

## Getting involved with Ubuntu development: reach and perception

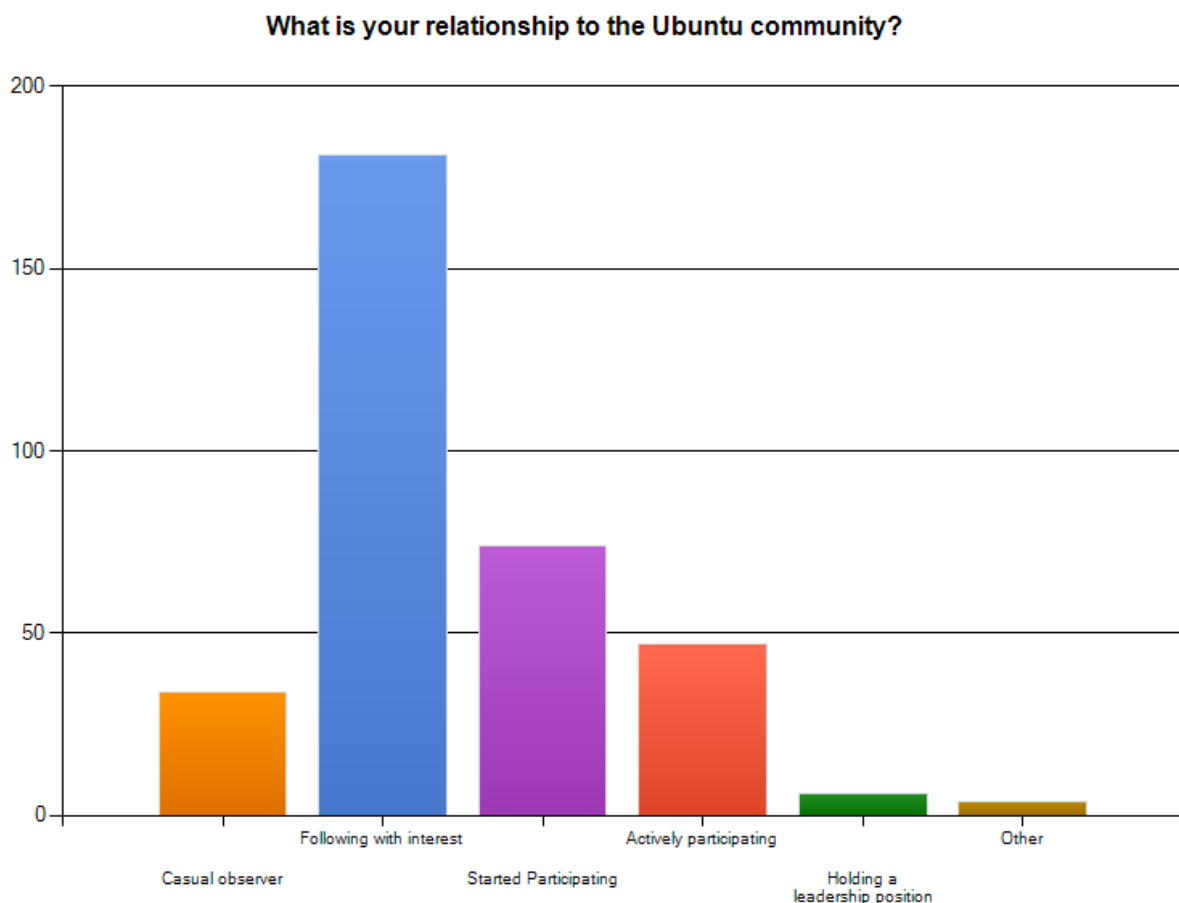
The goal of this survey was to find out more about how and where interested Ubuntu users find out more about Ubuntu development. Secondly the survey asked participants to rate the quality of documentation and suggestions for improvement.

The participation to this survey was open to everyone and it had 347 participants from 12th-24th October 2011. The survey was published and referenced to on Planet Ubuntu, the Ubuntu Fridge, OMG!Ubuntu! and all over microblogging world.

Thanks a lot everyone for your answers.

### *Interest and motivation*

#### What is your relationship to the Ubuntu community?

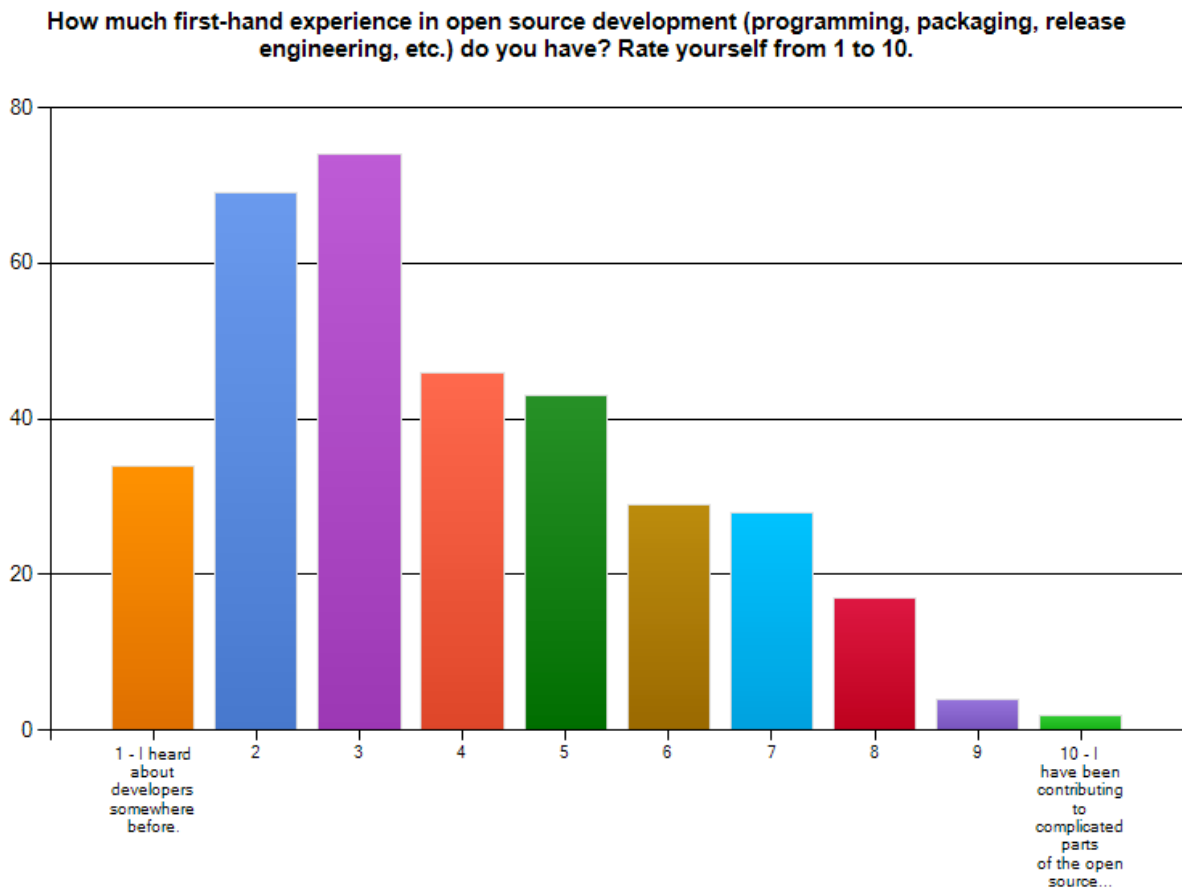


346 of 347 participants answered the question and the vast majority (87.3%) follow Ubuntu with interest (52.3%), started participating (21.4%) or are actively participating (13.6%) - together .

The people who answered „Other“ (2.02%) had this to say:

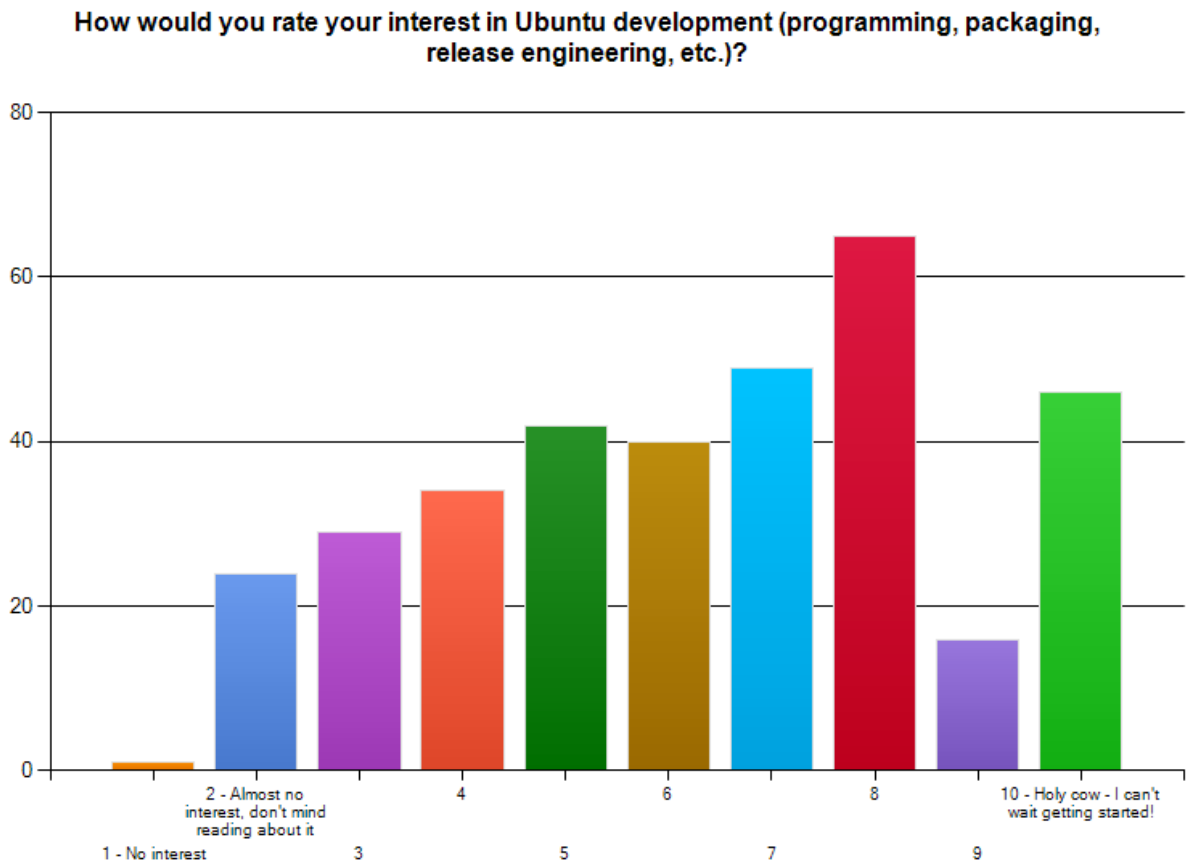
- bug reporting for oneiric beta.
- Former Ubuntu developer, now following with interest
- I am starting at Canonical in December, but I haven't been active in the community previously as I was using a rival distribution (I was very active there!)
- I work for Canonical, but would like to become an Ubuntu Developer.
- i use ubuntu
- i'm usually on the irc and i have some branches on launchpad from a couple of projects i'm participating in

## How much first-hand experience in open source development (programming, packaging, release engineering, etc.) do you have? Rate yourself from 1 to 10.



