most common problem.
- Defects by Assignee/Field/Issue Type – Who/what is most affected.

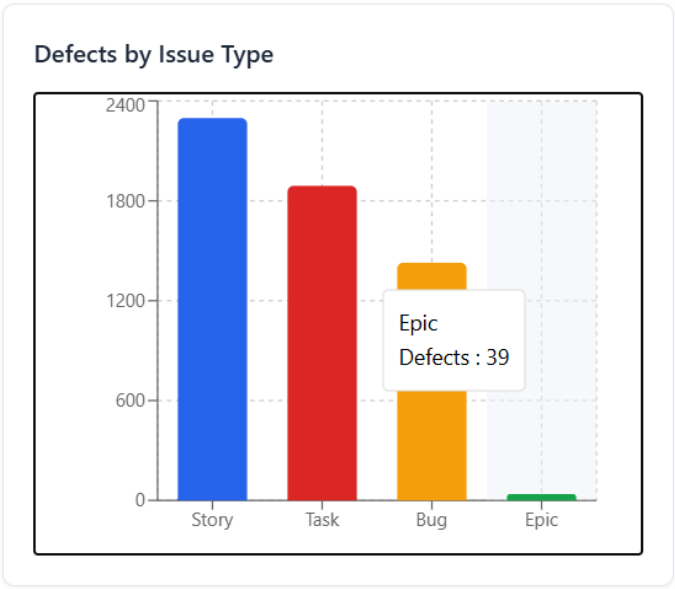
Total Audited **472**	Defects Found **5654**	Defect Rate **1197.88%**	
Top Violated Rule **isEmpty**	Top Assignee **Unassigned** 5428 defects	Top Field **Assignee** 1819 defects	

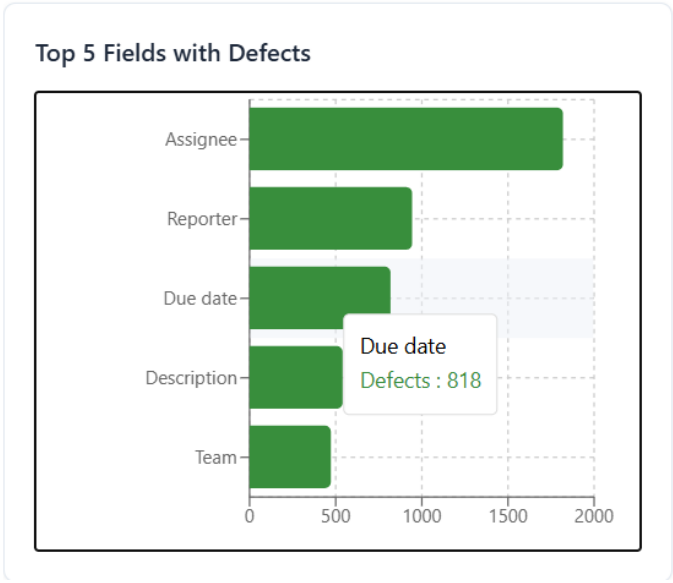
Charts

- **By Condition** – Which rules fail most often.
- **By Field** – Spot fields with recurring issues.
- **By Assignee** – Detect team members needing more support or clarity.
- **By Issue Type** – Understand where problems cluster (Stories vs. Bugs).





