Debian Med sprints

Andreas Tille

Debian

Copenhagen, 4. February 2016

Back in 2001 at DebConf 1

the first talk about the concept was prepared

About Debian Med sprints

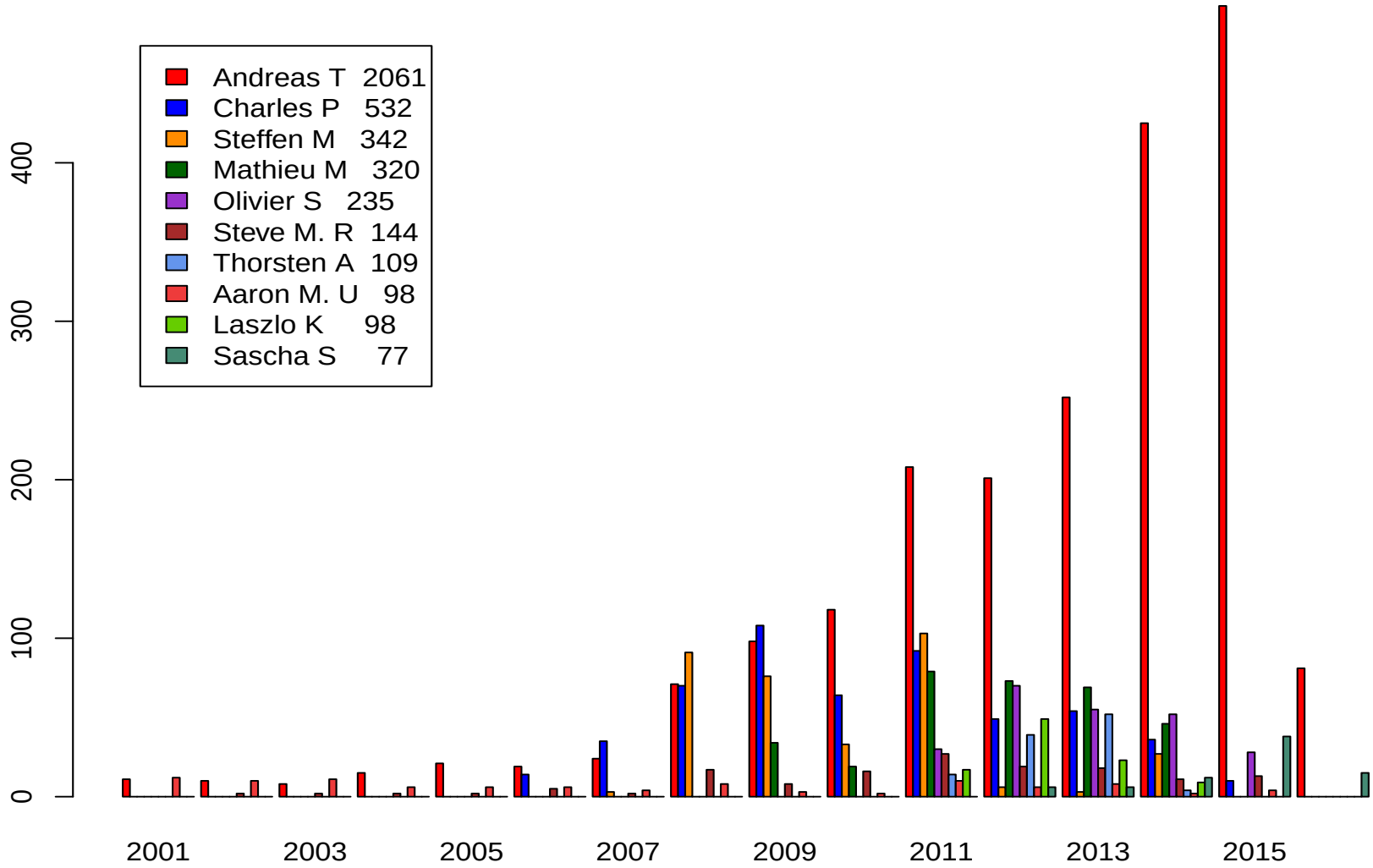
From: Gunnar Wolf <gwolf@gwolf.org>
Date: Thu, 12 Feb 2015 07:13:47 -0600

