

Debian Med sprints and mentoring

Andreas Tille

Debian

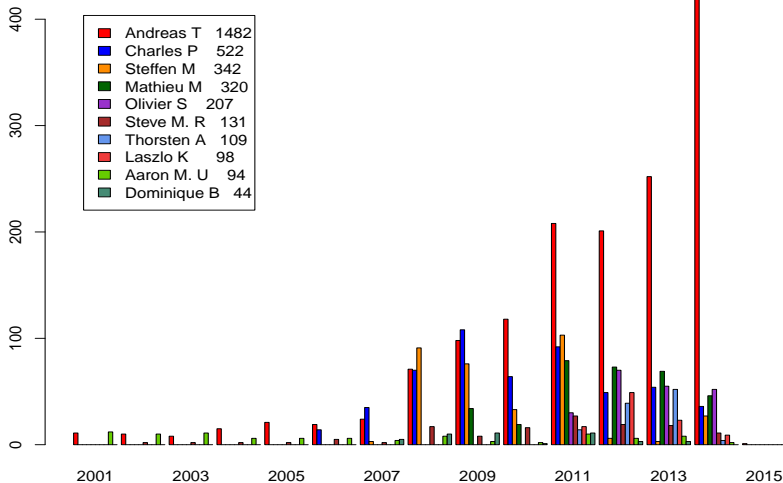
Saint Malo, 31. January 2015

Face to face meetings with users and upstream

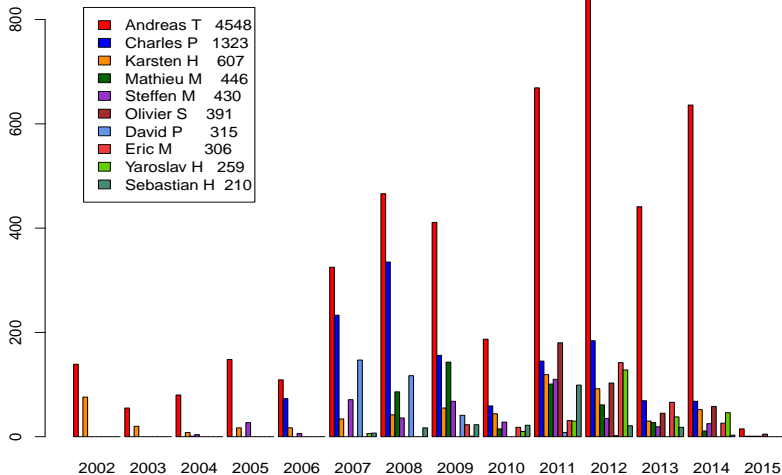
Have a quick look at what happened since 2011 when we started doing yearly sprints by also inviting users and upstream developers.