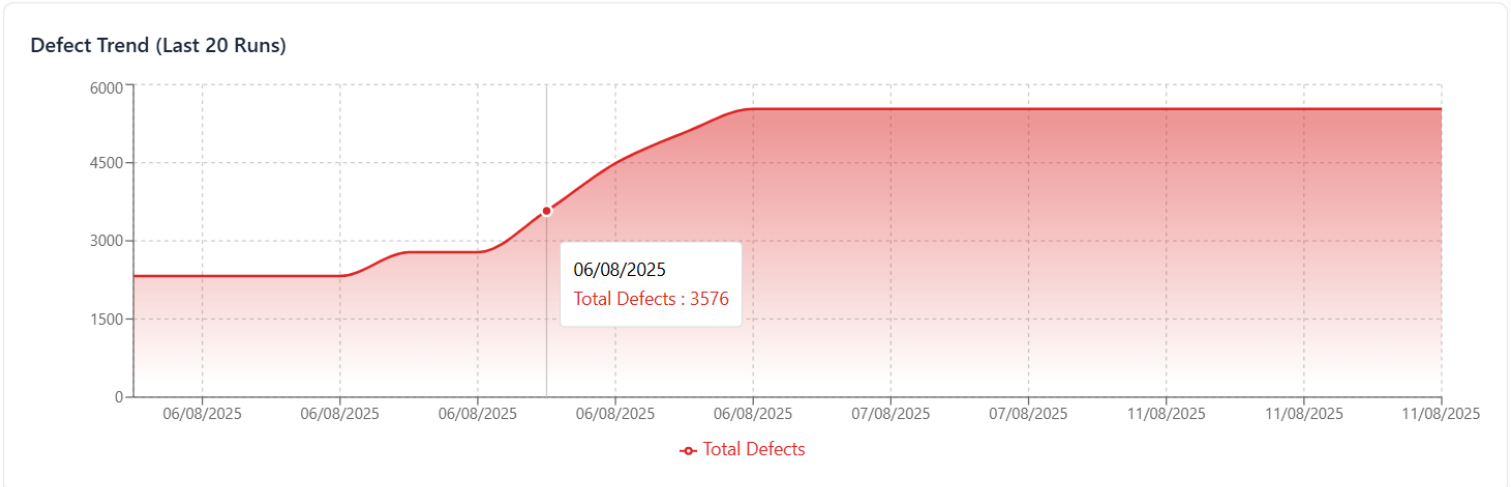




Defects Trend Chart

One of the most powerful features. It shows how violations evolve overtime.

- If violations decrease → Backlog health is improving.
- If violations increase → Standards are not being followed, and corrective action is required.