> My punishment to not have invited you
> is to stand inside the about 5°C
> cold water for 3 minutes. ;-)
>
> Next Debian Med sprint will be as
> usual at end of January on some
> seaside location. You are welcome
> there. :-P

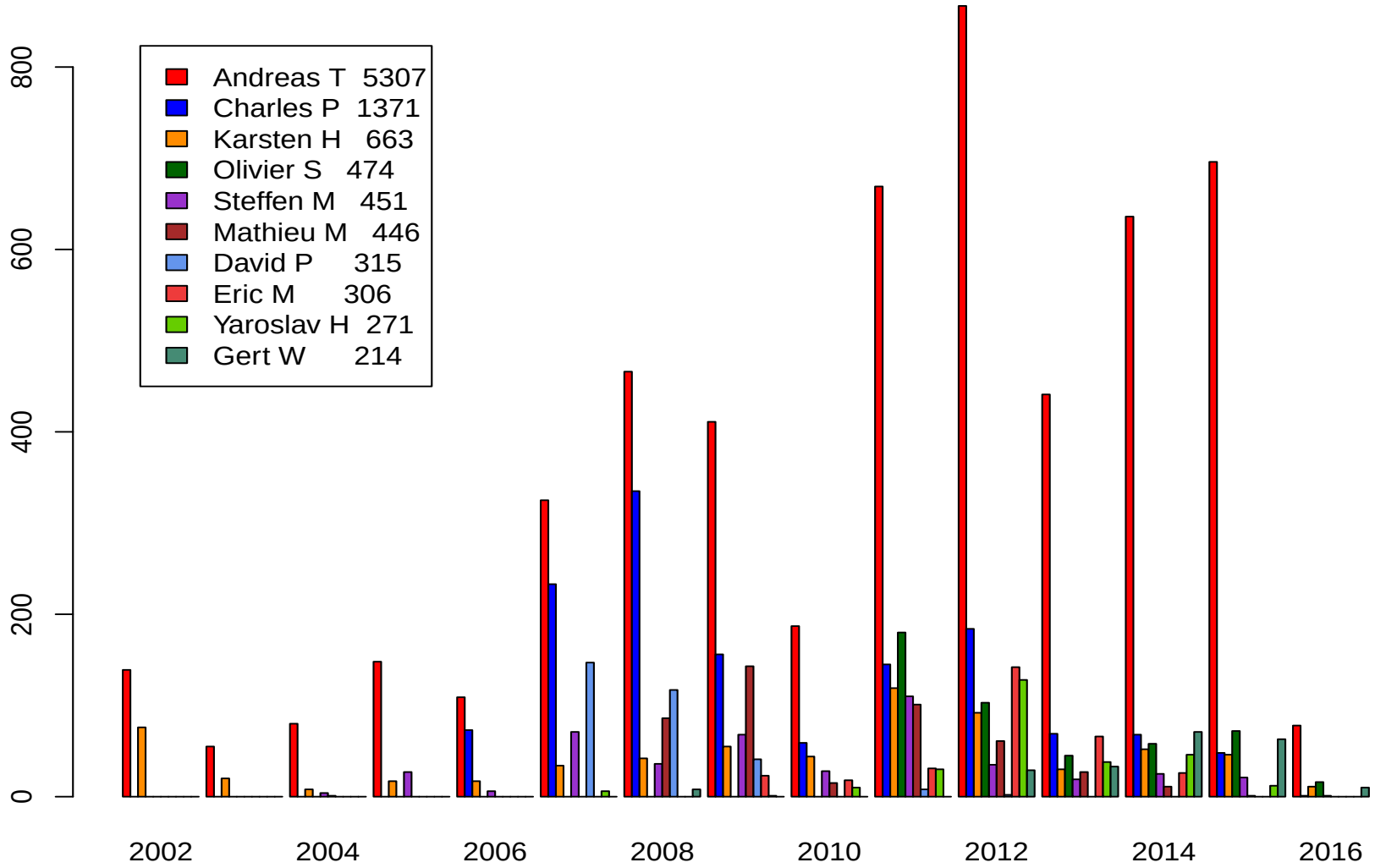
Oh, I understand -- I thought Debian-Med focused on packaging medical software. With this note you mention and your choice of location, however, I understand better: After meeting every year, you need medical attention, possibly including a psychiatrical evaluation.

