# Company-Wide Hierarchy of Requests, Requirements and Tasks at a Large Financial Institution

*Case Study – Structure Plugin for JIRA*

## The Customer

The customer is the IT department of a large multinational financial services company, responsible for internal software development. (Company name is left out in compliance with the customer's request.)

The development cycle is supported by several integrated solutions used for requirements management, software architecture and design, testing and quality management.

## Problem and Solution – Summary

With the introduction of Agile lifecycle model, the company needed a tool to support it and chose Atlassian JIRA and GreenHopper to do the job. However, JIRA and GreenHopper did not allow to conveniently organize and view the full hierarchical range of the company's book of work, which was important for strategic backlog prioritizing, program management over multiple teams, project tracking and reporting.

Structure plugin was chosen to provide this missing functionality and now the company uses several structures for Business, IT-Management, Project Teams and Quality Management.

## The Workflow

The company's development process is quite typical for organizations with an internal development department serving the company's needs. Business Partners add their Requests to a JIRA-based central request management system and they are prioritized in a joint effort between Business and IT.

As soon as a Request is scheduled, it is in the hands of the development teams. Each Request can be implemented by several Epics from different projects and these Epics are linked to the Request via Feature link type.

Epics are assigned to different teams and each team may be working on Epics from different projects (there are about 30-40 different projects). Each Epic is broken down into Stories, Tasks, and Sub-Tasks, and may have a number of related bugs.

> **Issue types**: Request, Epic, Story, Task, Sub-task, Requirement, Bug.
>
> **Issue links**: Feature (Epics are linked to Requests)
>
> **Statuses**: New, Analysed, Scheduled and other
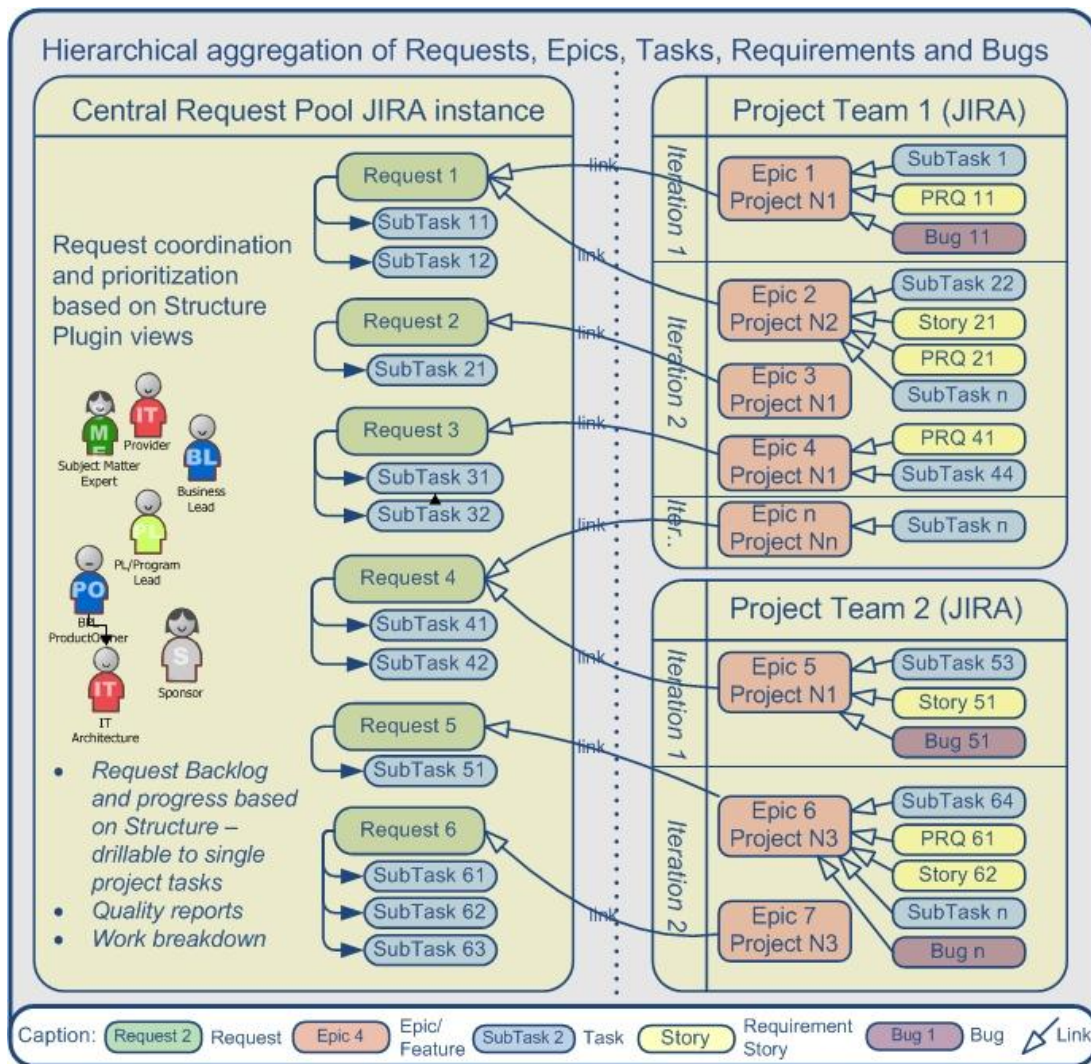
*Picture 1. Schematic representation of hierarchical relationships between different issues in Business and Project domains.*

## The Solution in Detail

The company created several structures to have the detailed overview of the whole set of JIRA issues.

