

Project Timelord	2
Rationale.....	2
Summary	2
Parts.....	2
Path	2
Kubuntu's Issues	3
Translations.....	3
Quality	3
Maintainability of import system(s)	4
Marketing	4
Branding	4
Promotion.....	4
News publication.....	4
Publications in magazines.....	4
Presence at conferences and fairs	4
Product Management	4
Vision	4
Mission	5
Roadmap	5
Software	5
Packaging Issues	5
Integration of Kubuntu applications	5
Bug triage	6
Bugfix deployment.....	6
Feature parity with Ubuntu.....	6
User <=> Developer interaction	6
Recruiting.....	7
Support.....	7
Development	7
Promotion.....	7
Artwork	7
.....	7
Action plan.....	7
Find/propose solutions	7
Translations	7
Marketing	8
Branding.....	8
Promotion	9
News publication	9
Publications in magazines.....	9
Presence at conferences and fairs.....	9
Product Management.....	9
Vision	9
Mission	9
Roadmap	9
Software	10
User <=> Developer Interaction	11
Recruitment.....	12
Call all hands	12
Set the record straight (Roadmap of final solutions)	12

Project Timelord

Rationale

Get a solid base to work on (community and code-wise) that will last the lifespan of a timelord (counting regenerations).

Summary

Relaunch Kubuntu "on-the-fly". This is probably a lot of work, but generally preferably over doing a real relaunch. Also it bares the danger of sudden death due to issue overload and thus enforcing a hard reset after all.

should further feature deployments be on hold unless necessary (foundations enforcing them etc.)? -Harald Sitter 9/29/09 5:49 PM

I would support putting them on hold, unless they're necessary for less-alien Kubuntu application integration with KDE (e.g. KAuth support for our KCMs) But with 10.04 being an LTS, this would be a good excuse to not do too much. -Jonathan Thomas 9/30/09 4:06 PM -Jonathan Thomas 10/1/09 8:47 PM

AFAIK the only KCM that is specific to Kubuntu is userconfig, so yes, that should receive kauth support. All others come from upstream, so that ought to be handled by them -Harald Sitter 10/1/09 9:09 AM

Parts

1. Find most important issues and write them down
2. Assign each issue a priority
3. Call all hands with a sensible plan to work on
4. Resolve issues one by one
 - Requires long-term working solutions. It should be possible to move on to the next issue without spending any, or at least very little, time on already resolved ones => reign of solid base
 - Breakage needs to be resolved at a sensible time before each release to not compromise the schedule. If this can not be archived the release must be rescheduled or skipped completely. No release > crap
5. Create task force to ensure things get done way beyond project Timelord

Path order needs to be discussed -Harald Sitter 9/30/09 6:03 PM

1. Establish upstream link
 - Ask what upstream expects Kubuntu to do and be(come), where they think Kubuntu provides a inferior KDE implementation and what they like about Kubuntu. In general get an idea what upstream thinks.
2. Include upstream in Timelord actions

- Where necessary and useful, include upstream in the decision making and/ or execution (considering they are willing to). Poke upstream into things we need to create a better product.
- Possibly get someone to become upstream interaction node
- 3. Establish long-term solution for bug triage and other intrinsic policy
- 4. Re-evaluate ALL patches
 - Upstream-able to Debian or KDE?
 - Necessary?
 - Sane?
 - Working 100% properly?
- 5. Fix translations
- 6. Find a way to assure quality
- 7. Re-evaluate support policies
 - Support level of all deployments
 - Support time (we are not able to deploy fixes for 4 releases)
 - Write them down
- 8. Re-evaluate ALL default applications for ALL purposes
 - Keeping in mind: size, functionality and localization
- 9. Polish Kubuntu applications We should probably prioritize the integration features of the below two items (9 and 10) and span them across several releases if need be. I can't see us rewriting all of our tools to perfection in a mere 6 months. How much of this can we accomplish during an LTS cycle anyways? -Jonathan Thomas 10/1/09 8:45 PM
 - Complete feature assessment
 - Triage all bugs and resolve them
 - Make them follow KDE HIG What KDE HIG? -scott@kitterman.com 10/8/09 6:28 PM
 - <http://techbase.kde.org/Projects/Usability/HIG> -Harald Sitter 10/10/09 7:39 PM
 - If necessary revise UI to be more usable Really needs to be done with/by upstream -scott@kitterman.com 10/8/09 6:29 PM
 - Why should upstream care about the usability of our Kubuntu-specific tools? It should be pretty easy to tell where our tools fail at being usable. -Jonathan Thomas 10/8/09 6:49 PM
- 10. Integrate Kubuntu applications into the surrounding system
 - Tighter integration with existing apps/technologies
 - Avoid use of 'start foobar' buttons This is particularly important for netbook - scott@kitterman.com 10/8/09 6:29 PM