346 of 347 participants answered the question. Two thirds (67%) of the participants put themselves into the section of 2-5 (on a scale to 10). Three quarters (76.8%) put themselves in the lower half of the scale (1-5).

## How would you rate your interest in Ubuntu development (programming, packaging, release engineering, etc.)?



346 of 347 participants answered the question. 62.5% of the participants are in the upper half of the interest spectrum (6-10). More than a third (36.7%) are in section of 8-10.

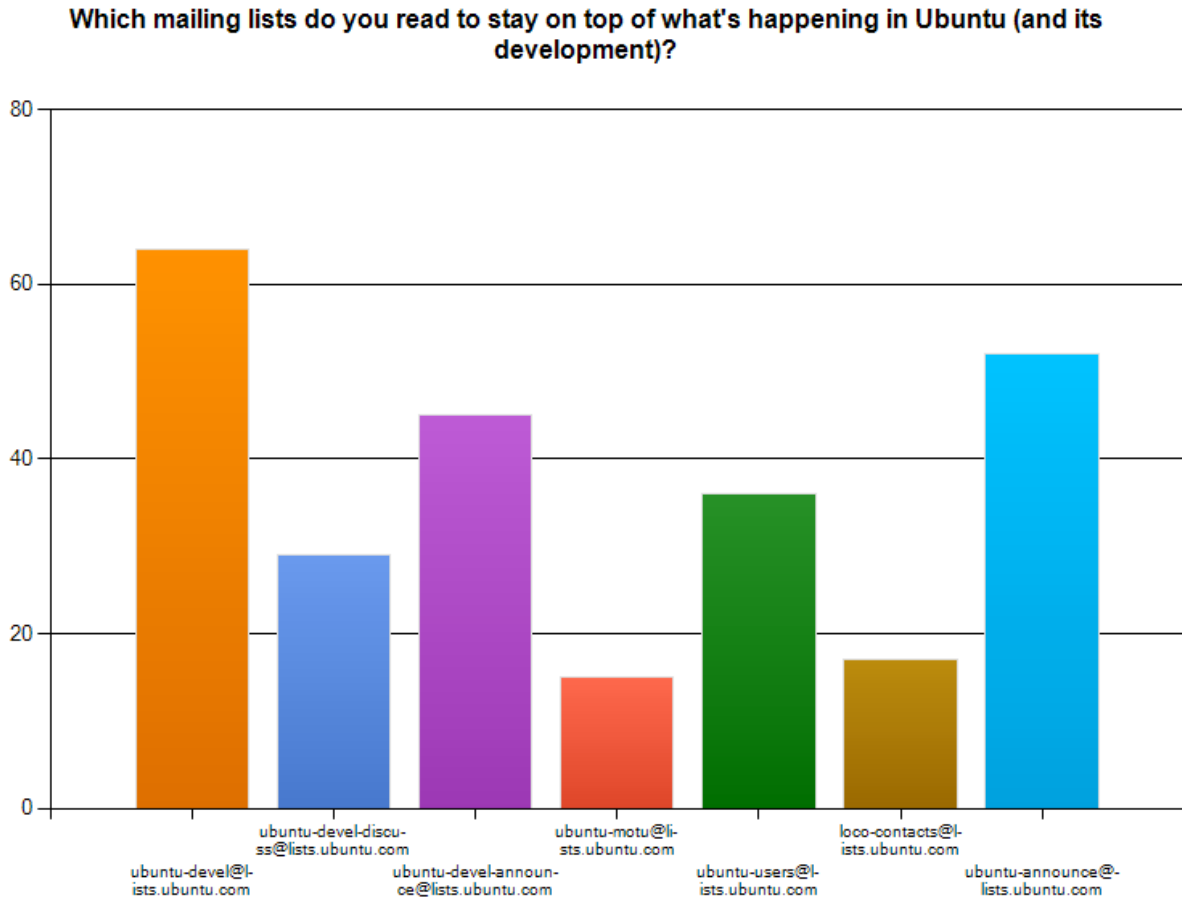
### Summary: Interest and motivation

This first part of the survey was intended to find out more about the participants of the survey. It is safe to say that the majority

- has a high interest in Ubuntu development,
- follow Ubuntu development with a high interest,
- despite a lack of first-hand experience are keen to find out more and get involved.

## Ubuntu News

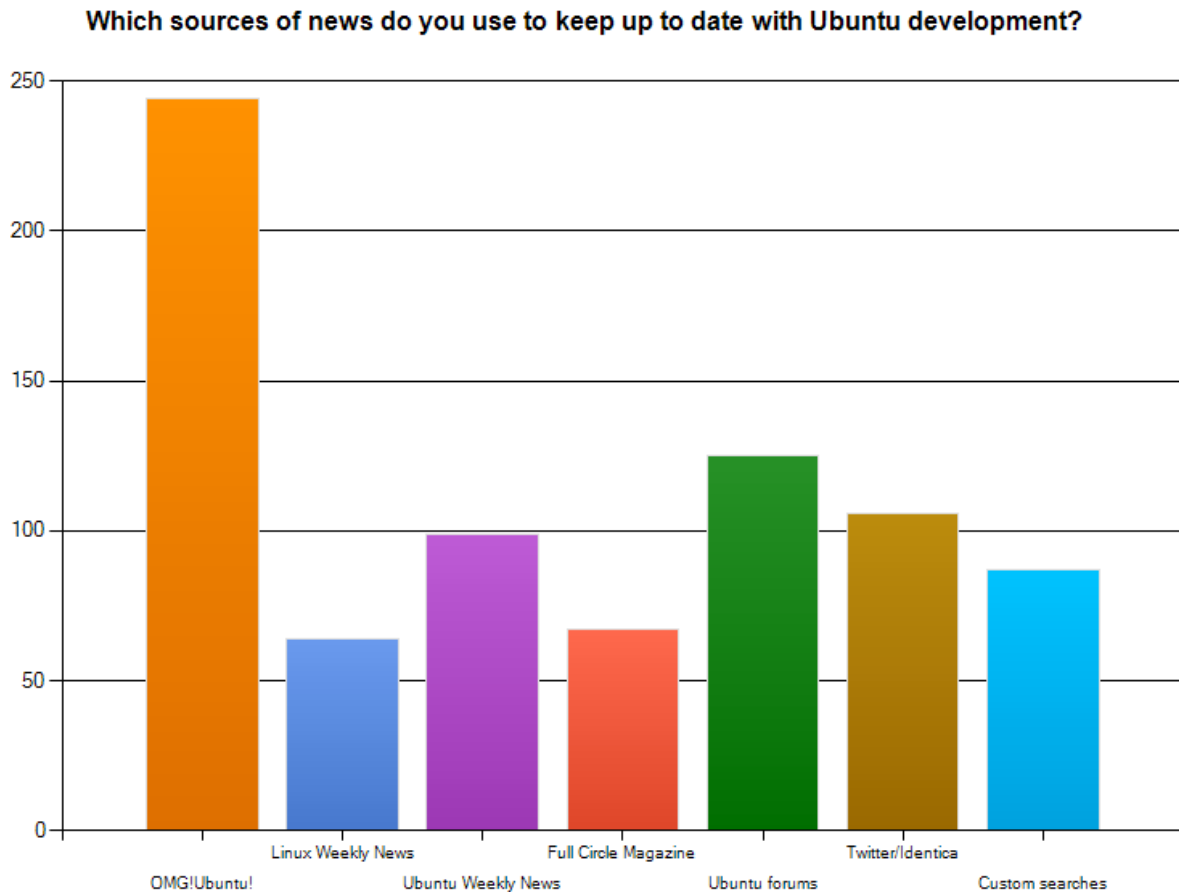
### Which mailing lists do you read to stay on top of what's happening in Ubuntu (and its development)?



Only 32.32% (133 out of 347) participants answered this question. The most-read mailing lists in this group are:

- [ubuntu-devel@lists.ubuntu.com](mailto:ubuntu-devel@lists.ubuntu.com) (48.1% - total 64)
- [ubuntu-announce@lists.ubuntu.com](mailto:ubuntu-announce@lists.ubuntu.com) (39.1% - total 52)
- [ubuntu-devel-announce@lists.ubuntu.com](mailto:ubuntu-devel-announce@lists.ubuntu.com) (33.8% - total 45)
- [ubuntu-users@lists.ubuntu.com](mailto:ubuntu-users@lists.ubuntu.com) (27.1% - total 36)
- [ubuntu-devel-discuss@lists.ubuntu.com](mailto:ubuntu-devel-discuss@lists.ubuntu.com) (21.8% - total 29)
- [loco-contacts@lists.ubuntu.com](mailto:loco-contacts@lists.ubuntu.com) (12.8% - total 17)
- [ubuntu-motu@lists.ubuntu.com](mailto:ubuntu-motu@lists.ubuntu.com) (11.3% - total 15)

## Which sources of news do you use to keep up to date with Ubuntu development?



89.04% (309 out of 347) participants answered this question and in this group, these news sources are read:

- OMG!Ubuntu! (79% - total 244)
- Ubuntu forums (40.5% - total 125)
- Twitter/Identicia ( 34.3% - total 106)
- Ubuntu Weekly News (32.0% - total 99)
- Custom searches (28.2% - total 87)
- Full Circle Magazine (21.7% - total 67)
- Linux Weekly News (20.7% - total 64)

## Please list additional news sites you read to keep up to date with what's happening in Ubuntu.

41.78% (145 out of 347) listed additional news sources. Here are the most listed news sources:

- Planet Ubuntu: 36 mentions
- webupd8.org: 25 mentions
- Other Ubuntu related pages: 16 mentions (fridge.ubuntu.com: 3, Ubuntu Vibes: 3, Ubuntu Geek: 2, ubuntu.ru, ubuntu-fr.org, iloveubuntu, ubuntu.life.net, forum.ubuntu.org.cn, fridge.ubuntu.cn, ubuntu.com, upubuntu)
- ubuntuusers.de: 7 mentions
- Other planets: 7 mentions (planet kde: 2, planet gnome: 2, planet ubuntu women, Planet Ubuntu ES, <http://ubuntuweblogs.org/>)
- Slashdot: 5 mentions
- Blogs (Daniel Holbach, Raphaël Hertzog, Mark Shuttleworth, „blogs“, Ubuntu One, ubuntu.life.wordpress.com): 6 mentions
- <http://www.h-online.com/open/>: 4 mentions
- phoronix: 4 mentions
- Canonical Voices: 3 mentions
- ars technica: 3 mentions
- KDE related pages: 3 mentions (Kubuntu News, KDE Dot news, KDE Apps)
- Mailing lists: 3 mentions (Ubuntu Server list, [software-center@lists.ubuntu.com](mailto:software-center@lists.ubuntu.com), Ubuntu UK list)
- Pro-Linux.de: 2 mentions
- Launchpad: 2 mentions
- softpedia.com: 2 mentions
- IRC: 2 mentions
- muylinux.com: 2 mentions
- gamingonlinux.info: 2 mentions
- reddit.com: 2 mentions
- derstandard.at: 2 mentions
- Tech Drive-in: 2 mentions
- askubuntu: 2 mentions
- Other: Google+, pollycoke.org, [www.fabio.com.ar](http://www.fabio.com.ar), [www.hexus.net](http://www.hexus.net), brlinux.com.br, novatillasku, Google Alerts on "Ubuntu Linux", Ubuntu weekly news on YouTube, linux today, linux gezegeni, dailynux.com, wikipedia, Guardian Tech News, unixmen.com, OSNews, news.softpedia.com, jupiterbroadcasting.com, This Week In Linux, lifehacker, PC World Linux Line)

## **Summary: Ubuntu News**

Only a third (32.3%) of the participants of the survey read an Ubuntu mailing list. For the most-read mailing list, [ubuntu-devel@lists.ubuntu.com](mailto:ubuntu-devel@lists.ubuntu.com), this means that only 15.5% of the participants of the survey follow this mailing list.

79% of the participants read OMG!Ubuntu! which very likely has to do with the fact that the information about the survey was posted on this page as well. Generally it seems to be regarded as a trust-worthy and interesting news site. Next on the list are the Ubuntu forums with 40.5% of the participants.

The list of additional news sources is very interesting and provides a great list of resources for information, with webupd8 and Planet Ubuntu being the top entries.

## ***Documentation and Outreach***

**Where did you learn about Ubuntu development (packaging, development tools and processes)? Please list all resources that come to your mind, no matter if they are documentation, support forums or TODO lists.**

45.82% (159 out of 347) answered this question. Here are the answers:

- Unspecified: 29 (Google searches: 7, blogs: 6, web: 6, some additional linux docs: 5, tutorials: 3, podcasts, manpages)
- Forums: 25
- Ubuntu Wiki: 24
- Other Ubuntu sites: 21 (askubuntu.com: 4, planet ubuntu: 4, ubuntuusers.de: 4, help.ubuntu.com: 3, ubuntu.com: 3, ubuntu-fr.org: 3, UWN, ubuntu-manual.org)
- OMG!Ubuntu!: 18
- Ubuntu developer/open week: 10
- developer.ubuntu.com: 9
- IRC (#ubuntu on Freenode: 1): 9
- Launchpad: 7
- Debian resources (Debian Policy and New Maintainers' Guide: 5, wiki.debian.org: 5, Wiki Minidebconf Panamá, debian.org): 11
- friends / individual people: 4
- quickly tutorial: 3
- books: 3 (Martin Krafft, The Debian System, Ubuntu Server, „books“)
- ubuntu mailing lists: 3 (ubuntu-users list, ubuntu-in mailing lists, „lists“)
- Other sites: 5 (youtube: 2, pollycoke, wikipedia, siteduzero.com)
- Other: (not started yet, development at work, have not started yet, N/A)
- Other communities: 5 (gnome docs: 4, linux community, qt development)

## What do you think should be improved in terms of documentation? If you can, list specific examples.

30.25% (105 out of 347) of the participants answered this question. Here are the answers:

- requests for new or updated/improved articles: 20 mentions specific examples of fixing a bug, process, more info about useful bugs, troubleshooting common problems: 2, compiling a kernel (redhat/fedora tools don't work), processes, python-gtk3 (pygobject) packaging a debian file, „library packaging“ needs improvement, merge/sync, differences from Debian, technical overview, java packaging, updating a package, integration with lp, general workflow of a motu, send patches to upstream, hardware enablement, integration docs (libunity, sound menu integration (no examples/sources in C language, no docs)
- add a beginner's guide: 12 mentions (add more examples, baby steps, visual tour)
- make it up-to-date: 10 mentions
- avoid wiki clicking madness: 7 mentions
- videos: 5 mentions (video/.gif, screencast, also for translating, reporting bugs)
- article: packaging from scratch: 5 mentions (formal and in-house, package python scripts)
- i18n for developer.ubuntu.com: 5 mentions
- development tools: 5 mentions: improve tools, so you need less docs: 4, setting up a dev environment takes time and hdd space, launchpad should link to „fix a bug“ docs
- general advice – 5 mentions: explain principles/conventions better: 2, more structure / organisation, provide more links for background info, avoid listing too many alternatives
- docs have improved a lot: 4 mentions
- we need searchable, central place for docs: 4 mentions
- practise / training material: 3 mentions
- make developer.ubuntu.com pages discussable: 2 mentions
- Other advice: adopt new look, make it MSDN like, packaging docs should be installed by default, add forums stickies

## Out of scope:

- app development: 16 mentions (GTK tutorials GObject tutorials GList tutorials: 3, api docs often not useful: 2, better and more 'real-life' tutorials/examples: 5, document new features (eg, gtk obj. Introspection): 2, standard libraries to solve common problems, more focus on Qt, make it easier for first time contributors, more in-depth articles, have a preferred rapid GUI development tool: 2, quickly docs non-existent, tour of all relevant bits and pieces)
- software documentation: 8 (get rid of / hide information for old releases: 6, more thorough, per-version, non-core but common software, more technical examples, Package descriptions should be clearer, PPA descriptions should be clearer, have big repository of docs, simpler apps → less docs, improve forums search function)
- N/A: 3

**If you have any additional suggestions to make Ubuntu development easier to understand or more exciting to learn about, please list them here. If they are unconventional, please let us know anyway.**

- First steps – 16 mentions: help people with no experience help out: 4, videos: 3, explaining a how to fix a bug in a session step-by-step and maybe screencast it: 2, tutorials: 2, have more hand-holding/mentoring: 2, try to explain isolated tasks first (no need to understand everything first), clear link from ubuntu.com, add a packaging guide overview, check lists, give contributors a sense of progress, explain common pitfalls, encourage to use mailing lists, provide guides that go from easy to advanced
- Better docs – 7 mentions: more tutorials: 2, make them installable in software-center, howtos for everything, address right audience, explain how to set up ppa and build something with patches, point out how to help upstream.
- Tools – 7 mentions: make Launchpad easier navigatable, bug duplicates hard to find, use git, tools should do all the checking work, UDD has problems with quilt and uptake, make Red-Hat makefiles work on Ubuntu, Provide development live cd, improve tools, just recommend one, improve PPA story (different version numbers, etc.), fixing stuff upstream should be easier.
- List of easy bugs (the harvest bugs seem still too difficult, also the bitesize ones) – 4 mentions
- LoCo – 4 mentions: work with local teams/schools to spread the word: 2, worldwide distributions with another run of shipit, make LoCo teams more inviting
- Easier insight into what's currently happening in the release - 2 mentions
- QA – 2 mentions: Quicker replies to bugs, to avoid demotivating / less useful experience. Filing bugs should be easier.
- Make developer.ubuntu.com central place to go.

Out of scope:

- app development – 18 mentions: turn quickly into GUI IDE: 4, add platform overview: 2, focus on Qt, hide useless API docs from Google, GNOME docs better than Ubuntu Desktop library docs, pointers to learning preferred language, Canonical should hire a team to write proper docs, too many alternatives, tools are too hard, more examples, more interesting financial eco-system, better docs, more android support, explain steps graphically, automatically generated docs don't cut it, define stable API
- Ubuntu general – 4 mentions: like Ubuntu Classic better, more/easier configuration options, follow GNOME 3 way, rolling release

## Summary: Documentation and Outreach

The sheer mass of suggestions is fantastic. It shows that the participants of the survey deeply care about understandability of documentation and are deeply interested in finding a way to get involved.

While some straying away from the topic in survey answers is unavoidable, the survey could have made it clearer that the focus was Ubuntu development (as in Ubuntu, the platform) as opposed to Ubuntu software in general or Ubuntu App development. Still the results are very interesting, so this list will provide a great grounds for discussion at UDS.

The most important requests were:

- additional and updated articles
- make the guide more beginner friendly, provide sufficient examples
- make and keep it up to date
- avoid having to read heaps of wiki pages, provide a fluent reading experience
- provide videos or screencasts
- an article that was requested very often was one about packaging from scratch
- make the guide translatable
- a number of concerns about the tools were raised as well, most fairly generic

Interestingly enough a few remarked that the quality of the documentation has improved recently.

In the section about general improvements to the experience of getting involved in Ubuntu development, these high-profile requests were made:

- Improve the experience for the very first steps.
- Provide a list of easy tasks to get started with.
- Generally improve the docs.
- Make [developer.ubuntu.com](https://developer.ubuntu.com) the central point to go to.
- Improve the tools experience.

Thanks a lot for all the answers. For everybody interested in tools or documentation reading the entire answers is heartily recommended.

