Versions

Get familiar with version policy and release roadmap

Tiki has a time-based, "Release early, Release often" approach with Long Term Support (LTS) versions, to balance the needs of stability and innovation.

Summary

- There is a new major version (e.g. 18.x -> 19.x) every 8 months.
- Minor versions (18.1 -> 18.2) are released as needed, typically every 1-2 months
- You can upgrade from any supported version to any subsequently supported version at any time. However, there is no downgrade path. Please make sure you make a backup first.
    - Some can have concerns about upgrade difficulty but this isn't the case because if you skip ahead several versions, you are going through the same path as everyone else did in the previous years, but in an accelerated fashion. You can upgrade from Tiki 1.9.0 (released on 2005-04-27) to any later version of Tiki because all the upgrade scripts are part of the code base. This policy will be maintained to keep upgrades as easy as possible.
- The support period for non-LTS versions is at least until the x.1 release of the next major version. For example, Tiki 13.x is supported at least until Tiki 14.1 is released.
- Every 3rd version is a Long Term Support (LTS) version.
    - LTS support period is 5 years (since Tiki 12.x LTS), and consists of a period of ongoing features support (bug fixes and minor, self-contained enhancements) followed by a "security fix only" period over this timeframe.
    - LTS is expected to have minimal changes, no regression bugs, and little or no changes to Tiki behavior and continued security fixes for an extended period of time. It's a good idea to install LTS versions on a similarly long-term supported server environment.
    - You should not expect LTS versions to be updated to solve an issue introduced by a newer PHP version that is released after. However, issues of this type have historically been very rare and a patch to resolve such an issue would be analyzed by the development team.
    - LTS is typically used for Enterprise deployments or by other users that do not want frequent major upgrades (for example, if your environment requires software audits before every software installation)
- Since Tiki has an all-in-one model, all features are released at the same time, and you don't need to wait for 3rd party extensions / plugins / modules to be updated (which is very common and often painful in other web applications). See also: http://pluginproblems.com/

Goals

- Have a predictable system
    - At download time, users should know until when the version is supported.
    - Time-based releases permit community members to self-organize to make sure specific

developments are ready in time for the release.
- A rapid release cycle
  - To take advantage of new technological innovations
  - Encourages contributions to the main code base as contributors see their enhancements rapidly become available as part of the next release, where they are tested, documented and improved.
- Caters to diverse needs, so the community converges efforts
  - Users that want to have early and frequent access to the latest and greatest, and experimental features.
  - Users that want a supported version (with security fixes), with minimal upgrade effort / change, typically, enterprise users.
- Easy to manage
  - Easy to explain and understand
  - Manage as few branches as possible

Why predictable and rapid are both important

Feature enhancements are often the fruit of sponsored feature requests, or for needs to deliver a specific project. Site managers or IT departments can do themselves or choose to hire Tiki Consultants. The vast majority of the time, the sponsor is happy or requires that enhancements become part of the official code base. However, the timeline has to be reasonable. By having a set date (which is not too far off), there is an incentive to commit to trunk (the development version which will eventually become the current stable version), and know when it will become stable. Otherwise, customer/project pressure will make them fork their version and the community will often lose those contributions. They might plan to upstream *later*, but the *later* never happens, or in the meantime, someone else implemented the feature in a different way, and it becomes very difficult to converge.

Here is a recap of innovations in past years:

- 2009-05: Tiki3 LTS
- 2009-11: Tiki4
- 2010-06: Tiki5
- 2010-11: Tiki6 LTS
- 2011-06: Tiki7
- 2011-11: Tiki8
- 2012-06: Tiki9 LTS
- 2012-12: Tiki10
- 2013-07: Tiki11
- 2013-11: Tiki12 LTS
- 2014-08: Tiki13
- 2015-05: Tiki14
- 2016-04: Tiki15 LTS
- 2016-11: Tiki16
- 2017-07: Tiki17
- 2018-01: Tiki18 LTS
- 2018-11: Tiki19
- 2019-06: Tiki20

The plan ahead:

- 2020-02: Tiki21 LTS
- 2020-10: Tiki22
- 2021-06: Tiki23
- 2022-02: Tiki24 LTS
- 2022-10: Tiki25

Not only Tiki has been for many years the FLOSS Web Application with the most built-in features, Tiki is also the one of the FLOSS Web Applications with the fastest release cycles.

