

10.7 Bidirectional Review and Impact Analysis

[Go to Top](#)

[Traceability](#) supports review in two directions.

Forward [Traceability](#) starts with a conceptual concern and follows it outward through logical design, implementation mapping, deployment, and [Evidence](#). It helps reviewers confirm that the system implements and evidences the intended architecture.

Backward [Traceability](#) starts with [Evidence](#), deployment artefacts, implementation artefacts, or logical elements and traces back to the conceptual concern they support. It helps reviewers determine why an artefact exists and which architectural claim it supports.

Bidirectional [Traceability](#) supports impact analysis. When a concept changes, reviewers can identify affected logical elements, implementation artefacts, deployment artefacts, and [Evidence](#) packages. When an implementation artefact changes, reviewers can identify the logical and conceptual elements at risk. When [Evidence](#) is lacking, reviewers can identify the claim, deployment artefact, implementation artefact, and logical element that require review.

Examples of impact-analysis questions include:

- Which implementation artefacts realise this logical [Communication Endpoint](#)?
- Which deployed [Nodes](#) use this [Data Structure Definition](#)?
- Which [Evidence](#) items support this release-control claim?
- Which tests exercise this [Runtime Plane](#)?
- Which profiles inherit this [Classification Path](#)?
- Which artefacts require review if a [Communication Endpoint](#) changes?

From:
<https://wiki.didosolutions.com/> - Dido Solutions, Inc Wiki

Permanent link:
<https://wiki.didosolutions.com/fixdemo/01-part/10-conceptual-traceability-model/10-7-bidirectional-review-and-impact-analysis?rev=1782864631>

Last update: 2026/06/30 17:10