## Appending: entire answers

***Please list additional news sites you read to keep up to date with what's happening in Ubuntu.***

- <http://daniel.holba.ch/blog/> [ubuntu-server@lists.ubuntu.com](mailto:ubuntu-server@lists.ubuntu.com)  
<http://raphaelhertzog.com/>
- <https://plus.google.com/>
- Facebook
- [launchpad.net](http://launchpad.net)
- Slashdot in english There's an argentinian site ([www.fabio.com.ar](http://www.fabio.com.ar)), who's an OpenSource enthusiast and he always mentions Ubuntu releases.
- /. <http://www.h-online.com/open/> <http://voices.canonical.com/>  
<http://www.markshuttleworth.com>
- Ubuntu Weekly Newsletter Planet Canonical ("Canonical Voices")
- [ubuntuusers.de](http://ubuntuusers.de)
- [ubuntuusers.de](http://ubuntuusers.de)
- Ubuntu planet, Ubuntu UK mailing list
- [ubuntugeek](http://ubuntugeek.com), [planet ubuntu](http://planet.ubuntu.com), [linux gezegeni](http://linux.gezegeni.com), the H,
- [www.dailynux.com](http://www.dailynux.com) [ubuntu-fr.org](http://ubuntu-fr.org)
- wikipedia
- [Ubuntugeek](http://Ubuntugeek.com), [Gamingonlinux.info](http://Gamingonlinux.info)
- Planet Ubuntu
- [slashdot.org](http://slashdot.org)
- [webupd8.org](http://webupd8.org) [unixmen.com](http://unixmen.com) [softpedia.com](http://softpedia.com)
- irc
- Web Upd8
- Phoronix; Gaming on Linux
- [forum.ubuntu.org.cn](http://forum.ubuntu.org.cn) [fridge.ubuntu.cn](http://fridge.ubuntu.cn)
- [webupd8.org](http://webupd8.org) [news.softpedia.com](http://news.softpedia.com)
- [ubuntulife.net](http://ubuntulife.net)
- [ubuntuusers.de](http://ubuntuusers.de)
- [webupd8.org](http://webupd8.org)
- [phoronix.com](http://phoronix.com)
- <http://www.webupd8.org/>
- Planet Ubuntu

- webupd8
- webupd8.org
- Tech Drive-in, This Week In Linux
- www.ubuntuusers.de derstandard.at/web
- AskUbuntu, Ubuntu's home page, Lifehacker, Ubuntu-News.net, PC World Linux Line, Slashdot, Ubuntu One Blog
- all sorts of blogs
- planet.ubuntu.com ubuntuusers.de
- phoronix
- <http://www.webupd8.org/>
- Webupd8
- ArsTechnica
- Phoronix
- jupiterbroadcasting.com
- brlinux.com.br h-online.com
- www.phoronix.com www.hexus.net
- webup8
- WebUpd8
- Linux Today, Web Upd8
- [reddit.com/r/ubuntu](https://reddit.com/r/ubuntu)
- Web UPD8 Ubuntu Vibes
- webupd8.org
- Exploring Launchpad.
- Tech Drive-in (<http://www.techdrivein.com/>) Web Upd8 (<http://www.webupd8.org/>) Ubuntu Vibes (<http://www.ubuntuvoices.com/>)
- <http://www.webupd8.org>
- planet.ubuntu.com
- planet
- planet.ubuntu.com
- Ubuntu weekly news on YouTube
- Phoronix Ars Technica Slashdot
- pollycoke.org
- WebUpd8, Canonical Voices
- gnome and ubuntu planets, [software-center@lists.ubuntu.com](mailto:software-center@lists.ubuntu.com)
- planet.ubuntu.com
- planet.ubuntu.com freenode

- webupd8, planet.ubuntu.com, novatillasku, ubuntu.life.wordpress.com, muylinux.com, askubuntu
- ubuntu.vibes.com webupd8.org
- Ubuntu Planet
- planet.ubuntu.com
- PlanetUbuntu.es
- planet.ubuntu.com
- Planet Ubuntu Pro-Linux.de derStandard.at
- planet.ubuntu.com
- planet.ubuntu.com
- Planet Ubuntu
- planet.ubuntu.com
- Planet Ubuntu
- planet.ubuntu.com
- Web UPD8
- planet.ubuntu planet.gnome ubuntuusers.de webupd8
- Planet Ubuntu, Planet Ubuntu Women
- Planet Ubuntu
- Guardian Tech news
- Ubuntu Planet
- planet.ubuntu.com
- ubuntu.ru
- planet.ubuntuusers.de, planet.ubuntu.com, pro-linux.de
- <http://planet.ubuntu.com/> <http://ubuntuweblogs.org/>
- Google Alerts on "Ubuntu Linux"
- planet ubuntu
- planet.ubuntu.com
- planet.ubuntulinux.org
- arstechnica
- webupd8.org
- Planet Ubuntu Planet KDE Ubuntu Fridge
- Planet Ubuntu
- Following feeds: Planet KDE Planet Ubuntu Ubuntu Fridge Kubuntu News KDE Dot news KDE Apps
- webupd8 upubuntu iloveubuntu
- planet

- Planet.Ubuntu.com
- PlanetUbuntu Webup8
- Reddit, OSNews
- www.webupd8.org and www.muylinux.com
- planet.ubuntu.com planet.gnome.org
- <http://ubuntu-ast.org/>, <http://blogs.altuxa.com/tapaponga/>
- webupd8.org, planet.ubuntu.org
- <http://planet.ubuntu-it.org> <http://planet.ubuntu.com> <http://www.webupd8.org/>  
<http://www.ossblog.it/> <http://www.lffl.org/> <http://www.chimerarevo.com/>
- Planet Ubuntu Slashdot NYTimes
- google laert
- Planet Ubuntu Ubuntu Power Users Ubuntu-UK
- Planet Ubuntu
- ubuntuusers.de (huge german community site)
- planet.ubuntu.com Daniel's Blog, Ubuntu Development Blog
- WebUp8
- webupd8 is great too!
- webupd8.org
- Planet Ubuntu
- Webupd8.org
- - Planet Ubuntu - Ubuntu Root - On Ubuntu - This Week in Linux - Linoob - Open Ended - The Linux Action Show
- slashdot, distrowatch, ubuntu.com
- lxr.com distrowatch.com planet.ubuntu.com
- Distrowatch, Linux power
- planet.ubuntu.com
- Webupd8.org, also Libreoffice.org
- Distrowatch
- planet.ubuntu.com
- planet ubuntu, planet gnome, bazaar developers blog, planet launchpad, ubuntu one blog, canonical blog, planet ubuntu users, ubuntu app developer,
- WebUpd8
- web upd8
- planet.ubuntu.com planet.ubuntu-fr.org www.phoronix.com
- www.webupd8.org
- Ubuntu UK Podcast

- WebUpd8 UbuntuGeek Tech Drive-In
- OSNews and FSDaily occasionally, also DistroWatch and various Linux related planets.
- canonical voices
- Webupd8.org
- planet.ubuntu.com
- WebUp8 Ubuntu Planet Ubuntu UK Planet
- <http://jupiterbroadcasting.com> <http://webupd8.org>
- linuxmagazin, prolinux.de ubuntuusers.de
- ayatana mailling list
- RSS Feeds from blogs of recently known ubuntu people
- planet.ubuntu.com webupd8.org ubuntuusers.de
- ubuntu.se
- ubuntu-uk mailing list
- lxnews.org
- ubuntu.com
- <http://techlaze.com>, <http://www.phoronix.com>, <http://www.webupd8.org>, <http://iloveubuntu.net> Ubuntu, Gnome & ubuntu-fr planets & lot of dev blogs.

***Where did you learn about Ubuntu development (packaging, development tools and processes)? Please list all resources that come to your mind, no matter if they are documentation, support forums or TODO lists.***

- help.ubuntu.com ubuntuforums.org #ubuntu on Freenode askubuntu.com ubuntu-users and ubuntu-in mailing lists developer.ubuntu.com
- <http://omgubuntu.co.uk>
- [wiki.ubuntu.com](http://wiki.ubuntu.com) [ubuntu developer](http://ubuntu.developer.ubuntu.com) [wiki.debian.org](http://wiki.debian.org)
- Ubuntu forums
- Through the aforementioned sources.
- ubuntu forums, launchpad
- <http://developer.ubuntu.com/>
- not started yet, btw I've read something doc/web etc...
- [ubuntuforums.org](http://ubuntuforums.org) [launchpad.net](http://launchpad.net) [wiki.ubuntu.com](http://wiki.ubuntu.com)
- [wiki.\[debian.ubuntu\].org](http://wiki.debian.ubuntu.org)
- OMG Ubuntu
- did a "quickly" tutorial - but not much more.
- work (we dev on Ubuntu, deploy on Debian; ie OpenStack, Django), planet Ubuntu.
- blogs
- [doc.ubuntu-fr.org](http://doc.ubuntu-fr.org) [developer.ubuntu.com](http://developer.ubuntu.com)
- [www.ubuntu.com](http://www.ubuntu.com)
- Launchpad, local forums about Ubuntu, OMG! Ubuntu, IRC channels.
- Debian Policy and New Maintainers' Guide
- I have read a some pages about the global process of development, from bugs to getting your stuff spread.
- From the Linux community at large
- [OMGubuntu](http://OMGubuntu.com) [ubuntuforums.org](http://ubuntuforums.org) [ubuntuusers.de](http://ubuntuusers.de) [askubuntu.com](http://askubuntu.com)
- irc, forums, [debian.org](http://debian.org)
- Ubuntu wiki, google searches
- I don't know how to answer this , but I love QT, and I want to use it if I would had a chance to develop something based on ubuntu
- N/A
- Wikipedia, [OMGUbuntu!](http://OMGUbuntu.com), and some additional linux docs and stuff
- [omg ubuntu](http://omgubuntu.com), the ubuntu forums

- Ubuntu com
- <http://ubuntuusers.de/>
- Introductions to Ubuntu development on various blogs and websites and the Ubuntu Weeks
- Canonical run IRC lessons. Google.
- friend of mine, [ubuntu-manual.org](http://ubuntu-manual.org), [www.ubuntuusers.de](http://www.ubuntuusers.de), [www.omgubuntu.co.uk](http://www.omgubuntu.co.uk)
- OmgUbuntu
- google searches; essentially everything (docu, tutorials, ...) that is now linked in [developer.ubuntu.com](http://developer.ubuntu.com) (I only do Python and some C)
- support forums, ubuntu planet
- Ubuntu forums, few blogs and [help.ubuntu.com](http://help.ubuntu.com)
- Google searches.
- OMGUbuntu
- friends
- forums
- ubuntu-fr
- I dont know about packaging, or development tools. I am a doctor by profession
- Ubuntu Forums
- Launchpad, man pages, sometimes an Ubuntu forum or wiki if a particular search leads me there. (I'm not a developer though, just a user who sometimes strays from the beaten path)
- OMG Ubuntu
- it goes all the way back to free Ubuntu CD shipping times.
- <https://wiki.ubuntu.com/UbuntuDevelopment>
- Your documentation/wiki. OMG! Ubuntu's introduction to programming things.
- Mostly Ubuntu forum, other parts from various sites found with Google.
- packaging forums Ubuntu Development Update
- omgubuntu
- ubuntu wiki irc mailing lists
- OMG ubuntu ubuntu forums ubuntu development tutorials Ubuntu Developer Portal <http://developer.ubuntu.com/>
- Ubuntu Developer Week, Google
- -develop Homepage -Launchpad
- ubuntu forums
- Ubuntu Weekly Newsletter, Planet Ubuntu, Launchpad
- Debian maintainer's guide, some Ubuntu wiki pages that pointed to each other
- Support forums wiki