Kubuntu's Issues

Translations

Quality

- If Rosetta imports brokenness, it will remain broken
- Lang-packs can be out-of-sync with installed/deployed KDE
- There are too many ways for perfectly good upstream translations to break after being imported, whether by well-meaning GNOME translators making mistakes

(such as modifying variable names in translations) or anywhere in the export chain from Rosetta to lang-o-matic

- Rosetta UI makes it difficult for developers to trace such issues when they are caught
- Kubuntu translators are nonexistent, which results in Kubuntu patches that modify strings never being translated except in to one or two languages. (Defeating the purpose of Rosetta as a way to be able to modify our strings)
- Lower level of QA compared to KDE due to the above points, especially the lack of a translation team

Maintainability of import system(s)

- Always one step behind KDE
- System needs regular updates and love to keep it somewhat up-to-date with what KDE has
- Ongoing issues with integration
- Creates maintenance overhead if po(t) generation/import fails
- Increases diff to Debian since the gettext domain needs to be defined in debian/rules.

Marketing

Branding

- Close to no branding inside Kubuntu

Promotion

News publication

- One load of geek information not useful to the majority of users

Publications in magazines

- Missing

Presence at conferences and fairs

- Close to none
 - could be way more "professional" also -Lydia Pintscher 9/29/09 9:40 PM
- No promotional work done unless done by the community (e.g. kubuntu-de)

Product Management

Vision

- Missing

Mission

- Missing

Roadmap

- Missing
 - quote by drupal founder: "roadmaps are bad - they kill innovation" ;-) - Lydia Pintscher 9/29/09 9:41 PM
 - I do not think that we have resources to implement innovation, besides, this is a roadmap as part of product management not a roadmap as part of release management. The diff is quite big TBH since the former maps the development of a product to achieve more market penetration et al while the latter maps the development of a product in context of a specific release -Harald Sitter 9/30/09 5:58 PM
 - Our roadmap is pretty well inherently some intersection of KDE's vision and Ubuntu's. No need to plan it all out too much, just figure out how they mesh (and work to sync them where they don't) -scott@kitterman.com 10/8/09 6:30 PM
 - Obviously we don't need to plan out when we include the next version of KDE or Xorg, but we have more things that need done with our distro-specific tools than we can address in one release. It would make sense to prioritize and spread out the work to the Kubuntu-specific apps we ship. - Jonathan Thomas 10/8/09 6:54 PM
 - It is also about when switching apps to what, to prepare a migration path, and when to include which apps in the default install, to work out issues,... there are plenty of things that need to be well prepared and planned, other than inclusion of KDE, where we rely on what upstream pushes anyway - Harald Sitter 10/10/09 7:39 PM

Software

Packaging Issues

- Too little QA done on in-dev-release-uploads
- Too little QA done on kubuntu-backports/updates
- Implementation issues (kdesudo vs. kdesu via dpkg-divert)
 - Didn't this get fixed for 9.10? -Jonathan Thomas 10/1/09 11:03 PM
 - Yes, we need some way (policy?) to prevent such things from happen _ever_ again. Like you need approval from at least 2 members of kubuntu-dev to go ahead with dpkg-divert or similar complex stuff -Harald Sitter 10/10/09 7:41 PM
- Missing features due to main vs. universe
- Patches are evil
- We have had too much pre-release shizz in the past

Integration of Kubuntu applications

- Incredibly alien integration (language-selector in locale KCM, software-sources in Kpk)

- Overuse of python applications for notification purposes. They use way to much RAM for what they do (printer-applet, update-notifier-kde)
- Too much stuff that we maintain that could be handled by other upstream apps

Bug triage

- Not nice enough
- Not helpful enough
- Not fast enough
- Too many bugs
- Too few triagers (possibly causing the above)
 - Definitely causing the above, from my experience. -Jonathan Thomas 9/29/09 3:41 PM

Bugfix deployment

- Not enough SRUs
- Difficult to SRU due to meta-package structure of KDE
- Too few developers to do SRUs at large scale

Feature parity with Ubuntu

- We lag behind in implementing features that Ubuntu implements, if we ever implement them at all.