Thanks to Debian for supporting sprints which we will prove to be successful

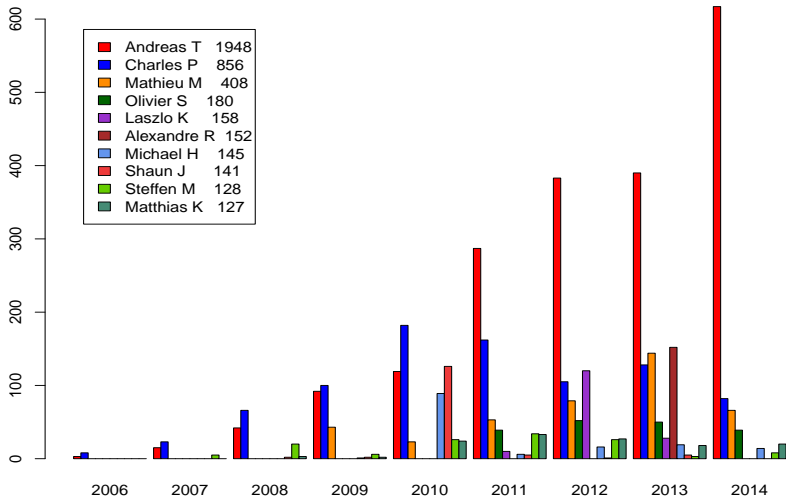
Top 10 Uploaders of Debian Med team



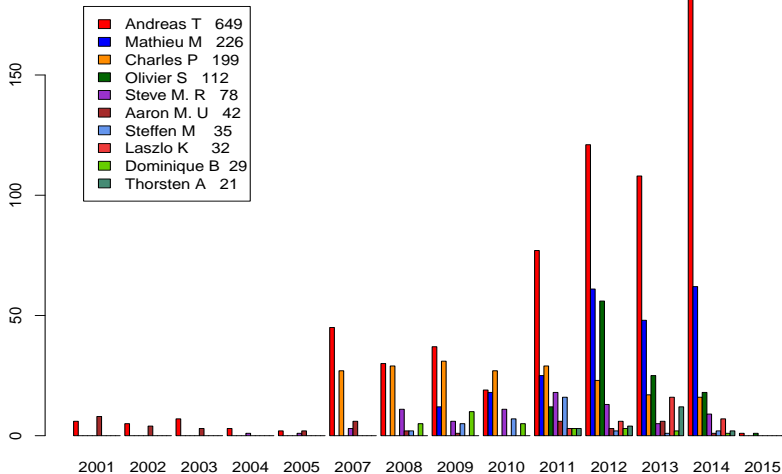
Top 10 discussion on Debian Med user mailing list



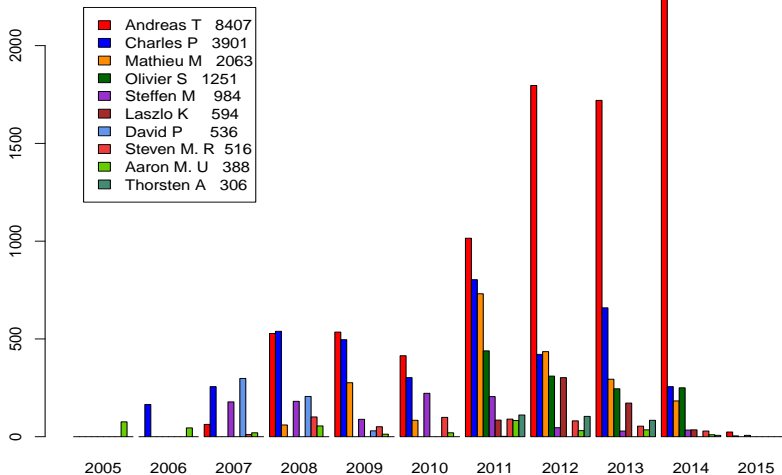
Top 10 discussion Debian Med developer mailing list



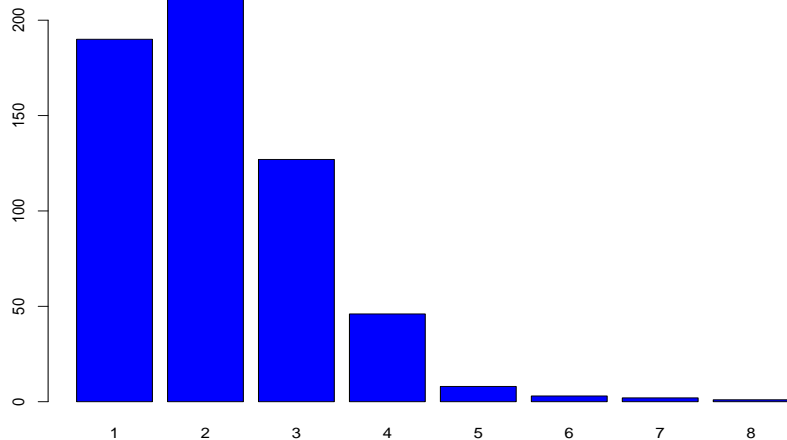
Top 10 bug hunters of Debian Med packages