But anyway, if we forget for a bit the fact that most (all?) Debian-Med members are European, I can invite you (and if you are interested, I'll surely get contacts for a nice, sponsored venue!) for a meeting next january in a *real* beach, with nice 30°C winter sun, of course >=25°C sea water where you will dwell for much more than three minutes and all!

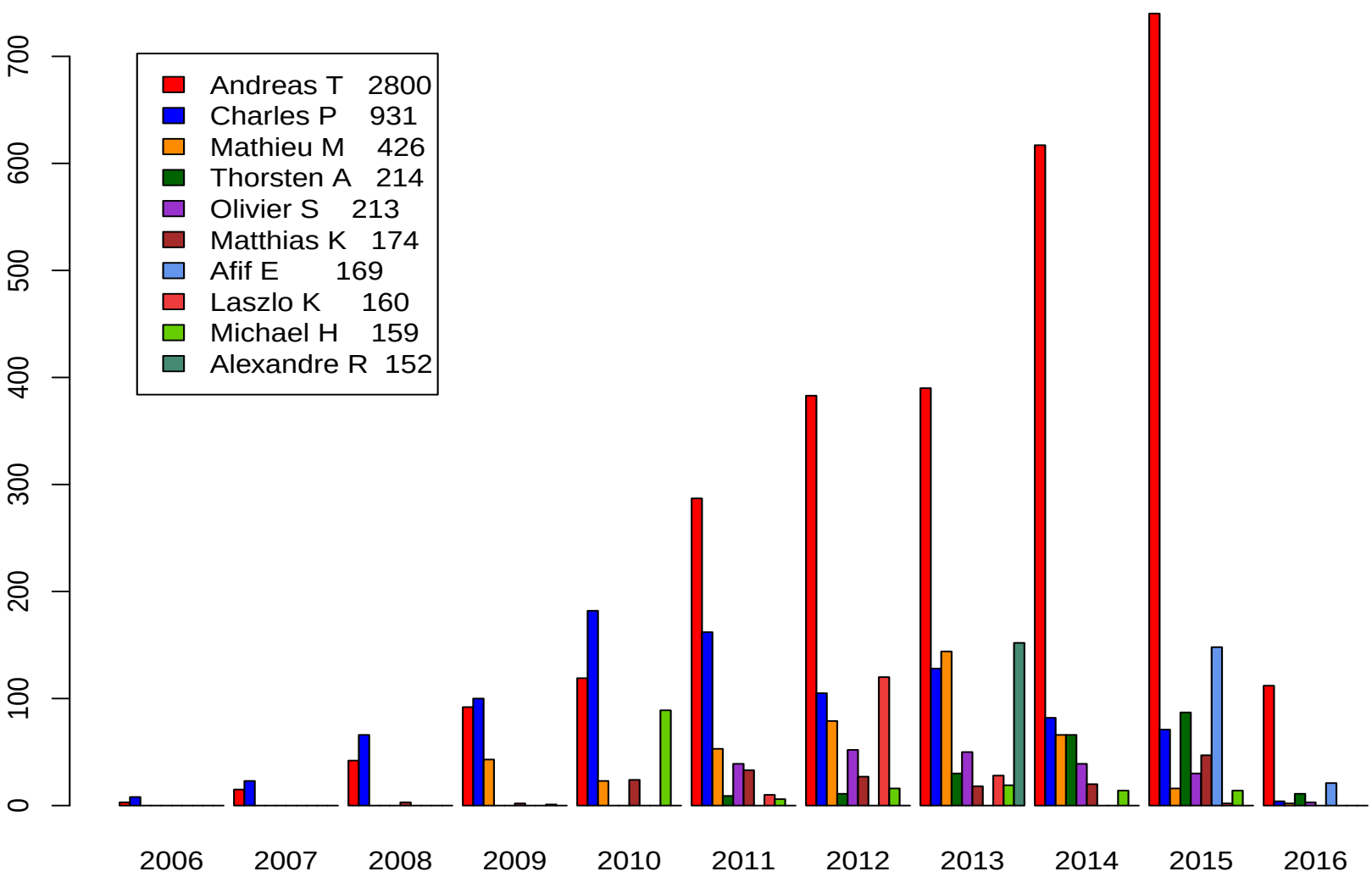
Top 10 Uploaders of Debian Med team



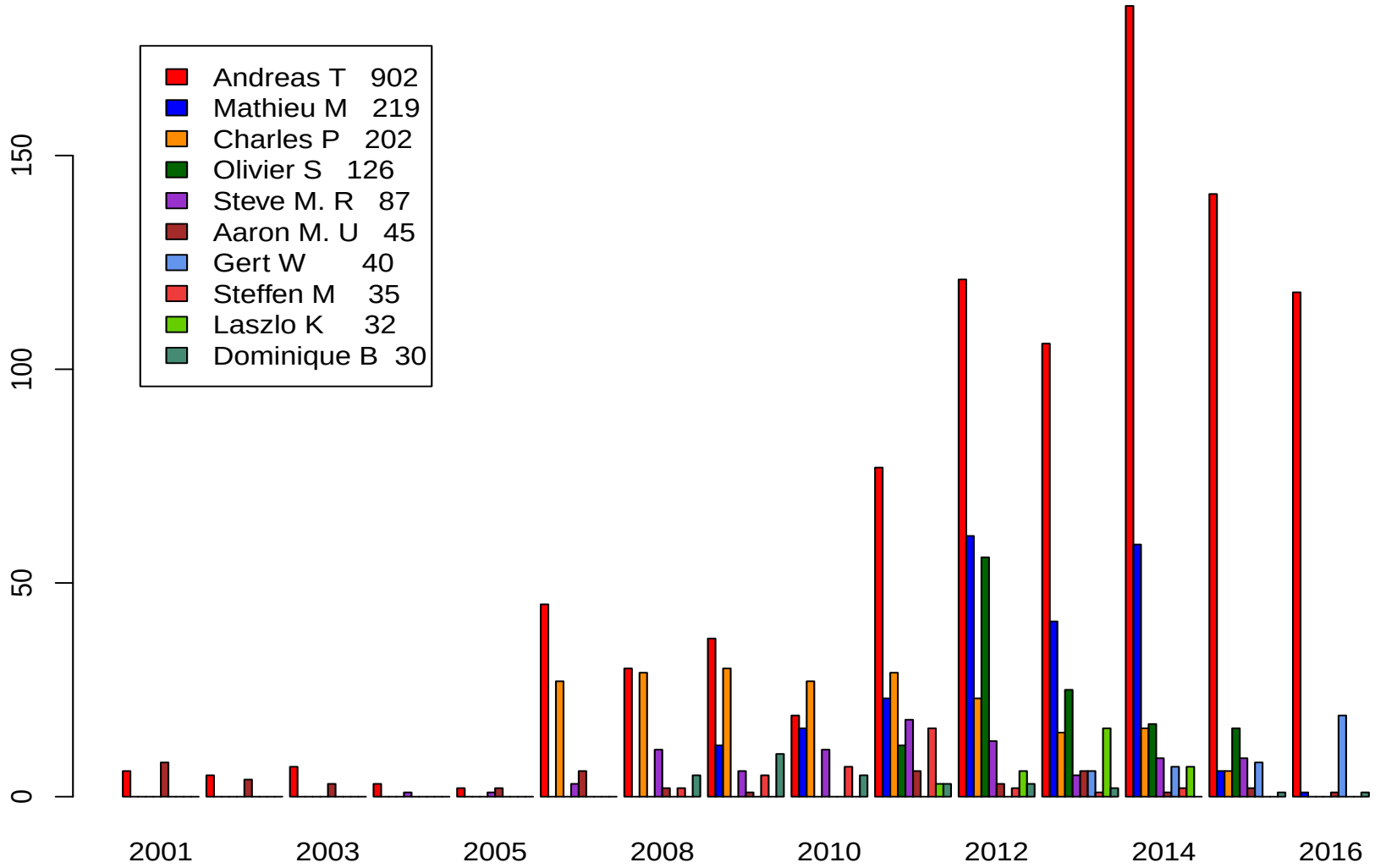
Top 10 discussion on Debian Med user mailing list



