

Test Case and Impact Analysis Generator by Clovity AI



1. Overview

The AI Test Case & Impact Analysis Generator is a Jira Forge app that helps QA engineers, developers, and product owners:

- Automatically generate test cases from Jira issues
- Perform impact analysis for better risk management
- Seamlessly create subtasks in Jira from AI suggestions
- Edit, manage, and export results in a structured format

2. Getting Started

Follow these steps to begin using the Test Case and Impact Analysis Generator by Clovity AI app in Jira.

2.1 Prerequisites

- **Jira Permissions**
 - Rights to create subtasks and linked tasks.
 - Edit permissions for updating test cases.
- **App Installation**
 - Verify that the app has been installed and enabled by your Jira administrator
 - The app will appear inside the issue view panel
- **Supported Issue Types**
 - Works with Epics, Stories, Tasks, and Bugs
 - Cannot be used on issues that are already Test Cases

- **Quality of Inputs**

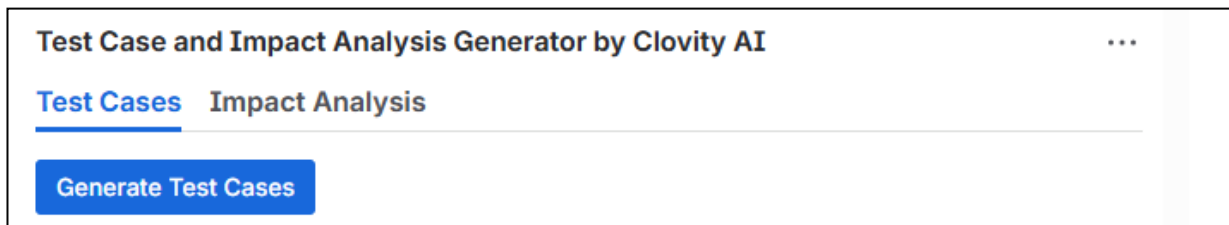
- Provide clear and detailed issue descriptions and acceptance criteria for best results
- Ambiguous or incomplete requirements may lead to less accurate test cases

2.2 Launching the App

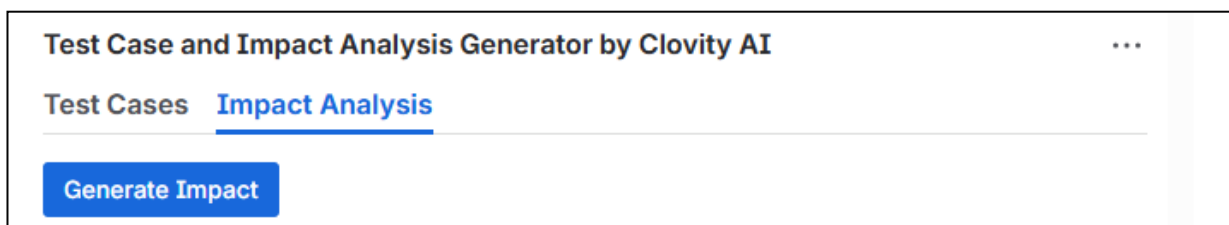
1. Open a Jira issue (Epic, Story, Task, or Bug).
2. Locate the panel titled “Test Case and Impact Analysis Generator by Clovity AI” in the issue view.

3. Tabs available:

Test Cases → For generating and managing test case suggestions



Impact Analysis → For generating AI-based impact insights



4. If the issue is a Test Case, the panel will display the message:
“This issue appears to be a Test Case. To use the AI Test Case & Impact Analyzer, please open an Epic, Story, Task, or Bug instead.”