The global structure lists all Requests with status New, Analysed and Scheduled with their related Epics, Tasks, Requirements and Bugs. This structure is mainly used for prioritization and progress overview.

Starting with the Requests, Business and IT-Management can now drill down to all related Epics and further on to various issues belonging to the Epics.

A structure (concept added by the Structure plugin) is a hierarchical list of issues.

Issues in a structure may be placed under other issues to form hierarchy, with no limits on nesting depth or issue projects or issue types.

The order of issues is also arbitrary and editable by the users.
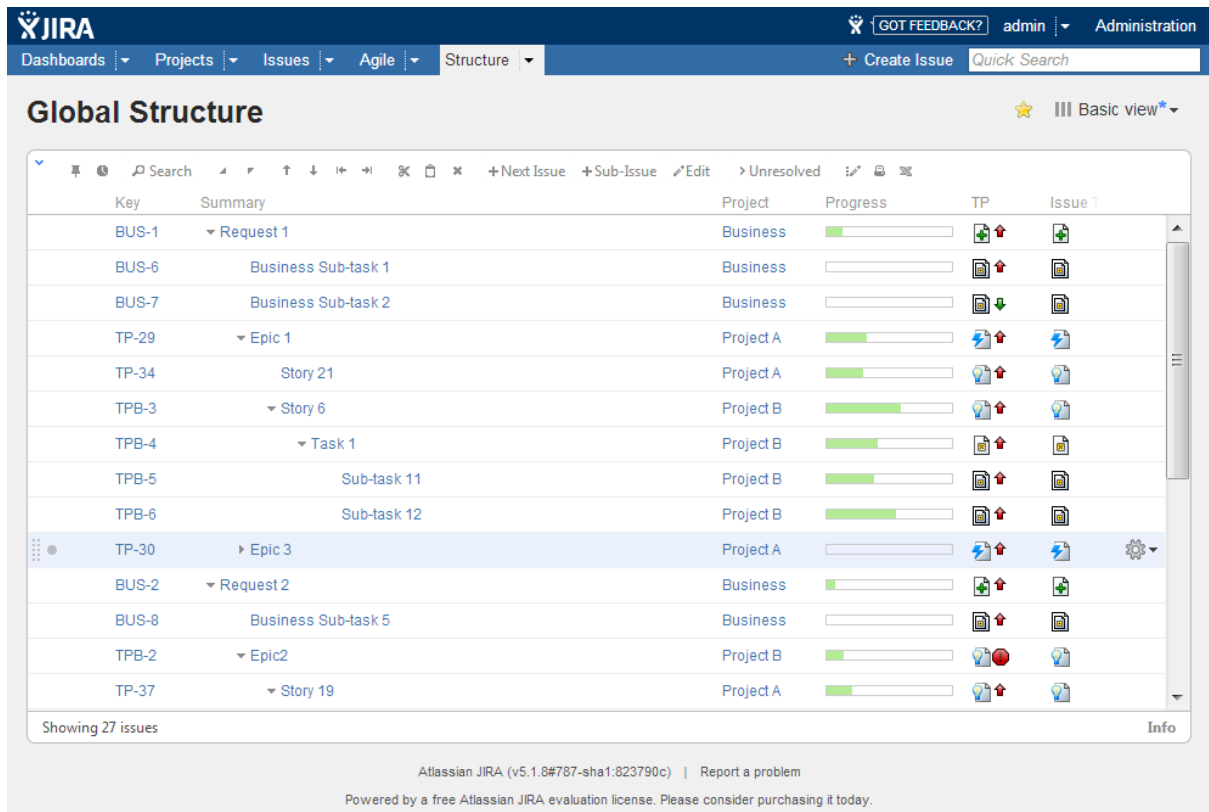
*Picture 2. Implementation of the schema from Picture 1 with Structure plugin. Note the aggregated progress and issues being combined from different projects.*

This structure has a number of synchronizers, which ensure that the structure is up to date.

- At the top level there is a filter synchronizer that adds to the structure all active Requests from the Request Pool.
- After that, a Sub-Tasks synchronizer adds Sub-Tasks related to Requests.
- The Epics are automatically added under the Requests by a Links synchronizer, when a link between the Request and the Epic is created.
- The GreenHopper synchronizer takes care of the Stories belonging to the Epics, updating their Epic/Theme field.

Synchronizers are special services, which let you keep Structure hierarchy in sync with some other issue properties.

For example, you can add a synchronizer that will make sure that JIRA sub-tasks are always located under their parent in the structure.

The resulting structure provides the full picture of issues existing in the company JIRA and allows users to quickly see the relationships between all issues and progress for each Request.

## Using Different Structures

Apart from the global structure, different departments have their own structures.

Quality Assurance team has a structure, which provides a single view for tracking the progress of all quality-related tasks. The structure is periodically exported into Excel for further usage in the QA process.

Project teams use structures to create risk views, WBS (work breakdown structures) and risk reports.

With custom integrations, developed in-house, the company integrated JIRA/Structure with other applications in the Tool Chain – SPARX Enterprise Architect and HP Quality Center. So now their teams always look at the same structures of Epics and Requests, be it in JIRA, EA or QC.

## In Conclusion

According to project manager, who had introduced Structure in the company, "the Structure Plugin was the missing piece for effective and efficient usage of JIRA. More and more our users rely on structures to navigate in their universes".