There is a huge part of our user base that does not want/need new features, but only bug fixes and security fixes. This is why Long Term Support (LTS) versions were introduced in 2009. The support period was extended in 2012 to 3.5 years (applicable to 6.x LTS and 9.x LTS), and extended once more in 2013, to 5 years (applicable for 12.x LTS, 15.x LTS and 18.x LTS). Even if they don't need the new features now, site admins are pleased to know that they will have easy access to them when they upgrade.

The cycle

- A new major version every 8 months (February, October, June, etc.)
- Minor versions as needed, usually every 6-8 weeks
- Every 3rd major version is a Long Term Support (LTS) version (Tiki 9.x, 12.x, 15.x, 18.x, 21.x, ...)
    - So the LTS cycle is 2 years (3 x 8 months = 24 months)
- Support period
    - LTS versions are supported for 5 years (starting from Tiki12)
    - Non-LTS versions are supported until the x.1 release of the next major version (exceptions might be made to delay this to the x.2 version if the x.1 version is released very close to x.0).

Version Lifecycle

Version	Release of x.0	Regular Feature Support (Security Fixes only thereafter)	End of Life
12.x LTS	2013-11	at least until 2016-04	2018-11
13.x	2014-08	until End of Life	2015-11
14.x	2015-05	until End of Life	not before the release of Tiki 15.1
15.x LTS	2016-04	at least until 2018-04	2021-04
16.x	2016-10	until End of Life	not before the release of Tiki 17.1
17.x	2017-07	until End of Life	not before the release of Tiki 18.1

Version	Release of x.0	Regular Feature Support (Security Fixes only thereafter)	End of Life
18.x LTS	2018-01	at least until 2021-01	2023-01
19.x	2018-11	until End of Life	not before the release of Tiki 20.1
20.x	2019-06	until End of Life	not before the release of Tiki 21.1

- Development phase in trunk (8 months)
- B: Stabilization phase, from Branch creation (forking of trunk) to initial release. About 6 weeks to 2 months before the release.
- R: Regular support period with bug fixes and minor, self-contained enhancements (normally 8 months; 16 months for LTS versions)
- Q: All developers commit to the branch (as usual) (12 months)
- S: Security fixes only
- E: End-of-Life (EOL)

A visual chart of the support period

```
Year:   |   2013    |    2014   |    2015   |    2016   |    2017   |    2018   |    2019
|    2020   |    2021   |    2022   |
Month:
JFMAMJJASONDJFMAMJJASONDJFMAMJJASONDJFMAMJJASONDJFMAMJJASONDJFMAMJJASONDJFMAMJJASONDJFMAMJJA
SONDJFMAMJJASONDJFMAMJJASONDJ
        |     |     |     |     |     |     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |
9.x     ...RQQQQQQQQQQQQSSSSSSSSSSSSSSSSSSSE |     |     |     |     |     |     |     |
|     |     |     |     |     |     |
10.x    ...RE |     |     |     |     |     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |
11.x    ...RRRRRRRE |     |     |     |     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |
12.x    | DDDDDDBBRRRRRRRRRRRRQQQQQQQQQQQQSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSE |     |
|     |     |     |     |     |     |
```

Strategies

Site admins have many choices.

Main concern	Reason	Strategy
I'll be developing new features	Tiki has most of what I need, but I also want to add X	Develop in trunk. Your features will be in the stable/official Tiki version within months, and it will become community supported. Please see how to get commit access
I want the new features	Because they are cool!	Upgrade with each new stable version . Depending on your eagerness vs tolerance to bugs (capacity to fix them), you may want to skip x.0 versions
I want the least possible number of bugs	I have a huge user-base and they are grumpy.	Go from one LTS to the next starting from midway in the series (e.g. starting from x.3 or x.4). Since the version you are using will have been more in use, bugs will be rarer. You should at least test (and if possible deploy) before the security-only period so you may detect any bugs, report them (and ideally, you arrange to get them fixed)
I want to upgrade less often. Once I deploy, I want minimal changes	A heavily customized theme or need to audit the code	Use an LTS version, and just before it reaches end of life, upgrade to the latest LTS.
I want the best possible security	My site is a potential target	This is a tricky one, and it depends on your use case (type of site, features, etc.) Could be: 1- Stay on latest stable version all the time, as it will have all the possible fixes and enhancements to security. 2- Move from one LTS to another (ex.: 9.x to 12.x) but start using when it is in security-fix-only phase. Since the version you are using will be over 2 years old, most vulnerabilities have been found and resolved. But you won't have access to the latest security features. 3- Another option is to stay on latest LTS version all the time. You will have more security features, but you may experience more vulnerabilities (the ones that are introduced in one version but fixed in the next).