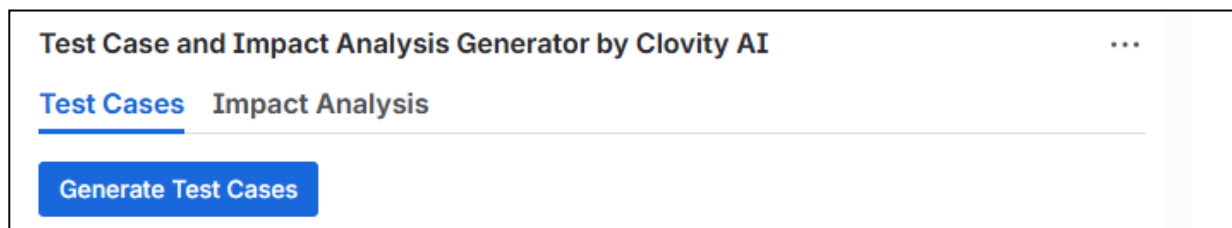
2.3 First-Time Use

2.3.1 Generate Test Cases

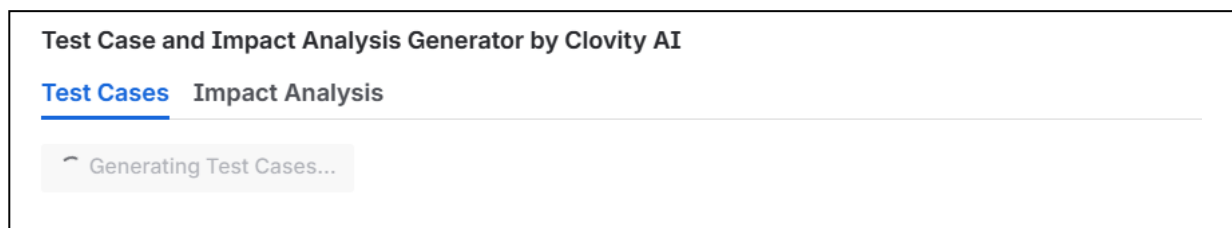
The Test Cases module allows you to automatically generate, refine, and create detailed test cases directly from Jira issues.

Steps to Use

1. Open a Jira issue (Epic, Story, Task, or Bug).



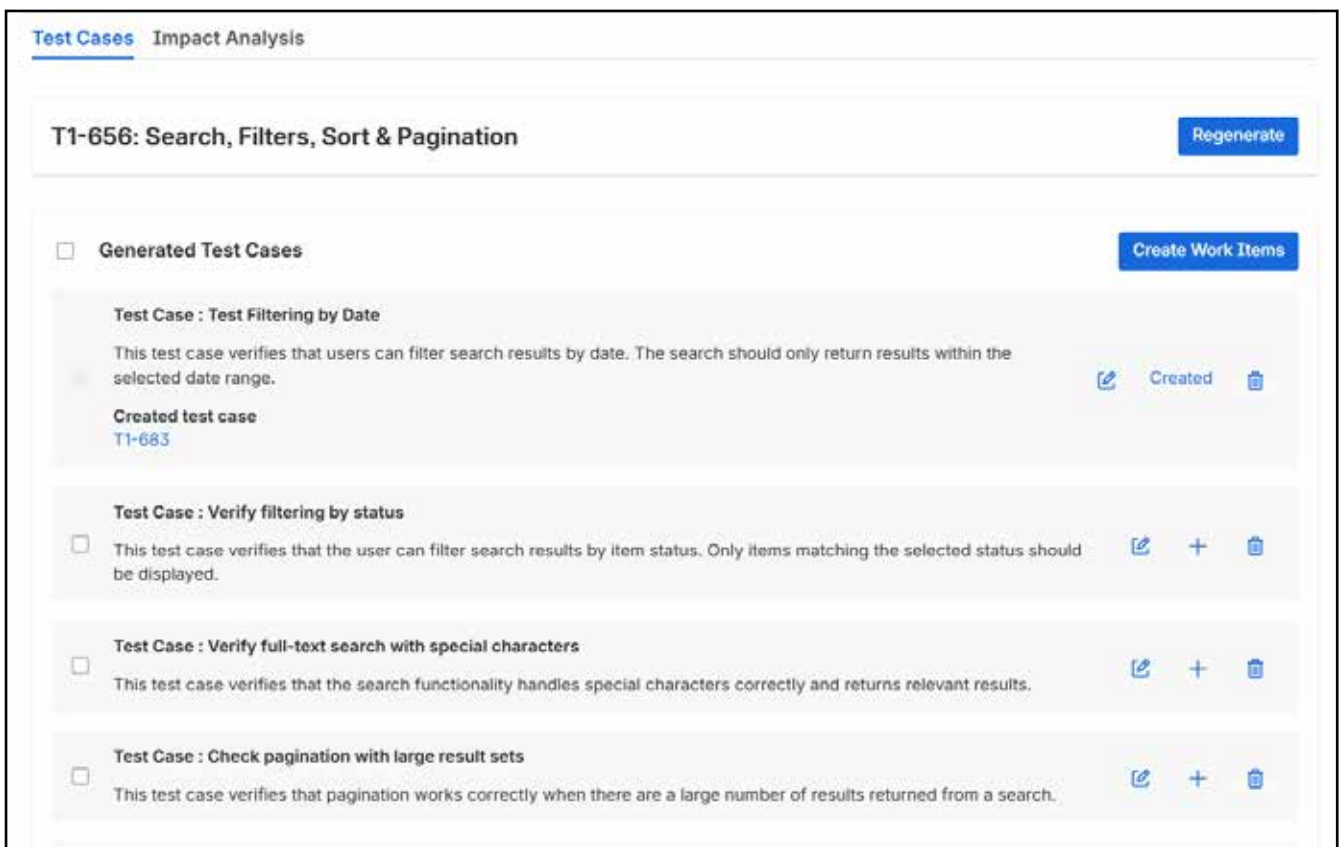
After Click Test case generation is in process