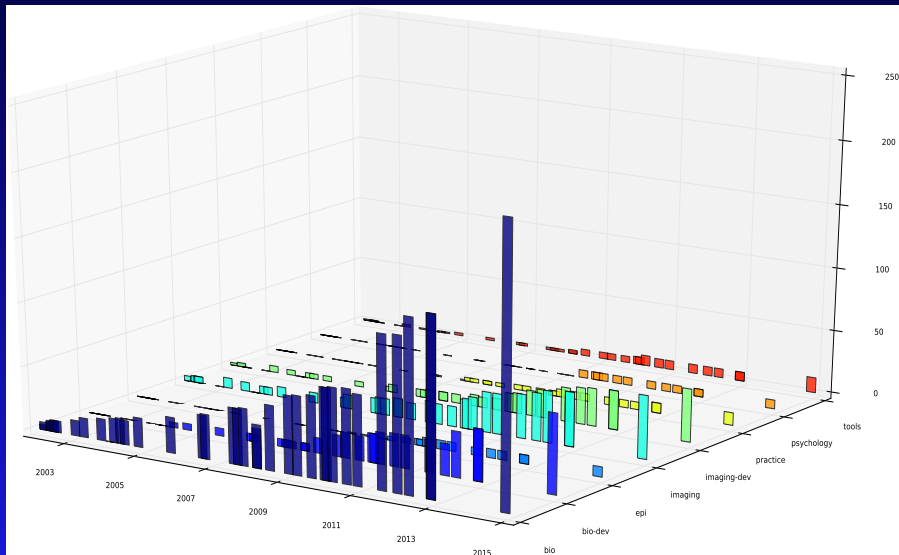
Top 10 committers to Debian Med VCS



Maintainer per package relation in Debian Med



Number of Packages in selected tasks



Inviting environment

In several talks at DebConf and in discussions before I have heard the argument that it is hard to find friends. But it is not. The answer is to create inviting teams.

*Waking up in the morning and realising
that somebody else has solved your
problem from yesterday*

Mentoring of the Month (MoM)

- **mentor dedicates a part of his spare time to a newcomer**
- guide the student kindly into all secrets of Debian packaging
- learning per example
- student is free to pick the package (mentor has a vetoing right)

Mentoring of the Month (MoM)

- mentor dedicates a part of his spare time to a newcomer
- guide the student kindly into all secrets of Debian packaging
- learning per example
- student is free to pick the package (mentor has a vetoing right)

Mentoring of the Month (MoM)

- mentor dedicates a part of his spare time to a newcomer
- guide the student kindly into all secrets of Debian packaging
- learning per example
- student is free to pick the package (mentor has a vetoing right)

Mentoring of the Month (MoM)

- mentor dedicates a part of his spare time to a newcomer
- guide the student kindly into all secrets of Debian packaging
- learning per example
- student is free to pick the package (mentor has a vetoing right)

MoM Goals

- The package in question is finished and uploaded.
- The student is able to do advanced packaging tasks and is introduced into the communication channels of Debian Med team.

MoM Goals

- The package in question is finished and uploaded.
- The student is able to do advanced packaging tasks and is introduced into the communication channels of Debian Med team.

Summary of MoM formalism

- **MoM is *work***
- hopefully it shows that this work is also *fun*
- mentor trades his spare time for the work of the student
- attempt to train "silent observers on the list"
- focus not only on technical details but also on learning to know relevant communication channels

Summary of MoM formalism

- MoM is *work*
- hopefully it shows that this work is also *fun*
- mentor trades his spare time for the work of the student
- attempt to train "silent observers on the list"
- focus not only on technical details but also on learning to know relevant communication channels

Summary of MoM formalism

- MoM is *work*
- hopefully it shows that this work is also *fun*
- mentor trades his spare time for the work of the student
- attempt to train "silent observers on the list"
- focus not only on technical details but also on learning to know relevant communication channels

Summary of MoM formalism

- MoM is *work*
- hopefully it shows that this work is also *fun*
- mentor trades his spare time for the work of the student
- attempt to train "silent observers on the list"
- focus not only on technical details but also on learning to know relevant communication channels

