Choosing Ubuntu LTS or Going with the 6 Month Update Cycle

As NatickFOSS prepares for an InstallFest later this fall, it may be wise for us to understand the needs of the people who will visit and ask us to help them become FOSS users. One of the issues is the amount of ongoing support that someone may need.

Even though NatickFOSS is a strong supporter of Ubuntu, there is still a choice to be made.
For users who are brand new to FOSS, the Long Term Support (LTS) version (12-04) may be more appropriate, especially if the person does not expect to become a participant in NatickFOSS meetings.

What are the differences between the LTS and 6-month releases?

LTS = Long Term Support

Conservative feature set (not "bleeding edge" software)
    Fewer regressions (greater amount of testing before adding a patched software version)
Supports new hardware
New LTS release every two years
Five year support for both desktop and server LTS releases

Six Month (non LTS) releases

Faster implementation of features than LTS
Greater potential for regressions
Stability not as well tested – software may not be "beta" officially, but it may have features which are new enough that they are not validated with all the available hardware or edge case conditions
Still may not have the latest "bleeding edge" which will require backport selection or even building from source.
Supports new hardware
The six-month releases provide real world testing that identifies unexpected, edge case problems. Six-month releases are not commonly considered enterprise ready.
Eighteen months of support – it is assumed that a person who is eager to keep up to date will probably not stay with an older six-month version for too long.
Every two years, the April six-month release is a long term support release built based on the experience with the prior. Since the 12-04 release was LTS, the 13-04 will not be LTS. We should expect the 14-04 version of Ubuntu to be the next LTS release.

Should we offer only the Unity Ubuntu or be ready to offer Kubuntu (KDE) and perhaps even Fedora, Open SUSE or others. What is our commitment to support?

Should we have a menu of options printed or posted?
What options are we recommending?
Dual Boot?
PC only or also Macintosh installs?

Volunteers?

Natick Foss
Natick FOSS
MetroWest Area
Free Open Source
Software Users

Sources

https://wiki.ubuntu.com/LTS
http://en.wikipedia.org/wiki/Software_regression

Glossary

backport – A volunteer packager has pulled the necessary dependencies together for an older release – the backport is often not directly supported by the official packaging teams and testers of Ubuntu/Canonical

binary code – When a program is compiled, the code is converted to the instruction set of a computer processor (CPU). Binary code is a series of 0/1 codes which computers run efficiently, but which are very difficult to interpret by humans. (opposite of source code)

bleeding edge – A light-hearted way to refer to software that has very new features, some of which are incomplete and work only part of the time or with a limited selection of hardware. Bleeding edge software is not recommended for systems that need stability first.

building from source – A manual task of gathering the libraries and the program source code files and compiling a working version of a program. Allows access to the "latest and greatest" version of a program

dependencies – A program/app needs the appropriate library versions packaged to run correctly.

edge case – Software is tested first on a developer's computer, then on the computers of the development team, then on the computers of beta testing volunteers. Finally it is used by people in a general release. Sometimes odd effects are not noticed because the typical user doesn't explore an odd combination of steps. These odd or unusual situations may produce unexpected results or a program crash. Testing for edge cases is difficult. The six-month releases often identify edge case issues so they can be patched before the program is added to the LTS releases.

enterprise ready – Software which has been carefully tested so it can be installed and relied upon for use by a typical user and which will NOT require regular attention by a professional technician from the IT department (company's computer staff)

libraries – A support program that other applications depend on for specific functions (ex: the dictionary tool used for word processor spell checking)

patch - If a change is made to the code of a program, it is referred to as a patch. Patches may fix an existing problem or add a new feature.

package – A volunteer gathers all the dependencies for a program together and sets up the pieces to be installed so that updated programs will properly function. A package maintainer makes it easier for the rest of us by doing the work that would be needed to do a build from source.

regression - When a change in a program relies on a new feature in libraries, the program will no longer work with older versions of the libraries. Backports and patches sometimes cause regressions.

source (source code) – The core of FOSS, source code is the human-readable version of a program before it is built/compiled into the binary instructions for a particular kind of computer.

Natick Foss

Natick FOSS
MetroWest Area
Free Open Source
Software Users