If you are using Tiki LTS, you should consider deploying on an LTS server, such as Ubuntu LTS or CentOS (which is LTS in nature). Or go for WikiSuite and pick ClearOS

Upgrade paths examples

Use case	Upgrade path
Eager for new features	12.0 -> 12.1 -> 12.2 -> 12.x... -> 13.0 -> 13.1 -> 13.2-> 13.x... -> 14.0 -> 14.1 -> 14.2 -> 14.x...-> 15.0 -> 15.1 -> 15.2 -> 15.x... -> 16.0 -> 16.1 -> 16.2 -> 16.x...
Prudent (skip x.0 releases)	12.1 -> 12.2 -> 12.x...-> 13.1 -> 13.2 -> 13.x... -> 14.1 -> 14.2 -> 14.x...-> 15.1 -> 15.2 -> 15.x... -> 16.1 -> 16.2 -> 16.x...
LTS	12.2 -> 12.3 -> 12.x... -> 15.2 -> 15.3 -> 15.x... -> 18.2 -> 18.3 -> 18.x...

PHP versions

Thanks to the availability of LTS versions, Tiki can be aggressive in increasing the requirements and take advantage of new possibilities of PHP. Indeed, if your infrastructure is not evolving as quickly (which is typical in large enterprise settings), you can use the Long Term Support (LTS) versions of Tiki. Please see: https://doc.tiki.org/Requirements#PHP

Browser support considerations

For desktop browsers, Tiki generally supports the current and previous stable versions. Firefox and Chrome have a rapid release cycle with auto-update and for Safari and Opera, users upgrade fairly quickly. So except for Internet Explorer and mobile browsers, we only have to worry about fairly recent browsers. For Internet Explorer, because upgrades are decided by the IT department or limited by the OS version, a significant portion of the user base doesn't use the latest version. Balancing the needs of innovation and stability, the Long Term Support (LTS) versions are perfect for Tiki to take advantage of all the new possibilities of newer browsers, while still offering a supported version for users that need to support older browsers.

Tiki version	Internet Explorer & mobile browser version
6LTS	IE6+
7, 8, 9LTS	IE7+
10, 11, 12LTS	IE8+ , which is the highest you can upgrade to if you have Win XP

Tiki version	Internet Explorer & mobile browser version
13, 14, 15LTS, 16	IE9+: see the discussion and stats. Note: Since IE11 has now been released, it is possible that support will be for IE10 and IE11 only, depending on how it goes with the Bootstrap integration.
17-18	IE 9+, Mobile Android 4+, Mobile Safari 7+
19+	IE 10 and below not officially supported. IE 10 note

The Future

As of 2018-11, the release cycle is not expected to change for the foreseeable future. However, if it does change, the same general principle will stay the same. And the total support period (as announced when you download) will not be reduced (but could be extended). Any eventual (and at this point, unlikely) change in policy will be amply discussed within the community before it happens. Potential changes are:

- A faster cycle speed: ex.: every 4 months in February, June, October, February, etc. or every 6 months in October and April. It could even be a combination (ex: 4 months, 4 months, and then 8 months). This would be while maintaining LTS versions for approximately the same periods. So instead of an LTS every 3rd version (18 months), it would be every 4th (16 months) or 5th version (20 months).
- Alignment with larger ecosystem: Ubuntu's Mark Shuttleworth once wrote: "There’s one thing that could convince me to change the date of the next Ubuntu LTS: the opportunity to collaborate with the other, large distributions on a coordinated major / minor release cycle.". The Tiki community would certainly participate in any similar initiative.

Related links

- http://php.net/supported-versions.php