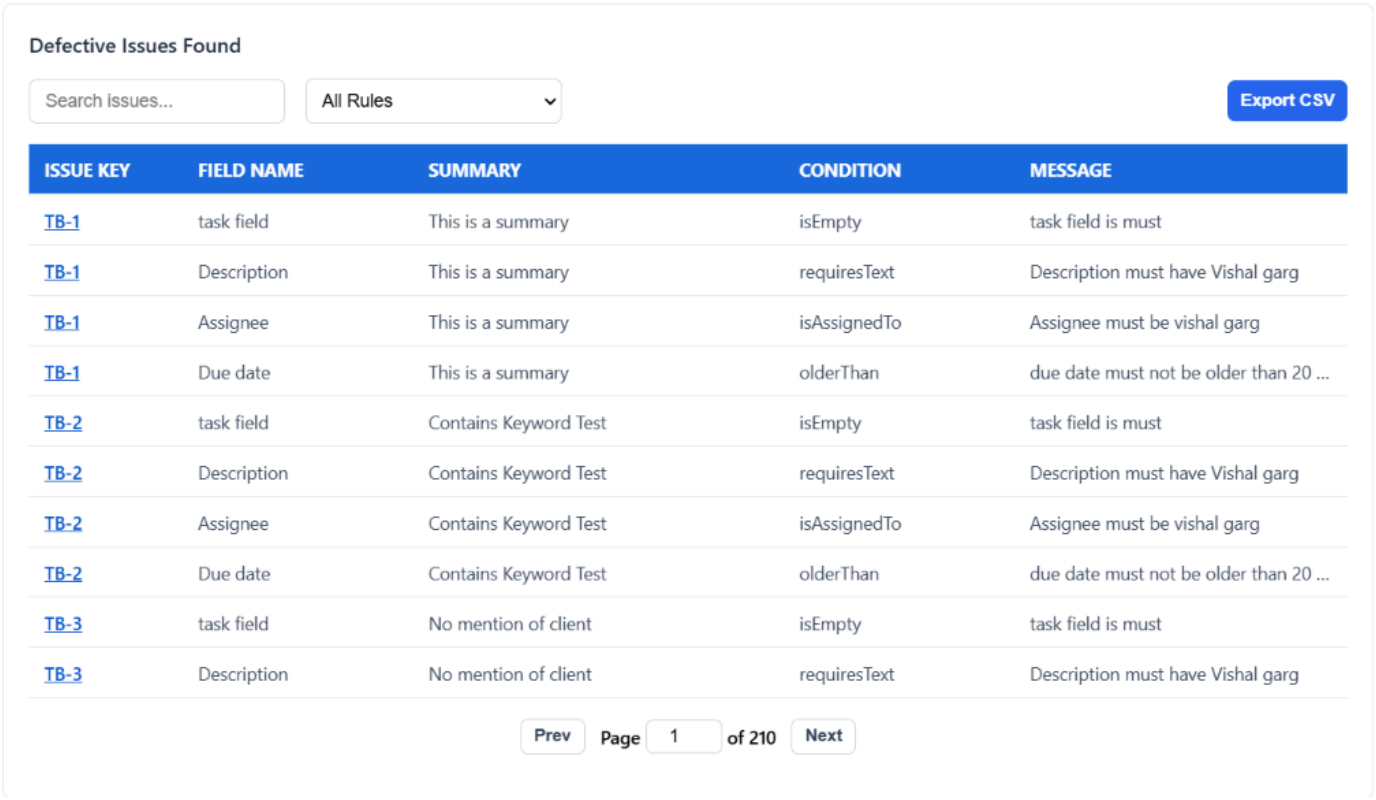


Defects Table

The table provides detailed evidence:

- Issue Key (linked to Jira).
- Summary.
- Field Name & Type.
- Condition that failed.
- Violation Message.

This table acts as the “to-do list” for teams. Every row highlights what to fix in order to clean up the backlog.

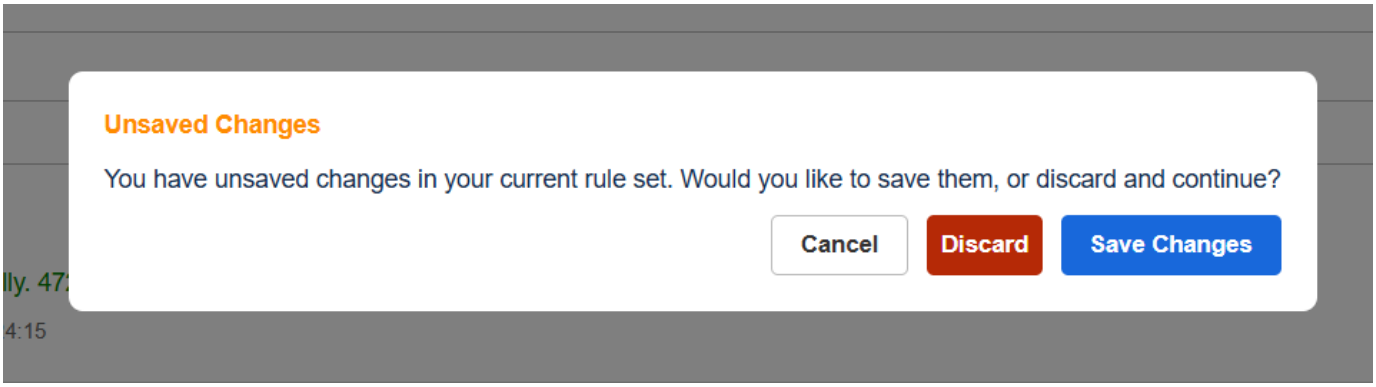


Friendly Fallbacks & Safety Nets

Backlog Auditor & Export is designed to be forgiving and user-friendly. Across the app you'll find protective fallbacks that prevent accidental data loss and guide you to the next best step. The most visible example is the Unsaved Changes protection shown below.