Summary of MoM formalism

- MoM is *work*
- hopefully it shows that this work is also *fun*
- mentor trades his spare time for the work of the student
- attempt to train "silent observers on the list"
- focus not only on technical details but also on learning to know relevant communication channels

Results

Month	Student	Package	Uploaded?
02.2012	Luis Ibanez	fis-gtm	yes
03.2012	Scott Christley	libswarm	no
06.2012	Eric Maeker	libquazip	yes
11.2012	Tomás Di Domenico	python-csb	yes
02.2013	Sukhbir Singh	hunspell-en-med	yes
12.2013	Lennart C. Karssen	probabel	yes
01.2014	Jorge Sebastião Soares	snp-sites	yes
03.2014	Stephen Smith	phyutility	yes
06.2014	Ian Wallace	openemr	
09.2014	Corentin Desfarges	camp	yes
02.2015	Paul Novotny	opensurgsim	

Some people put their names in the waiting queue but did not answered a ping.

Debian Med has attracted one developer per year

According to a questionnaire in Wiki

- Debian Med has 25 DDs+DMs
- 12 DDs *because* Debian Med exists;
4 DDs before Debian Med started
- 8 out of the 12 above extended their activity to other fields in Debian
- 10 out of the 12 above are active in Debian Med

Debian Med has attracted one developer per year

According to a questionnaire in Wiki

- Debian Med has 25 DDs+DMs
- 12 DDs *because* Debian Med exists;
4 DDs before Debian Med started
- 8 out of the 12 above extended their activity to other fields in Debian
- 10 out of the 12 above are active in Debian Med

Debian Med has attracted one developer per year

According to a questionnaire in Wiki

- Debian Med has 25 DDs+DMs
- 12 DDs *because* Debian Med exists;
4 DDs before Debian Med started
- 8 out of the 12 above extended their activity to other fields in Debian
- 10 out of the 12 above are active in Debian Med

Debian Med has attracted one developer per year

According to a *questionnaire* in Wiki

- Debian Med has 25 DDs+DMs
- 12 DDs *because* Debian Med exists;
4 DDs before Debian Med started
- 8 out of the 12 above extended their activity to other fields in Debian
- 10 out of the 12 above are active in Debian Med

MoM Conclusions

- Time spent into mentoring is worth the effort
- No student for each month so the workload is bearable
- Students have just read recent documents which I did ten years ago → I can learn new stuff from them
- Major advantage: training upstream to pool their knowledge about the code with ours about packaging is **very efficient** for the hard packages
- About 50% of students had strong connection to upstream and requests for upstream changes went very smoothly

MoM Conclusions

- Time spent into mentoring is worth the effort
- No student for each month so the workload is bearable
- Students have just read recent documents which I did ten years ago → I can learn new stuff from them
- Major advantage: training upstream to pool their knowledge about the code with ours about packaging is **very efficient** for the hard packages
- About 50% of students had strong connection to upstream and requests for upstream changes went very smoothly

MoM Conclusions

- Time spent into mentoring is worth the effort
- No student for each month so the workload is bearable
- Students have just read recent documents which I did ten years ago → I can learn new stuff from them
- Major advantage: training upstream to pool their knowledge about the code with ours about packaging is **very efficient** for the hard packages
- About 50% of students had strong connection to upstream and requests for upstream changes went very smoothly

MoM Conclusions

- Time spent into mentoring is worth the effort
- No student for each month so the workload is bearable
- Students have just read recent documents which I did ten years ago → I can learn new stuff from them
- Major advantage: training upstream to pool their knowledge about the code with ours about packaging is **very efficient** for the hard packages
- About 50% of students had strong connection to upstream and requests for upstream changes went very smoothly

MoM Conclusions

- Time spent into mentoring is worth the effort
- No student for each month so the workload is bearable
- Students have just read recent documents which I did ten years ago → I can learn new stuff from them
- Major advantage: training upstream to pool their knowledge about the code with ours about packaging is **very efficient** for the hard packages
- About 50% of students had strong connection to upstream and requests for upstream changes went very smoothly

This talk is available at
<http://people.debian.org/~tille/talks/>
Andreas Tille <tille@debian.org>

