

Technical Spike

From [Wikipedia...](#)

A spike in a sprint can be used in a number of ways:

- As a way to familiarize the team with new hardware or software
- To analyze a problem thoroughly and assist in properly dividing work among separate team members.
- Spike tests can also be used to mitigate future risk, and may uncover additional issues that have escaped notice.

A distinction can be made between technical spikes and functional spikes.

The technical spike is used more often for evaluating the impact new technology has on the current implementation. A functional spike is used to determine the interaction with a new feature or implementation.

[Engineering feasibility spikes](#) can also be conducted to de-risk an engagement and increase the team's understanding.

Deliverable

Generally the deliverable from a Technical Spike should be a document detailing what was evaluated and the outcome of that evaluation. The specifics contained in the document will vary, but there are some general principles that might be helpful.

- **Problem Statement/Goals:** Be sure to include a section that clearly

details why an evaluation is being done and what the outcome of this evaluation should be. This is helpful to ensure that the technical spike was productive and advanced the overall project in some way.

- **Make sure it is repeatable:** Detail the components used, installation instructions, configuration, etc. required to build the environment that was used for evaluation and testing. If any testing is performed, make sure to include the scripts, links to the applications, configuration options, etc. so that testing could be performed again.

There are many reasons that the evaluation environment may need to be rebuilt. For example:

- Another scenario needs to be tested.
- A new version of the technology has been released.
- The technology needs to be tested on a new platform.
- **Fact-Finding:** The goal of a spike should be fact-finding, not decision-making or recommendation. Ideally, the technology spike digs into a number of technical questions and gets answers so that the *broader project team* can then come back together and agree on an appropriate course forward.
- **Evidence:** Generally you will use sections to summarize the results of testing which do not include the potentially hundreds of detailed results, however, you should include all detailed testing results in an appendix or an attachment. Having full results detailed somewhere will help the team trust the results. In addition, data can be interpreted lots of different ways, and it may be necessary to go back to the original data for a new interpretation.
- **Organization:** The technical documentation can be lengthy. It is generally a good idea to organize sections with headers and include a table of

contents. Generally sections towards the beginning of the document should summarize data and use one or more appendices for more details.

Last update: July 22, 2021