

Artifact Relationship Management

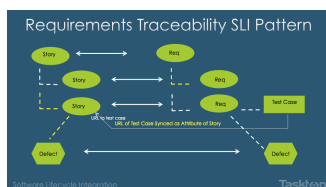
The tools that aid in software delivery tend to focus on a subset of the full scope of artifacts necessary for an effective SDLC. For instance, a requirements management tool will support a requirement artifact type but likely not a test case artifact. Or, a defect management tool will support defects but not an incident ticket artifact that was what caused the defect to be discovered.

But, regardless of the fact that the tool doesn't support a certain artifact, you still need to be able to answer questions like "have the testers written test cases to validate my requirement?" Or, "how many incident tickets have been logged related to this particular defect?". And with external associations you still can. External associations allows you to simply place a URL link to the necessary external artifacts. That way your practitioners can easily click the external link and launch directly into the tool that natively stores that artifact. One thing to keep in mind is that it is very viable and useful to actually combine this feature with the internal associations feature.

Key Benefits:

- Visibility into all aspects of the SDLC even when your tools don't support a certain artifact type
- Lightweight integration style for quick access to important artifacts *in the context of* the other artifacts that it relates to
- Increased flow of information between disparate artifact types

The picture depicts both internal and external associations with the following use case:



The development team is using Rally to develop and execute on their user stories and epics. There is a separate testing team that is responsible for testing at the end of each sprint as well as overall testing at the end of a release cycle - and that team is using HP ALM. In this scenario, the developers would like to sync their stories over to HP ALM - and maintain their epic to story relationships. Then, the testers can create their test cases directly in HP ALM – and the developers can have visibility by syncing an external association to the test case (yes, while Rally does support test cases, there was no need to have that more heavyweight style of integration at this point) in the stories. When the tester actually executes the test case, if he discovers a defect, he enters the defect directly into HP ALM – and then the defect is synced over to Rally so that the developers can work on fixing it. And, of course, the relationship from the defect back to the originating story is maintained in both systems.