

# IATI Software Deprecation Process

Last Updated: 21 Apr 2022

## Why deprecate a software tool or feature?

- Standardisation or increased consistency in naming
- The feature has been replaced by a more suitable alternative feature.
- The feature contains a design flaw, frequently a security flaw, and so should be avoided, but existing code depends upon it.
- The feature is considered extraneous, and will be removed in the future in order to simplify the system as a whole.
- A future version of the software will make major structural changes, making it impossible (or impractical) to support older features.

## What

... is deprecation?

- A software tool or feature is normally still accessible and usable (except in cases of serious security flaws that necessitate immediate removal) but no longer recommended for use or supported by the owning team. It may or may not be removed eventually from the codebase.

... should we deprecate?

- The IATI team has a responsibility to maintain software as defined in the following documents:
  - [2018 Technical Audit](#)
  - [2020 Technical Stocktake](#)
  - [IATI Core Software Non Functional Requirements](#)

These Non Functional Requirements (NFR) principles and the reasons **why** above should form the basis of the decision to deprecate IATI software.

## How will we deprecate a software tool or feature?

### Transparency and communication

Make it clear to the community that the IATI Core Software is available in GitHub:

- The consolidated IATI Core Software (non 3rd party hosted) can be found underneath the [IATI Organisation](#) on GitHub
  - This allows for continual community engagement and contribution
- Communicate to community members that they can fork repositories they would like to engage with after removal from the IATI Core Software, and ensure they add the "IATI" topic (<https://github.com/topics/iati>) for searchability

- Announce the plans before taking the above steps through the channels deemed appropriate by the communications team.

## IATI Deprecation Process

### Steps Overview

- 1) Identification
- 2) Community Research (conditional)
- 3) Planning
- 4) Deprecation Announcement
- 5) Deprecation Period
- 6) Removal Announcement
- 7) Removal

## 1. Identification

Identification of software to deprecate can happen through various processes:

- Organically as part of a new release
    - E.g. Feature B is being created and therefore Feature A will be obsolete and should be deprecated
  - When a new tool is being designed and implemented
    - E.g. Tool B is being created and therefore Tool A will be obsolete and should be deprecated
  - Periodic Review meeting
    - Proactive review of the software estate against our deprecation policies
    - Annually or Bi-annually, at the discretion of the IATI Technical Team
  - When an old version of the standard is no longer supported
    - Tools for the old version should be deprecated and removed
  - Serious flaws have been identified by the team or by the community that makes the deprecation plausible.
- See section on “Why Deprecate”... and “What... should we deprecate?”

## 2. Community Research (conditional)

In cases where we are unsure of the usefulness of a piece of software or tool and need to gather more information from the community before formally planning a deprecation a community research step will be performed.

Our goal in community research should be to understand what additional features or support may need to be provided or if timelines should be adjusted.

- Consider the number of users, how recently something has been used, who is using it, etc.
- Consider if we have a clear understanding of how the tool is being used and a good alternative

### 3. Planning

*"A developer that offers a meaningful and productive alternative will minimize user disruptions and improve customer satisfaction and retention. A deprecation plan should include one or more alternative paths for users -- if alternatives exist."*

*"Developers should plan and outline the deprecation process and clarify that plan to the user base. Such notice provides customers ample time to address the eventual deprecation, offer feedback to the developers and consider alternative features or functions to minimize disruptions."*

- [Source](#)

Once the decision has been made to deprecate software, the steps will be documented on this template:

- [IATI Software Deprecation Plan Template](#)

### 4. Deprecation Announcement

In cases where community research was not necessary this is the step at which the community is officially notified of IATI's intention to deprecate software.

The announcement should include a community facing plan from the Planning stage, which at a high level includes:

- Why
- What
- Timeline

### 5. Deprecation Period

During this period the development team would make the changes necessary for deprecation. This includes adding warning messages and indicators in the software itself about the deprecation (e.g. debugging notices). Deprecations should be released on minor versions in accordance with the [Software Versioning Protocol](#).

Examples:

- Message in the help text or documentation on deprecation and alternatives
- Response from an API including message about pending deprecation

Documentation and Support Guidance should be reviewed during this period as well.

Feedback from the community should be gathered during this period, and the IATI tech team may need to modify plans or do additional work to ensure the deprecation is smooth. Our goal in gathering feedback should be to understand what additional features or support may need to be provided or if timelines should be adjusted given new information, not to reverse course on the deprecation itself.

Examples:

- Modify the deprecation schedule
- Improve deprecation documentation or support
- Develop or make a commitment to develop an additional feature before removing code

## 6. Removal Announcement

There should be an additional announcement giving notice of a period of time before the code is fully removed. This should also have been communicated during the initial deprecation announcement, but may have been modified during the deprecation period.

## 7. Removal

In this step the team removes the deprecated software from the codebase and documentation. At this point the deprecated feature or tool would no longer function.

### **Removal timeline rules:**

- If we are removing an entire tool (and therefore versioning doesn't make sense), then provide at least 6 months for the **Deprecation Period** before removal if we are removing a feature in a versioned codebase, once the feature has been deprecated for 6 months in released code it should then be removed in the next major version upgrade in accordance with the [Software Versioning Protocol](#).

### **Exceptions:**

- In the case of security risks, the above removal timelines can be shortened as deemed appropriate by the IATI Technical Team.

## Deprecation Support Policy

During the deprecation period we will support existing users transition away from the deprecated software, but we won't release bug fixes/enhancements to the deprecated software.

## Documentation to Review

When deprecating a software tool or feature, the following list of documentation and support information should be reviewed.

- GitHub Repositories should be Archived and remain Public. Following GitHub's recommendations all Issues/PRs will be closed and a note added to the Readme.md
- Documentation posted on the software platform (e.g. Q&A pdfs or links to google docs)
- Search <https://iatistandard.org/en/> for all references of the software that is being deprecated. Change all references except in past news/events/meeting minutes.
- IATI Standard Publishing Guidance: <https://iatistandard.org/en/guidance/publishing-data/> and any publications that include this guidance.
- [IATI Speakers' Kit](#)
- Any external organisations who are known to provide IATI publishing/data access support through publications, workshops, consultancy etc.
  - E.g. Bond, UK govt, etc.

## Appendix

### References and Resources

- [Deprecation - MoodleDocs](#)
- [Deprecation guidelines](#)
- [How to deprecate software features without bothering users](#)
- [Open Contracting Deprecation Policy](#)
- [Open Contracting Deprecation Definition](#)
- [API Deprecation](#)