User <=> Developer interaction

- Users do not know whether to use LP, upstream, or Brainstorm
 - Whacking out policy for this shouldn't be too hard, but how do we get the info out to the user before they become confused? -Jonathan Thomas 10/2/09 7:32 PM
 - Well, as I see it we only have one sensible option: spread as much info about the policy as possible (prominent section on kubuntu.org et al). Also communicate it very well to those people actively working on forums and other means of support to guide users in the right direction. -Harald Sitter 10/10/09 7:43 PM
 - I have yet to find anything interesting on Brainstorm. -scott@kitterman.com 10/8/09 6:32 PM
 - Yes brainstrom is useless, the user doesnt know that though, so he posts stuff or votes there and wonders why we never do anything -Harald Sitter 10/10/09 7:42 PM
- LP is quite the complex beast to new users
- Developers do not follow resources used by users (most importantly forums)
- Developers do not follow news and blogs closely enough to detect demands and issues early on

Recruiting

Support

- No recruiting done

Development

- Too few new developers (junior jobs)
- Too few upstream developers (large scale feature implementation)
- Not enough bug triagers

Promotion

- No recruiting done

Artwork

- No recruiting done

not one of the big problems atm i'd say - the ones above are much worse and causing more harm -Lydia Pintscher 9/29/09 9:47 PM

How much recruiting needs done anyway? A little might need to be done to help with branding, but I agree that it's not as high-priority as other issues. -Jonathan Thomas 9/29/09 5:05 PM

in need of a solution nonetheless, also, there is more to artwork than software alone, if we ever want to get stronger press coverage we need to work on the public image, artwork in press releases et all is one important part of that -Harald Sitter 9/30/09 5:28 AM

Action plan

Find/propose solutions

Translations

- Until we get our act together (proper translation personnel within Kubuntu), we should not use Rosetta, but ship upstream translations instead. This would mean foregoing the privilege of introducing Kubuntu-specific strings, but it has been made very evident that our strings do not get translated, defeating the whole point of using Rosetta in the first place. Using Rosetta to handle upstream translations only results in one more layer of things that can and will go wrong. Solid translations and a Kubuntu translations community are more important than changing several instances of "abort" to "cancel" one cycle earlier than KDE. I surprise myself when I say this, but I think we ought to wait and see how Karmic turns out. We're already

better and if we skip out of Rosetta, that's more work for us we don't have people to do. -scott@kitterman.com 10/8/09 6:33 PM

- The thing is, being better than we were isn't good enough. As long as we are not at least at the same level as KDE, people will (justifiably) look down on our translations, and upstream will still not appreciate that they receive bugs for improper or missing translations that are our fault. We don't even have the resources to know whether translations in Karmic are OK beyond the one or two other languages a handful of Kubuntu developers use, and the oh-so-few bug reports about translations we receive. For all we know, a language that's not German or Spanish could have Ubuntu-translator-inflicted errors. Using KDE's translations ensures that we at least get a KDE-level QA on the translations. Not using Rosetta wouldn't be that labor-intensive either. We just have to tell pkgbinarymangler to not strip translations from the KDE packages, (perhaps check the cdb rule used?) making kde-l10n-* and KDE Main apps not using kde-l10n-* unstripped of text translations, and getting language-selector to install kde-l10n-*, which it should already be doing since they contain localized KDE sounds and graphics. For our apps, we can export what is currently in rosetta and make po/ folders in the bzd branches they are developed in. At the least we know how to do the work for ditching Rosetta, whereas we don't know what will need to be done to keep Rosetta working in a given cycle. -Jonathan Thomas 10/8/09 6:39 PM
- On the web, KDE translators and former Kubuntu translators have made it quite clear to me that they will never do Kubuntu translations while it still uses Launchpad due to the crappy interface and breakage. Our problems run deeper than breaking upstream's translations. The lack of a translations community has resulted in Kubuntu-specific applications never having good internationalization. This won't change until we get translator support. Having people to translate our apps is perhaps just as important as not breaking upstream's translations, since it takes very little to . Rosetta is just a hinderence to all of this, even if it doesn't break upstream translations quite so bad any more. -Jonathan Thomas 10/8/09 6:39 PM