- Internet books
- YouTube video
- official forums blogs: pollycoke
- Documentation Forums Q&A sites
- Podcasts
- i read the software-center wiki
- IRC online docs
- Ubuntu Wiki, Devhelp, GNOME Library, PyGTK documentation, Quickly
- Documentation.
- <https://wiki.ubuntu.com/UbuntuDevelopment>  
<http://www.ubuntu.com/community/get-involved>
- IRC development days Online docs [ubuntuforums.org](http://ubuntuforums.org) programming forum That 'quickly' thing
- Fortuitously on the net
- Javascript, Clutter, Gdk in JS (getting pixel colors)
- Mostly from the classroom sessions.
- tutoriales, documentacion
- [siteduzero.com](http://siteduzero.com) [omgubuntu.co.uk](http://omgubuntu.co.uk)
- #ubuntu-classroom and the logs <https://wiki.ubuntu.com> <http://wiki.debian.org>  
<https://blueprints.launchpad.net/> Upstream documentation e.g. Gtk, Gnome-Live others ;-)
- [Ubuntu-fr.org](http://Ubuntu-fr.org)
- [wiki.ubuntu.com](http://wiki.ubuntu.com) IRC Open Week sessions on IRC
- [wiki.ubuntu.com](http://wiki.ubuntu.com) developers guide deb package
- Open week Python packaging Packaging guides on Ubuntu and Debian wikis How to fix a bug guide on the wiki Launchpad help The bug triaging pages on the wiki Reading source code PyGTK online documentation
- IRC and documentation, mostly from individual people
- debian policy manual, debian wiki etc
- Reading about it via OMG! but have not started looking ot learn yet due to study commitments
- wiki, dholbach's IRC tutorials & videos, ScottK
- <http://help.ubuntu.com> <http://ubuntuforums.org/> <http://askubuntu.com/>
- While trying tests of Ubuntu 11.10Beta
- The official Ubuntu server book. I did unsuccessfully attempt to read the development forums. Ubuntu testing,,,,
- internet
- Book: Martin Krafft, The Debian System Ubuntu Wiki [Ubuntuusers.de](http://Ubuntuusers.de) Wiki (i am

contributor to) The Internet ;)

- Mostly through google searches and the Ubuntu Forums
- packaging guide, dholbach intro to ubuntu development, debian new maintainers guide
- forums
- Wiki Minidebconf Panamá
- debian packaging guide, blogs
- wiki.ubuntu.com
- Documentation, ubuntu wiki.
- I read the MOTU packing guide.
- ubuntu wiki, official documentation, chat logs of ubuntu training classes, blog entries
- alpha and beta versions
- planet.ubuntulinux.org omgubuntu.co.uk wiki.ubuntu.com
- Documentation, Forums, Blogs, Ubuntu User Mag.
- Unfortunately there is no direct source I remember, but most of the time it was a certain wiki (ubuntu & debian)
- ubuntuusers.de
- Ubuntu documentation, some videos i've came across on various sites, irc, debian documentation
- Debian
- see above plus help.ubuntu.com
- I've been a Linux developer since the mid-1990s. Packaing I learned on Debian and then transferred to Ubuntu when it launched. I've recently been reading the Ubuntu packaging guide on the Ubuntu Wiki to refresh those skills.
- Ubuntu wiki and forums mainly
- wiki
- ubuntu wiki
- IRC, documentation
- I get it rom friends.
- saw a article on LW (linux world ? ) can't remember name, linked to on slashdot
- on IRC and mailing lists
- Ubuntu forums, Ubuntu wiki, LoCo team
- <https://help.launchpad.net/Packaging/PPA>
- Planet feeds, both debian and ubuntu. Someone (Beila ? that's probably not the correct spelling) did a fantastic diagram and discussion about the various dpkg states and scripts invoked at each stage.
- google reader, facebook, blogs, full circle, omg!ubuntu, etc.

- Planet Ubuntu Ubuntu Forums (Fora?) :-)
- <http://forum.ubuntu-nl.org/> <http://wiki.ubuntu-nl.org/>  
<http://www.ubuntu.com/support/community> <http://askubuntu.com/>
- Documentation, ubuntu planet, forums, aforementioned news outlets
- [developer.ubuntu.com](http://developer.ubuntu.com) [ubuntuusers.de](http://ubuntuusers.de)
- I learned from a Debian Developer about packaging, everything else was searched for via google. In the future I hope to find everything on [developer.ubuntu.com](http://developer.ubuntu.com), but currently, the documentation there sucks (It's more an advertisement than a source of information.)
- Documentation, looking at source code, Python help(), IRC, blog posts
- IRC Talk on Getting started Redad the documentation available
- Google
- From Debian's packaging documentation and from the packaging guide on the Ubuntu wiki.
- News sites such as OMGU, official Ubuntu forums, AskUbuntu and the ubuntu-power-users mailing list.
- I started to observe how quickly did things, ubuntu-application template and help from IRC. Docs have started to get better but need a lot of work.
- Ubuntu Wiki, blogs talking about Ubuntu ecc...
- [irc.wiki.ubuntu.com](http://irc.wiki.ubuntu.com) (and external links in it)
- Python documentation and tutorials, quickly examples and documentation and <http://vimeo.com/dgsiegel/videos/sort:date>
- [wiki.ubuntu.com](http://wiki.ubuntu.com) [developer.ubuntu.com](http://developer.ubuntu.com) [ubuntuforums.org](http://ubuntuforums.org) [google.co.uk](http://google.co.uk)
- Ubuntu wiki
- [ubuntuforums.org](http://ubuntuforums.org), wiki,
- Official wiki, other people, IRC, identica, various blogs around the net.
- [Ubuntu.com](http://Ubuntu.com)
- [Habrahabr.ru](http://Habrahabr.ru)
- [omg.ubuntu](http://omg.ubuntu.com)
- dholbach's series on [omgubuntu](http://omgubuntu.com) ubuntu app developer weeks ubuntu forums
- [status.ubuntu.com](http://status.ubuntu.com) [wiki.ubuntu.com](http://wiki.ubuntu.com) [www.launchpad.net](http://www.launchpad.net)
- Searching in google
- Tutorial on blogs, reading source code of other projects/sample projects on Launchpad, Ubuntu wiki documentation.
- [ubuntuusers.de](http://ubuntuusers.de), [debian.org](http://debian.org), [gnome.org](http://gnome.org), [apache.org](http://apache.org), [paperc.de](http://paperc.de), books (e.g. linux kernel).
- mainly launchpad
- Canonical Developer week (introduction)

- Ubuntu Forums PyGTK Documentation Stack Overflow Rick Spencer
- askubuntu ubuntuforums ubuntu wiki
- <http://www.youtube.com/watch?v=zKLabXTqMc>
- The Ubuntu wiki
- ubuntu forums ubuntu development weeks (followed chat from ubuntu devs)
- Ubuntu wiki
- [wiki.ubuntu.com](http://wiki.ubuntu.com)
- ubuntu forum online lessons
- lernid, wiki
- ubuntu-uk mailing list and planet ubuntu
- Ubuntu wiki, askubuntu, debian docomrntatin, man pages
- some site or wiki, don't remind exactly
- Ask Ubuntu

***What do you think should be improved in terms of documentation? If you can, list specific examples.***

- More specific examples of actual bugs being solved to see how others start fixing a bug. 'fix the bug' isn't really specific.
- I had a hard time to compile the Kernel, "the Ubuntu way", this is very poorly documented, furthermore the standard tools (On Red Hat/Fedora) usually just do not work on Ubuntu.
- document the current processes, like the distributive way to fix bugs (bzt), the tools that developers are using and how, maybe some videos showing how the merge, sync, package, send patches to upstream, etc.
- I think there needs to be a "beginner's guide" that lays out the packaging process (using an example) in clear, baby steps. A "sticky" in Ubuntu Forums could reference this guide. There will be some programmers who want to create .deb files without formally adding to the Ubuntu repositories (e.g. for in-house computers), and other programmers who would like to add to the repositories. Both types need addressing.
- how about a more in-depth tutorial than the small tutorial 'hello world' video for Quicken, which is 'the' development method for Ubuntu?
- -
- developer.ubuntu.com in more languages
- First I need to start after I'll say something :)
- More about relationship with specific hardware
- It's difficult to find a starting point. (except for "quickly" - but then you need to program in Python)
- My experience was with Debian docs, and now I'm packaging only Python projects (with setuptools and stdeb is very easy!).
- Easier explanations and a no-much-click documentation to navigate between different "linked" pages.
- Localization of documentation should be improved also, one big easily searchable place for everything should exist. Documentation should be available directly from user desktop post-installation.
- More take-me-by-the-hand tutorials for people interested in getting started. Asking experienced contributors all the time feels like you're wasting their time, and banging your head against dead-ends on your own doesn't help anybody
- most of I think you need to learn as you go. The docs are good but once you get stuck it is hard to find your way. The folks on IRC are wonderful but a little more docs on what to do when you hit some of the usual snags would be great.
- I can't find useful information about libunity, sound menu integration (no examples/sources in C language, no docs)
- I think we should gather all the documents of all the parts that contained in

ubuntu. Ubuntu do a lot of work in standardising linux , but we often can't find the detailed information about one part. For example, I was bothered in how to make a sound in ubuntu, but there are too many docs around me, also, jack, openal, pulseaudio.... I found myself lost in them, I didn't know which one should I read to solve my problem.