Unsaved Changes Protection

Whenever you edit, add, or delete a rule—and then try to navigate away, switch projects/issue types, close the page, or perform another action without saving—Backlog Auditor & Export surfaces an Unsaved Changes dialog.



When it appears

- Editing a rule's field, condition, or message
- Adding or deleting a rule in a ruleset
- Switching the selected project or issue type while there are edits
- Leaving the Admin page, refreshing, or closing the tab with pending edits

What the buttons do

Button	Result
Cancel	Closes the prompt and keeps you on the current screen so you can continue editing. No changes are saved.
Discard	Proceeds with your action without saving the pending edits in the current ruleset.
Save Changes	Saves the pending edits to the ruleset and then continues with your action.

Other Built-In Safeguards

To keep your workflow smooth, Backlog Auditor & Export includes additional safety nets:

- **Delete confirmations**: Removing a rule or an entire ruleset asks for confirmation to avoid accidental deletions.
- **Helpful empty states**: If a selection (e.g., fields for a chosen issue type) returns nothing, the UI explains why and how to proceed.
- **Permission-aware messaging**: If you lack access to a project, field, or workflow element, you'll see a clear message instead of a silent failure.
- **Graceful loading & retries**: Data fetches show progress indicators and provide guidance if something takes longer than expected.

Supported Rule Conditions

Backlog Auditor & Export offers a wide variety of conditions tailored to field types, making rules precise and relevant.

String Fields (Summary, Description, custom text fields)

- **Require or block specific words.**
- **Enforce equality/inequality.**

Number Fields (Story Points, Estimates)

- Check emptiness.
- Require exact or non-exact values.
- Enforce numeric comparisons (>, <, ≥, ≤).

Date/DateTime Fields (Due Date, Created, Updated)

- Detects empty values.
- Identify older or newer issues based on days.
- Compare with a specific date.

User Fields (Assignee, Reporter)

- Check assignment.
- Match specific users.
- Detect inactive accounts.

Priority, Status, Issue Type

- Check for exact matches or mismatches.

Default (Fallback)

- Is Empty / Is Not Empty.

These conditions make rules extremely flexible. For instance, you can enforce that “Stories must always have story points,” or “Bugs older than 60 days must be reviewed.”

Example Rule Scenarios

Ensure Descriptions Exist

- Field: Description
- Condition: Is Empty
- Violation Message: “Please add a description so the team has context.”

Catch Old Bugs

- Field: Created Date
- Condition: Older Than (days) → 90
- Message: “This bug is over 90 days old. Reassess its relevance.”

Check Assignments

- Field: Assignee
- Condition: Is Empty
- Message: “This issue is unassigned. Assign it to a responsible person.”

Validate Estimates

- Field: Story Points
- Condition: Greater Than → 13
- Message: “Story points exceed recommended range (max 13).”

Best Practices

These conditions make rules extremely flexible. For instance, you can enforce that “Stories must always have story points,” or “Bugs older than 60 days must be reviewed.”

- **Start small:** Begin with a handful of high-value rules before expanding.
- **Write actionable violation messages:** Instead of just “Invalid,” guide users with “Story points must be ≤ 13.”
- **Audit regularly:** Run audits every sprint or month to ensure continuous improvement.
- **Monitor trends:** Use the trend chart to demonstrate backlog improvements to stakeholders.
- **Collaborate with teams:** Share reports during sprint reviews or grooming sessions.

Troubleshooting

- **Audit does not start** → Check project permissions and ruleset existence.
- **Ruleset missing** → Confirm it was saved under the correct project.
- **Empty charts** → Ensure issues actually match your audit conditions.
- **Unexpected results** → Validate field visibility and whether custom fields are indexed.

Support

Email: developers@Clovity.com

Website: clovity.com