2. The AI analyzes the issue description and acceptance criteria to produce suggestions.
3. Each suggestion includes:
 - Title
 - Description
 - Preconditions and Steps
4. Review the list of generated suggestions.
5. Select one or more test cases using the checkboxes.
6. Click Create Work Items to save them into Jira as subtasks (or linked tasks if subtasks are not supported under the parent issue).

Available Actions

- **Regenerate** → Creates a fresh set of suggestions with variations.
- **Edit** → Modify details before creating the test case (Title, Description, Preconditions, Test Steps).
- **Add (+)** → Create the individual suggested test case as a work item in Jira.
- **Delete** → Remove suggestions you do not want to create or keep.



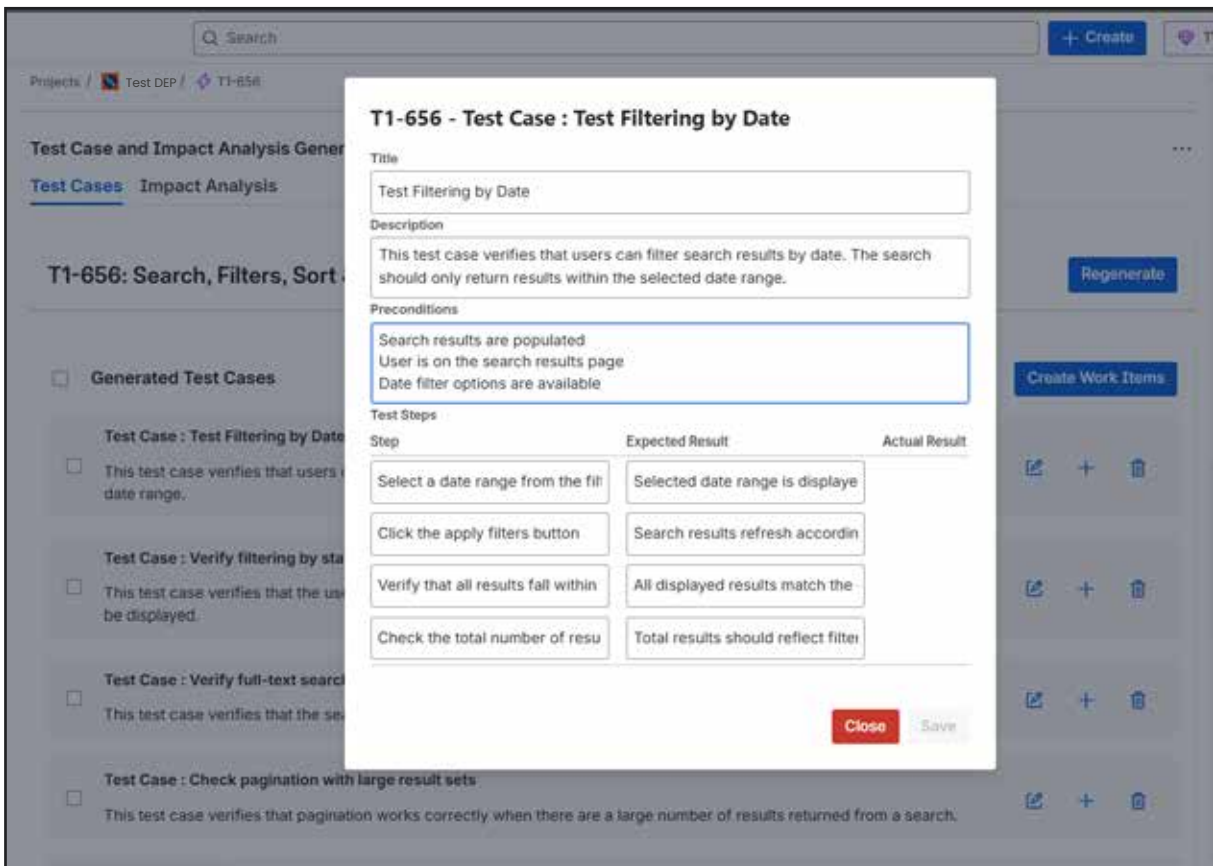
The screenshot shows the 'Test Cases' tab in a Jira interface. The main heading is 'T1-656: Search, Filters, Sort & Pagination'. There is a 'Regenerate' button in the top right. Below the heading, there is a section titled 'Generated Test Cases' with a 'Create Work Items' button. The section contains four test cases, each with a checkbox, a title, a description, and a 'Created test case' ID. The first test case is 'Test Case : Test Filtering by Date' with ID 'T1-683'. The second is 'Test Case : Verify filtering by status'. The third is 'Test Case : Verify full-text search with special characters'. The fourth is 'Test Case : Check pagination with large result sets'. Each test case has a description and a 'Created' status with a trash icon.