Marketing

Branding

- For branding, perhaps monochrome Kubuntu logos could be thrown in with the Air Plasma theme's background overlay. (The circles in the background of Plasma widgets)
 - I wonder how that would look, worth a try I suppose? -Harald Sitter 9/29/09 6:57 PM
- We should get a clear guideline on how much branding we want to apply -Harald Sitter
 - I personally find it quite stupid to not replace the most visible appearances of KDE with Kubuntu, the brand mixture only creates confusion IMHO - Harald Sitter 9/29/09 6:58 PM
 - In any case, all branding changes to the default themes should be discussed with upstream. (As long as we are modifying the default theme and not

making an entirely new theme, but I don't think we want to diverge that far from KDE) -Jonathan Thomas 9/29/09 4:37 PM

- In extend it might be useful to brand all and everything according to the guideline(s). Kubuntu branding was always a bit incomplete and inconsistent (in KDE 3 times the amarok/konqueror sidebar themes were only appearing one by one and by the time they were used, they didn't fit in with the other branding efforts). So in the end we should have as much branding as possible and desirable (by possible I mean how much we can actually maintain). -Harald Sitter 10/10/09 7:45 PM

Promotion

News publication

- Create templates that can be reused
- Carefully review which information you expose in what way (gpg GUI vs. CLI vs. not at all)

Publications in magazines

- Richard to do something about this ;)
- There should be some unified resource for information that could technically be used to write stuff for magazines (right now we have wiki vs. help.ubuntu vs. blogs vs. kubuntu.org vs. void)

Presence at conferences and fairs

- Possibly the only real solution to this is to enforce Kubuntu LoCos. Since we cannot finance travel et all, we depend on available presence
- Provide the means for LoCos to present Kubuntu (Kubuntu presentation templates and raw information presentations for example)

Product Management

Vision

- Create one
- Write it down
- Get it approved by the Council AND the community as a whole

Mission

- Create one
- Write it down
- Get it approved by the Council AND the community as a whole

Roadmap

- Create one
- Write it down

- Get it approved by the Council AND the community as a whole

Software

- Run automated package updates to detect file overwrite errors in upgrades
 - Need to run multiple times because package installation order can affect it and it varies each time you do it -scott@kitterman.com 10/8/09 6:39 PM
 - Plan is to rerun it everytime a new package version is available. Publishing is on hold until detection system reports success for all packages -Harald Sitter 10/10/09 7:46 PM
- To help see if we are missing functionality in our KDE packages, check the CMake configuration outputs of all core KDE modules, and set up a wiki page listing the results. There they should all be sorted (e.g. stuff we don't want such as aspell or fam in kde4libs) and reviewed to see what software packages need MIRs.
 - I don't know how the archive reorg (if it ever happens) will affect the above in regards to MIRs. Build dependencies should still be reviewed otherwise - Jonathan Thomas 9/29/09 4:23 PM
- Make it a practice to always consult with upstream personally to determine whether or not a currently-pre-release piece of software will be ready for our release. To assume makes an **ass** out of **u** and **me**
- Until a proper bug triage team can be assembled, deflect all upstream tasks to upstream bugtrackers and close them in Launchpad. We usually never backport these and we will pick them up when KDE implements them, leaving no reason to track them in Launchpad. They just increase the amount of reports for the currently all-to-few triagers to go through. When a bug is shown to be severe enough to require an early fix, we can use Launchpad for that.
 - imho we should be much more strict with closing bugs (not only wishlist). there is imho really no point in keeping stuff open in launchpad if we are not going to write patches for it ourselves. close them and link to upstream bug with a very friendly note saying that we do not have the resources and need to rely on upstream to fix it in the next release. as stated earlier we don't have the manpower for backporting anyway atm -Lydia Pintscher 9/29/09 9:51 PM
 - There is some merit to tracking these bugs, since if a patch is committed upstream for a somewhat-serious bug we can track it and provide a fix before the next KDE point release. Though I must admit that most bugs aren't serious enough to warrant an early backport. Mostly I just link the bugs upstream and then close them when we release KDE 4.x.x that contains the fix. It's somewhat of a useless charade if you think about it. (Though maybe it offers some communication? I dunno) Maybe we should close them in Launchpad after we can with a great degree of certainty determine that it is an upstream bug? If the bug gets a lot of duplicates we can consider re-opening it and backporting fixes early. -Jonathan Thomas 9/29/09 3:55 PM
 - yes if a bug is severe and needs backporting sure keep them open - but honestly there is soooo much crap in launchpad that no-one of us can and wants to ever fix because it is either very hard, impossible, pointless, not worth it or just wrong. and we keep soooo many of those bugs open to not piss off a few random annoying people and thereby making everyone's life harder. -Lydia Pintscher 9/29/09 10:05 PM
 - Proposal updated by Jonathan Thomas at 9/29/09 4:13 PM
- Postpone apport crash handler use, relating to the above point.

