

KEP: 1 Title: Bootstrapping the KEP process Version: \$Revision\$ Last-Modified: \$Date\$ Author: Georg Greve <greve@kolabsys.com> Status: Draft Type: Process Content-Type: text/x-rst Created: 2010-11-16

Abstract

KEP stands for Kolab Enhancement Proposal. A KEP is modeled closely after the Python Enhancement Proposal (PEP) [1] process. A Kolab Enhancement Proposal is a design document providing information to the Kolab community, or describing a new feature for the Kolab Groupware Solution or its processes or environment. The KEP should provide a concise technical specification of the feature and a rationale for the feature.

Some of the most important benefits of the KEPs are to (a) require ideas to be fully thought out, (b) communicate a change to those who did not participate in the discussion, (c) document it for future reference and use by new members of our community. We intend KEPs to be the preferred mechanisms for documenting the design decisions that have gone into Kolab after its re-launch in 2010 as well as proposing new features, for collecting community input on an issue. The KEP author is responsible for building consensus within the community and documenting dissenting opinions.

Because the KEPs are maintained as text files in a versioned repository, their revision history is the historical record of the feature proposal. [2]

KEP Types

There are three kinds of KEP:

1. A **Design** KEP describes a change to the Kolab Storage Format or other central design questions.
1. A **Technology** KEP describes a change to the integrated technologies.
2. An **Informational** KEP describes a Kolab design issue, or provides general guidelines or information to the Kolab community, but does not propose a new feature. Informational KEPs do not necessarily represent a Kolab community consensus or recommendation, so users and implementors are free to ignore Informational KEPs or follow their advice.
3. A **Process** KEP describes a process surrounding Kolab, or proposes a change to (or an event in) a process. Process KEPs are like Format KEPs but apply to areas other than the Kolab Groupware Solution itself. They may propose an implementation, but not to Kolab's codebase; they often require community consensus; unlike Informational KEPs, they are more than recommendations, and users are typically not free to ignore them. Examples include procedures, guidelines, changes to the decision-making process, and changes to the tools or environment used in Kolab development. Any meta-KEP is also considered a Process KEP.

KEP Process

The KEP Process is modeled after the PEP process, including the forms of markup [3][4] and headers, with Kolab Systems providing the moderator and reference point. The authoritative address to send KEPs is kolab-format@kolab.org when changes to the format are being proposed and kolab-devel@kolab.org for everything else.

We expect that we will need to evolve our processes away from the PEP process as we employ a different development methodology and have a different community. But starting from the established PEP process should at least provide us with a sane default.

References

Copyright

This document has been placed in the public domain.

-
- 1 PEP 1, PEP Purpose and Guidelines, Warsaw, Hylton
(<http://www.python.org/dev/peps/pep-0001>)
 - 2 This historical record is available in the Kolab GIT repository provided by Kolab Systems. For those without direct access to the repository, the information is available via GIT web access here: (.tbd.)
 - 3 PEP 9, Sample Plaintext PEP Template, Warsaw
(<http://www.python.org/dev/peps/pep-0009>)
 - 4 PEP 12, Sample reStructuredText PEP Template, Goodger, Warsaw
(<http://www.python.org/dev/peps/pep-0012>)