- N/A
- Delete many docs for unsupported versions, by that I mean things like fixing GRUB and stuff, NOT upgrades, release notes, etc.
- Allow developers to contribute using Qt libraries where possible. I think it is about time Qt is incorporated into the development tools for Ubuntu modules... More fun to write, more developers proficient with it, exceptional documentation. Sticking with Gnome exclusively for Ubuntu development limits contributors...
- Visible tutorials and guides not only for people that are already experienced programmers, but also for people like me with minimal programming experience and no experience with making a fully fledged program. Something like a mixture of "Your first Ubuntu app(s) in python" and "Distributing your app to the community".
- I find it still hard to contribute a bug to Launchpad. I once did and couldn't answer the technical questions of the other guys, since it was too specific!
- I think that testing, packaging and uploading should be done automatic by a software (roughly as quickly already does). But the typical tools to build software, the libraries, ... they typically are extremely bad documented. Since I only use Python, I call tell: - Python is extremely well documented - glade, gtk, gstreamer, the special ubuntu notification system, the unity components, quickly: FROM A PYTHON PERSPECTIVE they are extremely bad documented, if at all (to be specific: \* go through <https://wiki.ubuntu.com/NotificationDevelopmentGuidelines> and look for the python examples \* a quote from here: <http://theravingrick.blogspot.com/2011/02/python-sound-menu-integration-or-gdbus.html> : "It seems simple enough, if you implement the Impres2 interface, you get Sound Menu integration, along with other benefits. Well, it turns out that implementing such an interface in Python is an exercise in frustration." (I tried to implement the same last year and ended up scrapping code from different projects until it worked for me. If ubuntu would be serious about the sound menu, then there should be a well documented python lib that does this integration. I now have it working for me, with days in frustration, but for the architects of the sound menu, if they knew python, it would have been 3 hours to build the library and 3 hours to write a perfect documentation. I guess I could go on with lots of other parts. Glade for example needs lots of documentation, especially for the python support part. ) \* THIS: <http://developer.ubuntu.com/api/ubuntu-11.04/GIR/python/Unity-3.0.html> is NOT documentation. This is just part a of the code strangely formatted.
- Perhaps more thorough per-version software documentation for non-core but common software.
- I would think about adding some videos or gif animations.
- When new features are released (eg, gtk obj. introspection), it'd be nice if there was a half decent tutorial in the Ubuntu target languages (C, Python, Vala, C#)

- I'd like to see old forum posts archived or tucked out of the way so they are not prominent in searches. With each new Ubuntu release, it becomes harder to find relevant information.
- Have documentation updated and precise. Not 3 releases behind.
- video how to to report bugs, translate, triage...
- It should be like Arch Linux Wiki, where every possible problem is explained with solution.
- Sorry, like I said, I'm not a developer, so I can't really say what's good or not so good about documentation about development. For a user; Manpages could do with more technical examples sometimes so it will be easier to compose the command line I want. Package descriptions should be clearer/more extensive so I can tell which of two or more similar packages will be more useful for what I want to accomplish. PPA descriptions should be clearer/more extensive so I can tell which PPAs to add to keep my favorite applications up to date without needing to keep up with all of the development release (which very often breaks stuff).
- It seems to be improved quite a lot since I read it a few years ago. So sorry, nothing constructive to note as of now. Will certainly check it out.
- Your pages seem "haphazardly strewn" everywhere, or something. The information is there, but there is no clear path to get to it. Is wiki.ubuntu.com the same as help.ubuntu.com? Who knows!?
- I would love to see a small tutorial series for developing Ubuntu apps. I can pretty easily find lots of tutorials to show me how to develop an up to date rails app, but it can be difficult to find almost any idea of where to even begin to develop almost anything on the Ubuntu desktop. I'd love to write, for example, a rhythmbox plugin, but it seems fairly difficult to figure out how to even begin. Especially compared with other platforms like rails, php, or even cocoa.
- - a lot of tutorials like how to's, introductory stuff. I think more experienced contributors don't need this, but for a beginner in Ubuntu development, this is crucial.
- Documentation should read like a book. An example is Twitter API: <https://dev.twitter.com/docs/api/1/post/statuses/update> (picked one feature). It starts with what is the use of this function, what are the parameters, with examples. Next is a full example of how to call the function and the output. Followed by some oddities or other exceptional cases which should be noted. Now the Ubuntu doc for Unity: <http://developer.ubuntu.com/api/ubuntu-11.04/GIR/python/Unity-3.0.html> A generated list without any human readable text of what it does. I'm a novice Python programmer and with 8 years of experience in other languages yet this documentation scares me.
- It's hard to say. Packaging is pretty complicated. Even the Debian docs are hard to follow. But that's partly because there is a lot to explain. I checked out and built Unity from source early in the Oneiric cycle, and that was fairly difficult since it was a young, sparsely documented project split into several pieces (up-to-date Compiz source required etc). Accurate step-by-step instructions all the way from obtaining project source and dependencies, building, and proposing changes to be merged would be a magical and helpful combination.
- GTK tutorials GObject tutorials GList tutorials

- I'll like a reputation system for the members. Each article (man pages) under a wiki. I'll like a biiig repository where all sources make The Ubuntu Manual.
- for me it was more the tools, gtk+ that often lack decent documentation, especially for the some of the language bindings. people were helpful and showed me the way.
- Have less things to document, so documentation can be smaller.
- Structure and organisation. There is not a single, coherent platform. Not one place to go to.
- It would help to have a preferred rapid GUI app development tool - and then lots of good examples how to use it. I find python a bit confusing and the quickly thing (is it even recommended?) to be not enough in terms of more advanced development. For me, minor bug fixes isn't as interesting as building new user friendly applications.
- Too many 'alternatives' pop up - and they are too old.
- Need simplified packaging guide.
- tutoriales mas completos, por ejemplo para la creacion de lens o de app-indicators. No se encierren en un lenguaje, me gusta programar en python, pero prefiero c, c++ o vala y generalmente no hay buenos tutoriales para estos lenguajes
- \* <http://developer.ubuntu.com/> is a good start, but on some pages there is just dead end. More links, examples, workflows, ... \* The community should be better integrated to write documentation (both user documentation and developer documentation), but as long as the infrastructure will be abysmal, many do not participate. \* Connect forum threads with each page to be able to dicuss the content \* Guide for writing good Dokumentation \* QA: "Testet for Natty, ..." statements on every page \* Formatting guidelines so that the pages look similar \* Better wiki syntax, which is adapted to the requirements of such documentation. (as Inyoka at ubuntuusers.de) \* Compare to the German wiki "[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de)" how to do something like this right
- Documentation on the wiki needs to be update. We should think about a visual tour about main topics which serves as index of the learning process
- More PyGI documentation.
- I think more on the principles and reasons why we do things the way we do them. Practical examples are nice but there are so many corner-cases in development that it would be good to know the principle behind the rules/processes/etc.
- Show a simple example application build/packaging process
- I need practice, (I'm a 61 year old retired engineer a bit out of touch)
- Documents that have numerous links to other 'much' lengthier documents are very difficult to follow.
- Packaging Conventions/Overview The process to get a package (of my own App) into Ubuntu (Debian)
- Create some educational materials that can be taught.
- Translation to Spanish.

- up to date, reproducible examples of different aspects of packaging - packaging from scratch, updating a package, fixing a bug, integration with lp, general workflow of a motu
- MSDN like
- Updated tutorials, videos, courses...
- I know how to code in Java. But packing Java apps is hard.
- Much outdated documentation is around, in the Ubuntu wiki similar info is duplicated or important information hidden in different pages (but it improved a lot, e.g. with the restructuring of the packaging information).
- Screencast tutorials. While the IRC logs from developer week and similar events are OK, if the presentations there were done as screencasts, they would be a far more useful resource (something I'd come back to again and again).
- The official Ubuntu docs site still has the old look. It should be more modern and more inline with the current look of the OS and other Ubuntu related sites. I think the info is fine and well organized.
- Documentation should be either within the code and deliver a technical overview and description or in a easily findable wiki page, directly linked from launchpad. So if you want to contribute you can read through that documentation and afterwards know how the software is structured and build etc.
- I was recently asked to apply a patch after reporting a bug. I had no idea where to start looking for a how to, beginning with installing the needed dependencies.
- i'd like a "simple" step by step tutorial of how to package simple python scripts to debian packages, fill all the fields in all the files the deb packages requires including how copyright should be written and how copyright works, where should the files be saved etc.. A tutorial that would explain making .deb's for ubuntu in all aspects. Just after we have a package can we start thinking how to get it in repos
- an usable gui-desktop don't need docs, please follow the gnome3 way!
- keep help.ubuntu.com up-to-date
- Some documentation is out of date. Or there are duplicates of up to date documentation.
- nothing special,
- The documentation is pretty good actually - the root problem with packaing is that the DPKG process has so many tools and options around it that it can seem very complex an intimidating.
- How debian packages differ from Ubuntu
- I think the GTK, PyGTK and glade resources should be more and improved, maybe the Quickly resources too. Also the Vala and C++ resources.
- That it takes tons on hhd space, and takes forever to get enviroment up and running and debugged BEFORE you can actually do a tutorial, wish it was easy like tryruby website and a single deb or package to install via app store
- The Ubuntu Dev site is already a rather good step in the right direction.
- perfect documentation wiki

- The search facility on the Ubuntu Forum doesn't always pick up the best answer to the problem as one of the top hits. I don't see this as a major problem, it just means clicking through to the second or third page for the answer you need! :-)
- -There is on overall listing of information. -The information is either for beginners or for the developers. There is no inbetween.
- Support for other languages. Both in terms of literally provide documentation in alternate languages and in improved support for interacting with communities and software issues arising from international languages.
- Assuming that this question is about the whole development process: the tools need a lot of work! (pygtk, glade, quickly, gstreamer, pylutter, ) I would like to turn it around: a good open source documentation is e.g. here: <https://docs.djangoproject.com/en/1.3/> It has: - a tutorial that works from start to end - a tutorial that does more than list the code - a highly structured general documentation, sometimes even divided into designers and programmers, advanced topics separated from basic topics If I compare this to e.g. starting to program with quickly, there is no documentation at all. Not a website, no documentation, no nothing. Somewhere on the web are old tutorials, that do nearly nothing. (I have picked quickly as a negative example, mainly because I use it and care about it. The software is good, the documentation not existent.) Then there is e.g. glade. If you look at the documentation here: (from a link on <http://developer.ubuntu.com/resources/tools/graphical-designers/> <http://developer.gnome.org/glade/3.10/glade.html> you can see a documentation that has a single screenshot of glade, everything else is text. There is a description of the elements that can be seen in the screenshot, but not a single connection can be made between the text and the screenshot. If one assumes that the user does not know a priori what the "Inspector" is (see 2.2), how should the user know it after reading the text? It should at least look like this: <http://ubuntuone.com/085P8yMm40pSJNu1AfZZ2f> And then the next step would be to write something actually useful: e.g. how do I add an object to a signal and how do I receive this object in my python code? Feel free to try this. Then try to find some information about these steps.
- I really think there needs to be a Python tour of the key Ubuntu platform pieces (Unity, notifications, indicators, etc). It should be done using Sphinx (standard Python documentation tool) and should present the platform through a series of representative code examples... with screenshots of the UI result when appropriate. There should also be an API references of the same, with more details and fewer examples. I think each should be a single HTML page... which forces them to be well written and to the point.
- Documentation can only guide you to success, and that is always not the case. Additionally comments from people indicating issues here and there could help
- Creating deb's are a pain. Better tools needed.
- Although it's an advanced topic, the section on packaging shared libraries and creating more than one binary package from a single source package could use some love.
- A number of wiki pages are outdated, and deal with Ubuntu 9.10, when we are currently at 11.10.
- A centralized place for docs, tutorials etc would be great. [developer.ubuntu.com](http://developer.ubuntu.com) is awesome needs a lot of content.