- Can we handle all crashes from KDE being reported to LP in development releases? (Crashes are most always upstream bugs.) Can we handle all non-crash reports during stable releases? If not, should we postpone the use of apport until we have a proper triage team? -Jonathan Thomas 9/29/09 4:57 PM
- +1 on postpone, there is really no point in having everything piped through lp, considering it will not get through, but only and then rot for the next 3 years until someone happens to come around and triage some bugs -Harald Sitter 9/30/09 5:26 AM
- Compose a collection polite boilerplate responses for bug triaging that don't sound canned. Give them a bit of life and Genuine People Personality(tm), while at the same time being informative, without overly thanking them like the current Ubuntu ones do. (You can tell that those are canned responses from a mile away...) This will help improve niceness.
 - I was thinking about creating a websiteish thingy to spitout responsens with niceness attached to it in an alternating form so it gets an additional personal touch -Harald Sitter 10/1/09 9:04 PM
- Nuke install-package and gdebi-kde, replacing their functionality with KPackageKit
- Ubuntu will be replacing software-properties-gtk with functionality built in to the Ubuntu Software Center. This gives us an excellent opportunity to either write a custom KControl Module for our needs or hack missing functionality into KPackageKit. Continuing to use software-properties-* will lead to us maintaining the software-properties backend which we've never touched before, since it will be unmaintained by Ubuntu. This may or may not be urgent, depending on how soon software-properties-gtk is depreciated
- Ditch language-selector-qt as a standalone app. Either replace the KDE Locale & Language module with a custom module or patch language-selector's functionality into the existing one in a way that makes sense. If we go away from Rosetta, language-selector will be too tied to language-pack-* to suit our needs anyways.
- Consider rewriting printer-applet-kde in C++, if feasible. We should port it away from KSystemTrayIcon to KNotificationItem at any rate, even if a C++ port isn't feasible.
- Rewrite update-notifier-kde as a C++ kded module, which would use persistent KNotifications to notify of Reboots, upgrade hooks, etc. Maybe rename to kubuntu-notification-helper, since it's supposed to be a compilation of kubuntu-specific update tools and no longer notifies one of updates. Consider moving restricted extras notification elsewhere.
- Re-evaluate all default applications

User <=> Developer Interaction

- Push new bug report policy to Kubuntu user forums (kubuntu-de.org, kubuntuforums.net, etc) and inform the IRC support team.
- Consider composing a wishlist of Launchpad UI changes to make things easier for bug reporters.
- Perhaps consider a "newsletter" aggregating (more liek akregating, amirite?) user blogs of the week as they relate to Kubuntu.

Recruitment

- To help recruit bug triagers, regularly hold Kubuntu bug days. Publicize on blogs, etc. Improve current bug triage documentation. Hold tutorial/training sessions. Could be part of an general Kubuntu recruitment drive.
- Once Kubuntu regains some of it's image/street cred, ramp up recruitment drives for all "jobs" via blogs and other publications.

Call all hands

Set the record straight (Roadmap of final solutions)