Work Item Details

When a generated test case is created in Jira, it becomes a structured subtask or linked task with the following information:

- **Title** → Test case name (e.g., User Account Creation with Valid Details)
- **Description** → Explanation of what the test case verifies
- **Preconditions** → Conditions that must be met before execution (e.g., User is on the signup page)

- **Test Steps** → A structured table with:
 - Step Number
 - Test Step
 - Expected Result
 - Actual Result field (left empty for execution tracking)



The screenshot shows the Jira Test Case editor for 'T1-656 - Test Case : Test Filtering by Date'. The form includes fields for Title, Description, Preconditions, and a table for Test Steps.

Step	Expected Result	Actual Result
Select a date range from the filter	Selected date range is displayed	
Click the apply filters button	Search results refresh accordingly	
Verify that all results fall within the selected date range	All displayed results match the selected date range	
Check the total number of results	Total results should reflect filter	

2.3.2 Generate Impact Analysis

The **Impact Analysis** feature helps QA engineers, developers, and product owners understand the risks and dependencies of a Jira issue before implementation or testing.

Steps to Use

1. Switch to the **Impact Analysis** tab.
2. Click **Generate Impact**.

1. The AI analyzes the issue description, acceptance criteria, and related details to generate results.

Test Case and Impact Analysis Generator by Clovity AI

...