- If we had a developer documentation like the Qt one, we would be more productive.
- Centralization of resources, expanding developer.ubuntu.com with practical (and simple) examples on how to build different kinds of applications, and how to achieve certain commonly required tasks. It's important that those examples wouldn't just list what code you have to use to do something, but also what that code does exactly, so beginning developers can use them in different ways as well.
- python-gtk3 (pygobject) packaging a debian file
- Lots of related wiki pages not linked to each-other (e.g software updates and management, software center, etc), no Search Engine Optimisation, thus hard to find.
- Overall i was pretty happy with it, i think the major issue for me was finding wiki pages that had clear examples and explanations, rather than vague or old information, some topics i looked up are notoriously difficult to find good info on, i dont remember the specifics though.
- Localization
- Wiki pages
- I followed the examples but didn't know where to go from there. The paper cuts project is great, I'd like to see something like a step-by-step solution for an already fixed bug, just to see what needs to be done. I was interested in fixing bugs in unity, but the code base is just too big to get into it in the limited time I have to offer, so I merely reported the bugs I found instead of also trying to fix them. It would be nice if there was a wiki page roughly describing the different parts of projects like unity and what files belong to them.
- Out of date documentation is a bugbear. Also lack of examples on documentation.
- tutorials with code samples are always great. When tutorials are not (yet) available, target-oriented examples are great; but they should as simple as possible. Example: Howto program a evolution address book via dbus, Howto program a panel plugin to display something, eg cpu usage or whatever simple illustration, ... Links to deeper information about the topic would be wonderful. On my opinion, you can find a nice example for a collection of top-oriented samples within the Apache wicket project: <http://wicketstuff.org/wicket14/> OK, because they are web application samples, you can also directly try the samples online :-)
- More focus on actual development and not just packaging, etc.
- I think you need to separate the concepts: development and packing and publishing. Development: there are a number of options for development, but none of them are great to work with. I've used Quickly and QT. Quickly okay, but compares really badly with development on Windows or Mac where you have a set development environment and way of working. QT was much better to work with, even though I've done very little C++ before. Having an IDE to work with is important - gedit and Quickly really isn't good enough. The IDE can help with the development process. Also, having tutorials that walk you through creating useful sample apps with the IDE is essential. Packaging: it feels like a black art. There needs to be a much simpler way of building a package. Ideally using the IDE. Quickly helps a lot with this as it does most of the work for you. Publishing: once

an app is written and packaged, where do I go from there? How do I submit it to be included in the Ubuntu repositories? I still don't know that answer to that question.

- The documentation needs to be updated, well-written and succinct. I don't know how many times I've looked into the packaging procedure, and I'm still as confused about it as when I started.
- Ubuntu development for dummies :) (start was the hardest: could not find information just after a lot of searching)
- more examples
- I am missing an easy step-by-step "howto" or guide. On the one hand for the installation of the development environment and on the other side to fix a mighty bug
- More informations about IDEs and libraries needed. About library documentation itself ubuntu should host also the docs of all the main libraries it's using (basically mirroring gnome.org) with a docs portal site with better template using AJAX to easily allow to search in all the docs. So basically we need something like the qt site is doing for its libraries.
- some real world example (take an app (with gui and all), get sources, modify some code, repackage, test, publish fix, ...)
- PACKAGING PLEASE.

***If you have any additional suggestions to make Ubuntu development easier to understand or more exciting to learn about, please list them here. If they are unconventional, please let us know anyway.***

- \* Bug-fix sessions (pick one bug that seems easy, show how to fix the bug (maybe screensharing or sth like that)) \* List of easy bugs (the harvest bugs seem still too difficult, also the bitesize ones)
- Make the Red-Hat script (makefiles) work on Ubuntu, and if necessary document them. Make sure that the standard makefile (gconfig, xconfig. install etc...) work on Ubuntu as well as on RPM based distros.
- invite local teams to help out to spread the word, maybe #ubuntu-motu-es, #ubuntu-devel-fr and so on. Think locally and make clear that contributors don't need to be genius to send a patch correcting a typo
- worldwide distributions with another run of shipit
- Quicken, Ubuntu's apparently preferred way of app development, which is Python/GTK+ etc based would do well to have a GUI element to make the whole experience like an IDE. Having to type quicken [whatever] into a terminal smacks of all that is wrong with linux in the modern age; namely anything that requires going to a forum/wiki, then copy/pasting some 'magical spell' into the terminal to get linux to do what you want of it when the exact same thing could be done intuitively and with a simple touch or mouseclick in a GUI.
- I think the basic installation of Ubuntu should always have a clean, basic and light configuration for the desktop environment. Unity has lots of potential, but users still need to invest (may say waste) a lot of time to learn how to use it properly. And a lot more to configure it effectively. The option "Ubuntu Classic" in 11.04 was great, when I was working logged in Ubuntu Classic, when I was willing to test and learn Unity, I would log to Unity.
- I have a lot of experience in SW programming (mainly C, C#, Java), but do not find the right open source project for "a start"...
- Hi! How about having some program for IT beginners like me. pls let me know. Dwd (gqassem1@yahoo.com)
- easy way to just start helping even with no experience
- More official video tutorials would be helpful. Collaboration with local institutions e.g. schools should be strengthened, different seminars organized.
- I think a co-ordinated doc is really really important!
- Release the WebLive to actual Ubuntu, and hide some new features in alpha for surprises like APple or Microsoft does, seriously, I used to be excited about ubuntu releases, but following these news sites makes it soo predicatable lol
- Somehow irrelevant, you extensively use vala and Gnome libraries for development... As much as I want to start some development work on Ubuntu, and I have tried it many times, I always quit with frustration. I by no means am willing to spend the slightest time learning languages (e.g. Vala) and libraries (e.g. Gnome libraries) that I am never going to use in my professional life, simply

because there are better alternatives. The use of Python in many parts is great, but I am more into high performance work, so C++/C looks like a better fit for what I am willing to contribute. I understand there is great dependence, built over the years, with the Gnome tools and libraries in general. But, seriously, Qt is by far better in many respects. For one, it is more fun to write, excellent documentation, industrial specs and official top-notch support for other operating systems... I use Qt libraries for work, I am willing to contribute but by no means willing to learn programming using the Gnome libs, or vala... What's the use of it? Qt is better, in many respects. Its GPL now, so I think its time you broaden the candidate contributors by welcoming Qt developers as well... Start development for some of your new in the house projects using Qt libs and C++. Broaden your potential contributors group. And seriously, I can see how vala accelerates development using Gnome libs, but seriously, it just further alienates potential contributors. I for one, unwilling to learn developing for Gnome, tried simply contributing in parts where graphics were irrelevant... I have tried that many times, but found that the pertinent libs I was interested in were written in vala, so off of that as well...

- The easiest way to get into anything new is with a mentor. I think a video with clickable links (to make it more interactive) would be a great introduction and orientation. From there it's not so hard to build on, but the initial interaction has the most potential to be a turn off because of disorientation.
- Well, I don't want to understand the whole thing - it's the little things that matter! So could you make it possible to give me just one small piece I could contribute? So I could become an expert for just one thing. And that's it.
- When I search for documentation of development for ubuntu, I often end up on strange pages like this: <http://developer.ubuntu.com/api/ubuntu-11.04/GIR/python/Unity-3.0.html> This is the first ubuntu-unity related hit when searching for 'python unity' with google. Since stuff like this is far from interesting, it should maybe not be searchable by google in order to keep those pages from showing up in searches.
- Make it a bit easier to become a developer/patcher etc, or at least have access to dev info. A clear link from ubuntu.com would be nice. It would also be nice to have a link from ubuntu.com to info about the upcoming release (schedule/features/progress). It is currently absent or hard to find.
- You should make an official blog for development progress status to make that the average user could better understand changes which happens "under the hood". There shouldn't be too much boring informations understandable only by geeks.
- Canonical should stop rolling their own version of everything. The Gnome docs are pretty good, but it's hard to track down docs for Appindicator, the menu objects, etc.
- Provide documentation for beginners to start learning programming languages, the current developer.ubuntu.com site includes no real information about learning Python, even though that is supposedly the preferred language.
- Sometimes beginners can get lost in launchpad. some duplicates can be hard to find.
- I try to help development now and then by reporting bugs or shortcomings on Launchpad. However, it often takes ages before a response comes. By that time

I've usually moved on to a different package which does what I wanted, or have found a workaround for the bug, or even installed new hardware. Anyway, after that long I've either lost interest in the report or can't remember/replicate what was amiss in the first place. Result is a bug report that gets ignored because of a lack of attention from both sides. If a/the developer could respond within say, two or three days, I could provide more information and help development that way. I've had that happen once. The developer responsible (I think it was a bug in gnome-control-center) replied the same day and she made a bugfix release the next day. That was really cool and made me feel like I had helped a little in making Ubuntu better. Most of the time though, a reply comes after weeks or months or never, which makes me feel like reporting bugs or shortcomings in Launchpad is boring and pointless.

- consistency. I think there should be more consistency overall. Gathering up all settings in a one place as 'System Settings' is a good thing. And most of all customization. It would be great if ubuntu has a customization tool such as ubuntu tweak out of the box. because customizing really small details is very important some times :D
- A blog of an Ubuntu developer who is doing his thing with screen capture. How he fixes simple bugs, packs software etc. Fun, gives an overview on what is generally done, gets rid of mysteries and what not.
- Make some awesome development guides like guides.rubyonrails.org and put them on the web and in the software center and then keep them up to date. That would help bring down the barriers to starting, which are usually the biggest barriers.
- The current documentation is really good, and gives a clear overview of the Ubuntu development process. There's one thing I still haven't figured out, though: as someone that isn't involved in Ubuntu development, my main question is not "how do I get started?", but "When?".
- just how to's, even for the most trivial tasks.
- A real DevGuide like the one from developer.android.com. Maybe instead of relying on the community to submit documentation, hire a team to do the basics. If the basics are good then you can start your project but now its just impossible to program a single complete application with the developer site except the few complete tutorials.
- Use git. Everybody else does.
- more sex?
- Ubuntu development itself should be a no-brainer. Our tools and packaing is much more complex than iOS/Android's. We have no 'One true way' in almost any aspects of our development process. We waste too much time tending for packages and stupid build bugs instead of making the step from writing an app to shipping it in Ubuntu almost inexistent. We need to rethink the whole Debian/linux distro legacy so it does not prize diversity so much anymore as it comes with a huge cost. Half-assed tools that automatically generate code/packaging are not enough either. If something needs too much automatic generation and boilerplate it is too complex already and needs fixing.
- Prepare a liveCD with all tools ready to use and a easy way to download last code.
- Make developer.ubuntu.com also the starting place for reference manuals and

platform overviews.