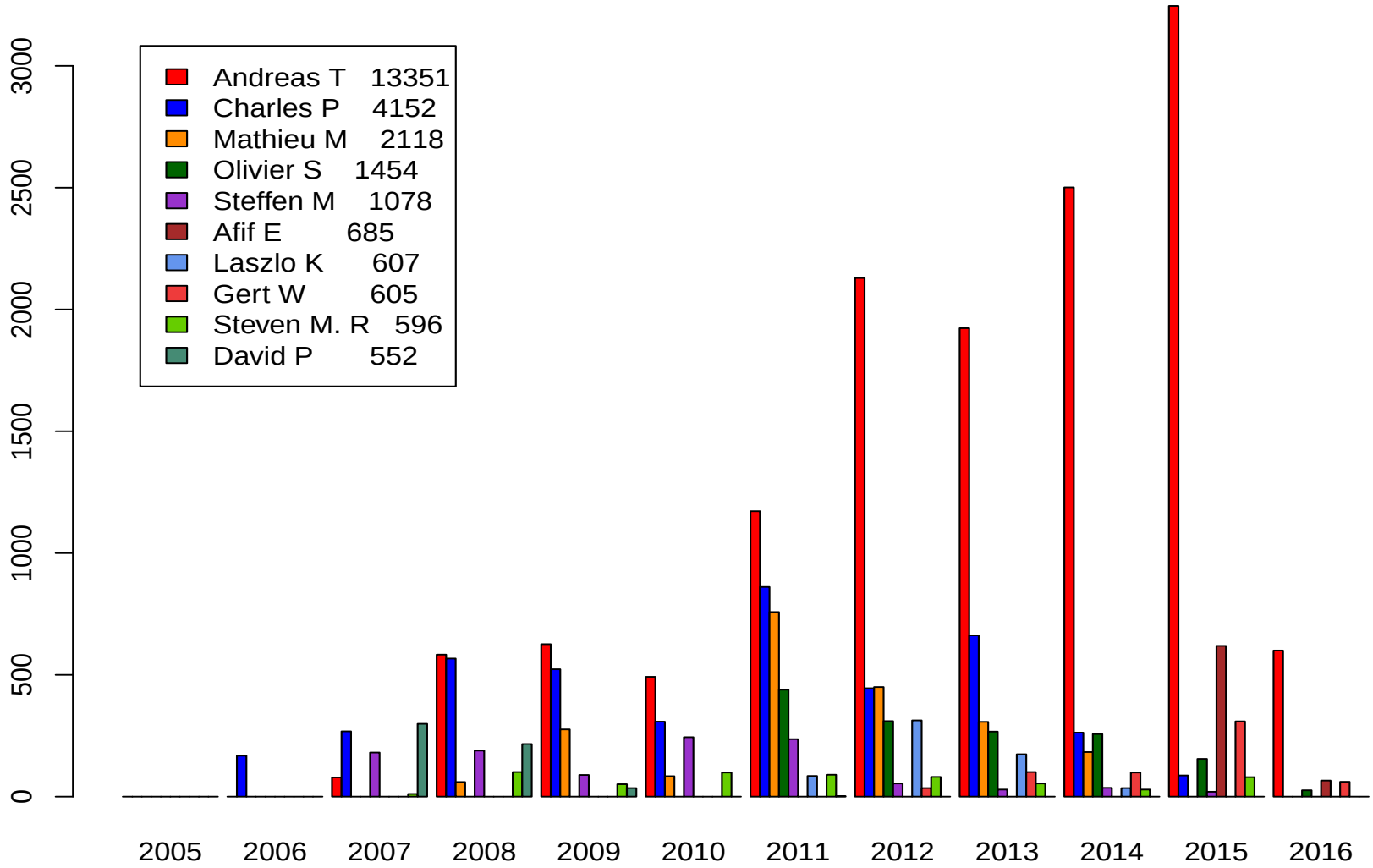
Top 10 discussion Debian Med developer mailing list