Test Cases
Impact Analysis

Generate Test Cases

Output Categories

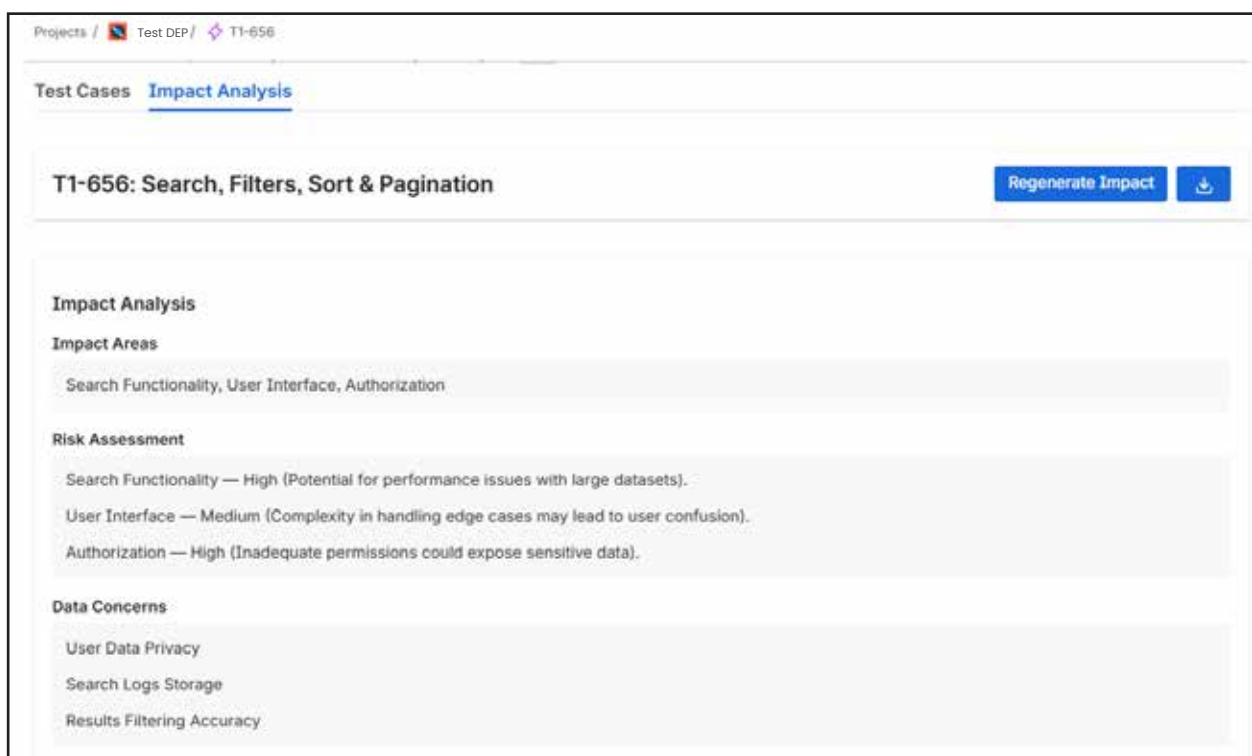
1. **Impact Areas** → Functional or technical areas affected (e.g., User Authentication, Data Security, User Experience).
2. **Risk Assessment** → Risk level for each area (High/Medium/Low) with reasoning.
 - Example:
 - **User Authentication** — High (Weakness in MFA can lead to unauthorized access).
 - **Data Security** — Medium (Improper session management could expose user data).
3. **Data Concerns** → Issues with data handling, security, or compliance.
 - Example: Personal data may be logged in error messages.
4. **Mitigations** → Strategies to reduce identified risks (e.g., input validation, encryption, session timeout).
5. **Regression Hotspots** → Areas likely to break due to changes (e.g., password reset impacting SSO).
6. **Smoke Suite Recommendations** → High-level tests to verify core functionality remains stable.

Refreshing Insights

- Click **Regenerate Impact** to generate updated insights if requirements have changed.

Export Results

- Click **Download PDF** to export a structured report containing:
 - Issue key and summary
 - Test cases with details
 - Impact analysis results
- PDFs can be stored for audits or shared with stakeholders.