- There are so many languages. In terms of programming often bash is easiest just to get the job done, but to get something slick it would be nice to have more examples, covering all aspects of the development process. There is so much stuff out there, but it's often out of date, on so many different languages, and for 'normal' people a bit of spoonfeeding/handholding doesn't hurt. In terms of GUI development, it's really limited.
- A brief overview following the packaging guide would be nice.
- I've not looked in the last year, but the old MOTU/packaging documentation was off-putting because it seemed more like branching lists of references to lots of very long pieces of documentation (e.g. Debian New Maintainers' Guide), so it felt like there was a forest of material that ought to be read before you could start doing anything. Training sessions on IRC have allowed people to get set up for packaging and try out some simple packaging tasks, but they are fixed in time, and "waste" a developer's valuable time. There is also no clear path to follow afterwards in order to progress. I would like to see a structured set of educational materials to train people, with a clear path to progress along. Providing a personal progress chart might help motivation and encourage people to persevere. It would track progress with personal milestones such as "Created GPG keypair", "Signed Code of Conduct", "Created Launchpad account", "Installed packaging toolset", "Created first package (HelloWorld.deb)", "Created my first PPA" etc. It would also allow you to check off each section of each piece of documentation that new contributors should be reading. This would create a sense of progress, and help people keep track of what they have and have not yet read. It would be an interactive training system using Moodle or similar which would allow people to practise basic tasks to begin with, such as installing tools at the command line. Then the student could perform basic packaging tasks with specially prepared tarballs etc. The packages created would be uploaded to the training system and checked by machine to ensure the packaging had been done correctly and provide feedback to the student. More complex tasks which cannot be machine marked could be checked by humans. People who are confident can ask for advice and help in IRC or elsewhere, but the less confident among the population of potential developers may be reluctant to do this. People with technical potential may be missed as contributors because of this. Making self-tuition easier by providing a more structured, clearer path may allow such people to gain the confidence they need to be able to ask later for help directly. Also, developer time is valuable, so the more automated the early training of contributors can be, the less time will be wasted in mentoring at a basic level. Logging people's progress may also clarify at what point people hit a stumbling block and give up trying to learn packaging or development. This information could guide further improvements in documentation, mentoring, or other support. Mentors could even pick up on people who seem to have stalled, and contact them directly to see if they can help get them back into learning to package/develop, or find out what went wrong.
- UDD is nice for merging but kind of a pain otherwise, especially with some sponsors being opposed to it and the quilt/.pc problems/inconsistencies
- A simple guide to new comers (or more information if it already exists)
- Maybe there should be a mechanism to define and categorize your audiences' technical knowledge and then have the documentation written accordingly.

Regardless of the amount of effort one puts into development documentation, if you address the wrong audience, all the effort would be wasted.

- 1. A better developer tool story. Quickly is a great start, but still feels a bit rough around the edges. 2. A more vibrant financial eco-system. The Software Center is looking better, and the cloud/OpenStack work is promising. Keep going there. 3. Better documentation. Much of the docs in the wikis and such are out of date or incomplete. See the docs as [djangoproject.com](http://djangoproject.com) for an example. 4. More developer-oriented tutorial screencasts (yes, I know I'm repeating myself).
- I'm not sure right now, but if there is no instruction of how to set up a ppa and building packages with patches - that would be great. And: Big thanks for this awesome product!
- follow the gnome3 way, please.
- I think in addition to Ubuntu development, further encouragement of development of the upstream components that make Ubuntu great should be promoted. Perhaps great applications can be highlighted with information about how upstream help can be provided.
- The most challenging thing I can think of is a complete reboot of the packaging system. Work to make the most common cases simple, obvious and consistent. Use and recommend as few tools as possible. New Ubuntu packagers are faced by a myriad of programs and files. By contrast, Arch Linux achieves most of the same set of functionality from it's package management system whilst ensuring that most packages require just a few minutes editing 1 file and then running 1 program (makepkg).
- I think we need more tutorials as the ones in [developer.ubuntu.com](http://developer.ubuntu.com), about packing .debs and other useful things.
- I think there must be more android support.
- Simple overview & graphical chart to show ex. 1) installl blow 2) run this dev tutorial 3) run this packager 4) upload to this Developer Store (even if apps were deleted after 7 days or something 5) application to put in real app store 6) do blow to put your app in the store
- This place isn't for noobs.. And there needs some kind of handholding and small projects people can do easily and well..
- Maybe a "Baby's First Steps In Ubuntu Development" type of article. I'm interested in getting into development, but I'm having difficulty finding an article about the first steps to take in getting there.
- It seems like a good idea to focus on a single development process: quickly, python, gtk, gstreamer, clutter, ... There should be documentation on [developer.ubuntu.com](http://developer.ubuntu.com) site about this workflow. Quickly: needs a tutorial that is always kept up to date. pygtk: documentation is currently not good, since there is no documentation for the future pygi version. This needs to be written. pygstreamer: documentation is a tutorial and a nearly useless API documentation: e.g. <http://pygstdocs.berlios.de/pygst-reference/class-gstbin.html> (as a test: please explain the recurse method: <http://pygstdocs.berlios.de/pygst-reference/class-gstbin.html#method-gstbin--recurse> and as a reference a good documentation try to explain the `render_to_response` method: <https://docs.djangoproject.com/en/1.3/topics/http/shortcuts/#render-to-response>). SO: there should be a useful gstreamer api documentation (clutter:

there is no python clutter documentation (<http://wiki.clutter-project.org/wiki/PyClutter>). Needs to be written if it should be part of the ubuntu tools.) Python and bazaar have very good documentation.

- G-I generated documentation doesn't cut it. If Python is going to be the "preferred" language we direct new developers too (and it should, IMHO), it's important that the Ubuntu documentation harmonize with the standard Python documentation. It's fine if the Ubuntu documentation uses custom Ubuntu branding (I think it should), but it shouldn't stray too far from the layout of the Python docs. The writing style needs to closely match the Python docs.
- Provide newcomers with common pitfalls and solutions, Also provide more interaction on the MOTU Mailing list. Little emails land there and it is a great way to archive issues in time. People do not use the Mailing list enough and probably it is because they do not feel encouraged to drop a line
- The problem with Ubuntu development is that, most of the time, I'd rather be packaging generically for Debian. Ubuntu should focus less on packaging and more on bug fixing, stability, and community involvement in the road map. A specific suggestion for improving the packaging experience regards the PPA system. It would be nice if, like other packages, PPA packages didn't need a different version string for each 6-month release. Another package-related suggestion is moving to a rolling release. As a developer, using the LTS release is out of the question but having to reinstall my system every 6 months/wait up to 6 months for a minor version change in a library to be packaged is really annoying.
- We should define stable API to do things, then we can also eventually change the implementation from a release to another but API must remain stable as long as we can. For example: using the notify area, interacting with Unity objects, ecc.... we don't have an Ubuntu API at all at the moment, just a bunch of libraries and many difficult to find documentation and examples.
- i always wanted to see a site that teaches by examples, starting simple and small stuff, then going up to harder levels later, i have this website that i like, if you've heard about it's called 'hackthissite.org'
- I would really like to see the gap between a full blown programmer and a beginning programmer become smaller. I think using a method like Game Maker (for Windows), which simplifies making software (in the case of GameMaker limited to games, but it could be extended to include all kinds of applications), by simple dragging and dropping elements, and allowing beginning developers to easily create good looking programs. This stimulates new developers to keep going. I think Quickly is a good start, but it needs to become even easier to use (a GUI is a must).
- Make developer.ubuntu.com the signal site to go to and work as a portal, so from this site you could access all documentation and forums needed.
- The getting started with ubuntu development blog post series was a good start in structuring the development workflow. Maybe a throughout wiki page with all the parts linked in/included would also help. Another thing I would like to see would be a more detailed description of the Ubuntu boot process and libraries used in it. This would help people to identify their target contribution areas: e.g. boot process (grub, plymouth) for hardcore-junkies, package management apps (apt, aptdaemon, software-center, update-manager, software-properties, gdebi, synaptic, etc), desktop interface (unity, lenses, nautilus, control center, indicators,

global menu) and so on. I have spent some time playing with everything I could find and fix until I found out that my favourite area is package management.

- I would like to see development release as a main stream ubuntu. I think software freeze is outdated. looking for a rolling release.
- Dont have any really, would just like to see the existing docs improved more. (who doesnt?)
- Irc classroom, Video tutorial explaining step by step packaging and some easy program since initial workflow to working app.
- it must be easier to file bugs. As a non-technical user I often find myself incapable of filing bugs, since I cannot gather the required information. It should be possible to send in key-settings and a description of the problem. Not a super-technical report about installed packages.
- not really, developer.ubuntu.com looks great, I hope it works :)
- Even I'm a experienced programmer, I don't have much experience in community development. For me it's hard to find a easy entry into doing a contribution. It is hard to have an overview which projects would need people for what tasks. Even trying to fix or locating bugs is often not really easy. I already tried to contribute a little bit around modem-manager (Problem with USB-link of the N900) or gvfs (copying files in Nautilus via ftp didn't work very well for a long time). I know, that're a bit more upstream topics. I think it would be great if there would be a list of developers around different projects, distributions, or whatever. So , I would have a chance to see who is living in my city for example. Assumed he or she is willing, I could contact him or her personally for brain storming, what could I do, where can I help... I know, there are a lot o opportunities in Berlin/Germany. But I live in Nürnberg. I have no idea if there is anybody who is programming for Ubuntu for example and what he or she is exacly doing. Maybe there are some, but I simply do not find information about them.
- The merging with upstream projects should be made easier - the community is too much fragmented thus bugs remain open in Ubuntu.
- More tutorials!
- I think there needs to be a recommended development environment that is nice to use, well documented and simplifies development and packaging. I prefer to program in Python, but I don't think Quickly is the right direction. I think that QT Creator is a great IDE on Ubuntu, and I enjoyed using it even though I don't know C++ that well. Being able to develop a QT app in Python in QT Creator would be great. It would give a more polished feel to development on Ubuntu, and make it easier for users transitioning from Windows or Mac development (like myself).
- Ubuntu development for dummies :) (start was the hardest: could not found information just after a lot of searching)
- Missing of Tutorial
- each time I came in contact with my loco team I not really felt welcome. Sometimes there is an aura of "closed circle" more than if I was in contact to canonical guys... and girls....
- Tutorials should be posted for beginners in open source developers..