

EXPANDING JIRA WITH HELIX ALM

Getting More Out of JIRA

Atlassian's JIRA was great for issue tracking when you were starting out, but now you may need more functionality than JIRA alone can deliver. How can you protect your JIRA investment and still gain the features you need? The answer is **Helix ALM**, with out-of-the-box JIRA integration.



Integrated Bug Tracking, Test Management, and Requirements Management



JIRA was designed and built for tracking defects. It focuses on tasks, issues, and defects, but there are no built-in test case management and requirements management features. You can buy add-ons from the JIRA marketplace, but they're not supported by Atlassian.



With **Helix ALM**, the entire development lifecycle is covered by one tightly integrated solution. Bug tracking, test case management, and requirements management are built into **Helix ALM**. As we develop new features, we consider each of these critical areas, ensuring everything works seamlessly without a drop in performance. **Helix ALM** even has a strong JIRA integration, allowing you to run tests in **Helix ALM** and log any failed tests directly in JIRA. You can also view the status of a JIRA item in **Helix ALM**.

Why Should I Care?

Helix ALM's strong integration allows you to track every development artifact, not just bugs. In turn, this helps you expose the true cause of project delays, reduce problems with communication and hand-offs, eliminate duplicated effort, and establish a repeatable process for better compliance.

Linking and Traceability



JIRA does have the concept of links, but its linking capabilities are not strong. JIRA also has no built-in traceability features; you must purchase a third-party add-on to achieve traceability.



With **Helix ALM**, the linking happens automatically as one item is created from another. Create a feature from a requirement, and the two are automatically linked. Create a test case from that feature, and now the test case is linked not only to the feature, but also back to the requirement.

Why Should I Care?

Helix ALM's dynamic linking capabilities allow you to establish traceability across the full development lifecycle—from requirements through test cases, test results, resolutions, and source code. Traceability is vital for ensuring product quality in any industry, but it's critical in regulated industries.

Traceability Matrix



A key traceability feature JIRA lacks is the ability to create a traceability matrix. There are add-ons you can purchase that will allow you to generate trace matrices, but it is not a built-in feature that comes with JIRA.



With **Helix ALM**, you can generate a trace matrix with just a few clicks, eliminating hours of manual effort. **Helix ALM's** traceability matrix lets you focus on test coverage analysis, root cause analysis, and other relationship-based analyses.

Why Should I Care?

Traceability matrices can show you information about all of the items you're tracking in **Helix ALM**. If you are linking items – for example, epics to features, higher-level requirements to lower-level requirements, or requirements with associated test cases and test results – you can now see all of that data in one report. You can even use trace matrices as failure modes and effects analysis (FMEA) reports, viewing features and their associated risks and mitigation.

Impact Analysis



JIRA has no backward or forward impact analysis capability. You must purchase an add-on.



Because of its strong traceability features, **Helix ALM** is able to easily perform both forward and backward impact analysis.

Why Should I Care?

You can use forward impact analysis to determine the child requirements and other dependent items that may be affected by requirement changes. For example, a change to a high-level business requirement may affect all child functional requirements or a change to a requirement may affect all test cases linked to it.

Backward impact analysis can be used to determine the parent requirements and other dependent items that may be affected by requirement changes. For example, a suggested change to a child requirement may conflict with its parent requirement or a feature request may affect the requirement created to address it.

Folders and Private Folders



JIRA has no folder or private folder concept, which can make it difficult to organize development artifacts.



With **Helix ALM's** folders, you can easily organize all of your development artifacts. A folder can contain any combination of issues, requirements, requirements documents, test cases, and test runs. An item can also be added to multiple folders, which adds flexibility when it comes time for reporting and what-if scenario planning. Folders can also be private or public.

Why Should I Care?

Folders let you group items together for any reason, whether it's to create time buckets for a release or sprint, or for organizing collections of items like test suites and feature requests. Public folders can be accessed by other users and are helpful for organizing items for your team. Access to public folders can be limited to specific security groups. Private folders cannot be seen or accessed by other users and are helpful for organizing items related to your own work. You can also use private folders to "bookmark" items in public folders that you want to find quickly.

Support for Hybrid Agile Teams



If your team is not purely Agile, it can be challenging to configure JIRA for your needs.



Helix ALM can be easily customized to suit your unique development methodology—from pure waterfall to pure Agile and anything in between.

Why Should I Care?

Helix ALM's flexibility means you can use it to manage every project, from the smallest to the most complex. Because of **Helix ALM's** strong linking and traceability features, you can even use Agile methods in a regulated environment and still maintain compliance.

Native Client



JIRA is completely web-based, requiring you to use a browser.



Helix ALM has a web client, as well as native clients for Linux, Mac, and Windows.

Why Should I Care?

Helix ALM's native clients handle large data better, requiring fewer hops, less network traffic, and no browser overhead.

JIRA Integration

JIRA integration is Helix ALM feature that's available "out-the-box," allows a user to associate a JIRA issue to a Helix ALM item. The details of the JIRA issue remain in JIRA. This integration is included with your Helix ALM license. It's easy to configure, too, so you can be up and running in no time.

With Helix ALM's JIRA integration, you can attach JIRA issues to Helix ALM items and open attached issues from Helix ALM. You can also create new JIRA items from Helix ALM and automatically attach them to Helix ALM items. If you work in JIRA, you can also use our free Helix ALM for JIRA add-on – available from the Atlassian store – to see your linked Helix

ALM items. With the add-on installed, you can simply click a link to work on your Helix ALM items without leaving JIRA.

With our JIRA integration and Helix ALM for JIRA add-on, users of both tools can work in the environment they prefer.

Helix ALM

Visit perforce.com/products/helix-alm for more information and to start your free trial today.

[START MY FREE TRIAL](#)