2.4 Best Way to Start

To maximize value from the app:

- Start with well-documented Jira issues that contain clear descriptions and acceptance criteria
- Use Regenerate if initial results are not fully relevant
- Create only selected test cases to avoid cluttering Jira boards
- Use bulk creation for Epics with multiple scenarios
- Run Impact Analysis after major requirement updates or code changes

- Regularly export results to PDF for reviews, audits, or compliance
- Ensure team members refine and edit test cases before final creation
- Use Impact Analysis findings to prioritize testing (e.g., regression hotspots, high-risk areas)
- Take advantage of duplicate handling (the app prevents creating previously generated test cases)
- Incorporate generated cases into your existing manual test suite for broader coverage

3. Administration

The Test Case and Impact Analysis Generator by Clovity AI provides administrative options to control access, usage, and configuration.

3.1 Access

- The app is available inside Jira issues (Epic, Story, Task, Bug).
- Users must have permissions to create subtasks or linked tasks.
- Jira administrators can manage who has access to the app panel.

3.2 Capabilities

- Automatically generate test cases from issue descriptions and acceptance criteria
- Perform AI-powered impact analysis with risk assessments and mitigation suggestions
- Create subtasks or linked tasks from selected test cases
- Prevent duplicate creation of test cases through history tracking
- Export structured PDF reports for planning and audits

3.3 Storage and Tracking

- Uses Forge Storage API to track:
 - Already created test case titles

- Recently generated (but not yet created) test case suggestions
- Ensures duplicates are not created when regenerating or bulk creating

4. Tips & Best Practices

The following practices will help you get the best results from the Test Case and Impact Analysis Generator by Clovity AI:

4.1 Test Case Generation

- Always start with issues that have clear descriptions and acceptance criteria. AI outputs are most accurate when inputs are detailed.
- Use the Regenerate option if the initial suggestions do not fully match your testing needs.
- Review each generated test case before creating it in Jira to avoid unnecessary clutter.
- Use the Edit option to refine test cases so they match your team's format and standards.
- Use bulk creation for Epics with multiple scenarios to save time.

4.2 Impact Analysis

- Run Impact Analysis after major requirement changes or code updates to identify risks early.
- Use the Mitigations section of the analysis to inform development and QA planning.
- Prioritize testing based on Risk Assessment (High-risk areas should be tested first).
- Leverage Regression Hotspots to plan regression test suites more effectively.
- Apply Smoke Suite Recommendations before full regression runs to ensure system stability.

4.3 Collaboration & Reporting

- Export results to PDF for sprint planning, QA reviews, and compliance audits.
- Share analysis results with stakeholders (QA Leads, Developers, Product Owners) for better decision-making.
- Use exported test cases and impact reports as part of your test strategy documentation.

4.4 Efficiency & Quality

- Take advantage of duplicate handling to prevent creating the same test case multiple times.
- Combine AI-generated test cases with your manual test suite for broader coverage.
- Regularly review and refine generated test cases to maintain consistency with your QA standards.
- Use the app during sprint planning sessions to quickly convert requirements into actionable test coverage.
- Treat AI-generated outputs as a starting point, not a replacement, for expert QA judgment.

5. Limitations

The Test Case and Impact Analysis Generator by Clovity AI has the following known limitations:

5.1 Supported Issue Types

- The app works only with Epics, Stories, Tasks, and Bugs.
- It cannot be used on issues that are already defined as Test Cases.

5.2 Dependency on Jira Configuration

- Users must have the correct Jira permissions (create subtasks, linked issues, and edit issues) for the app to function.
- The app respects Jira workflows and field requirements — if certain fields are mandatory, they must be completed manually.

5.3 AI-Generated Content

- Test case and impact analysis quality depends heavily on the clarity of the Jira issue description and acceptance criteria.
- AI suggestions should be reviewed by QA engineers before being finalized.
- Generated outputs are intended as recommendations, not replacements for expert QA judgment.

5.4 Performance Usage

- The app relies on the OpenAI or test case generation and impact analysis.
- In rare cases, there may be delays or retries if AI calls fail due to connectivity issues.
- Large or complex Jira issues may take longer to process.

5.5 Storage & Duplicate Handling

- The app uses Forge Storage to track generated and created test cases.
- If storage is cleared or unavailable, duplicate handling may not work correctly.

Support & Contact

- **Email:** [support@clovity.com]
- **Website:** <https://clovity.com/contact>
- **Documentation:** Atlassian Developer Portal