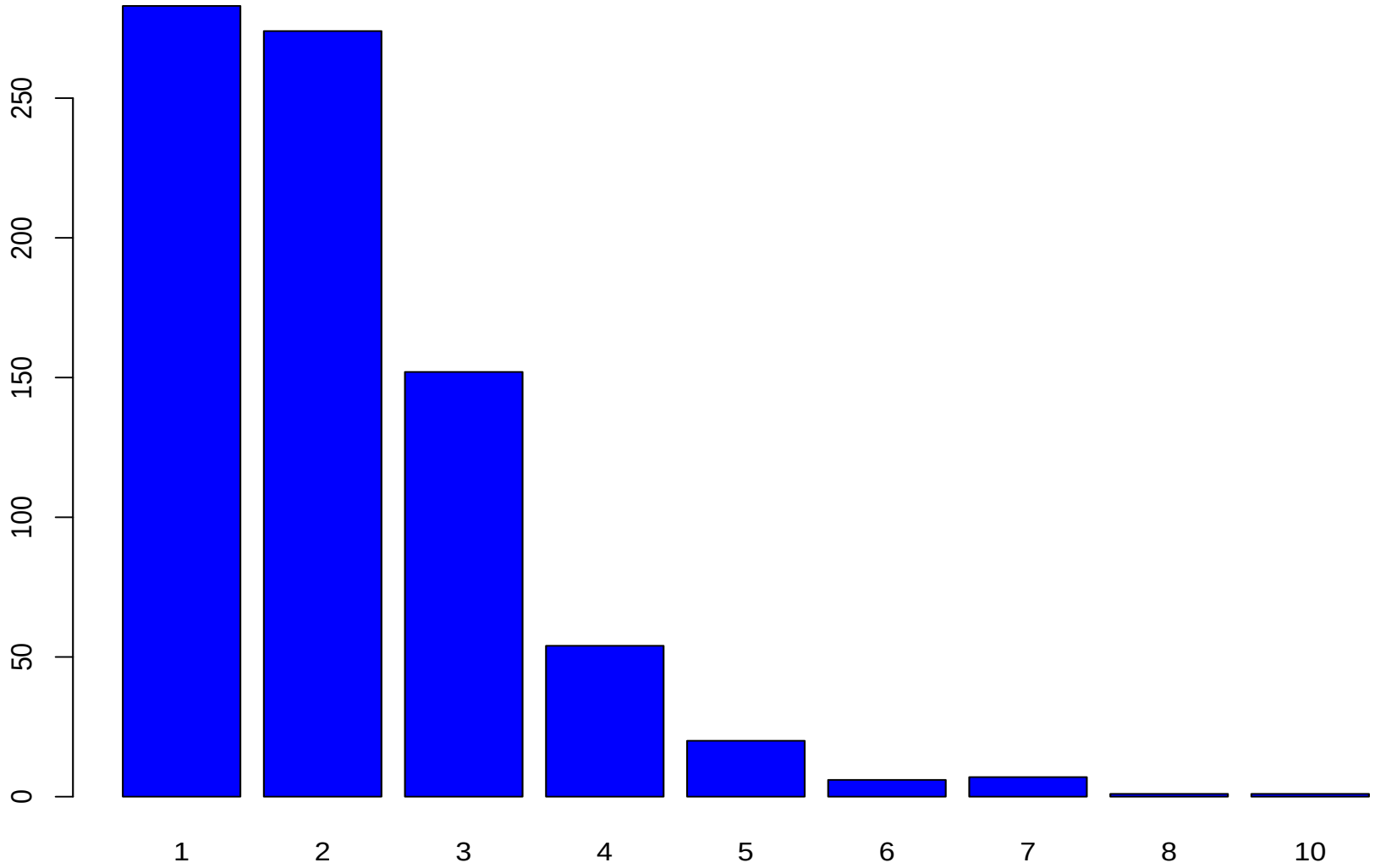
Top 10 bug hunters of Debian Med packages



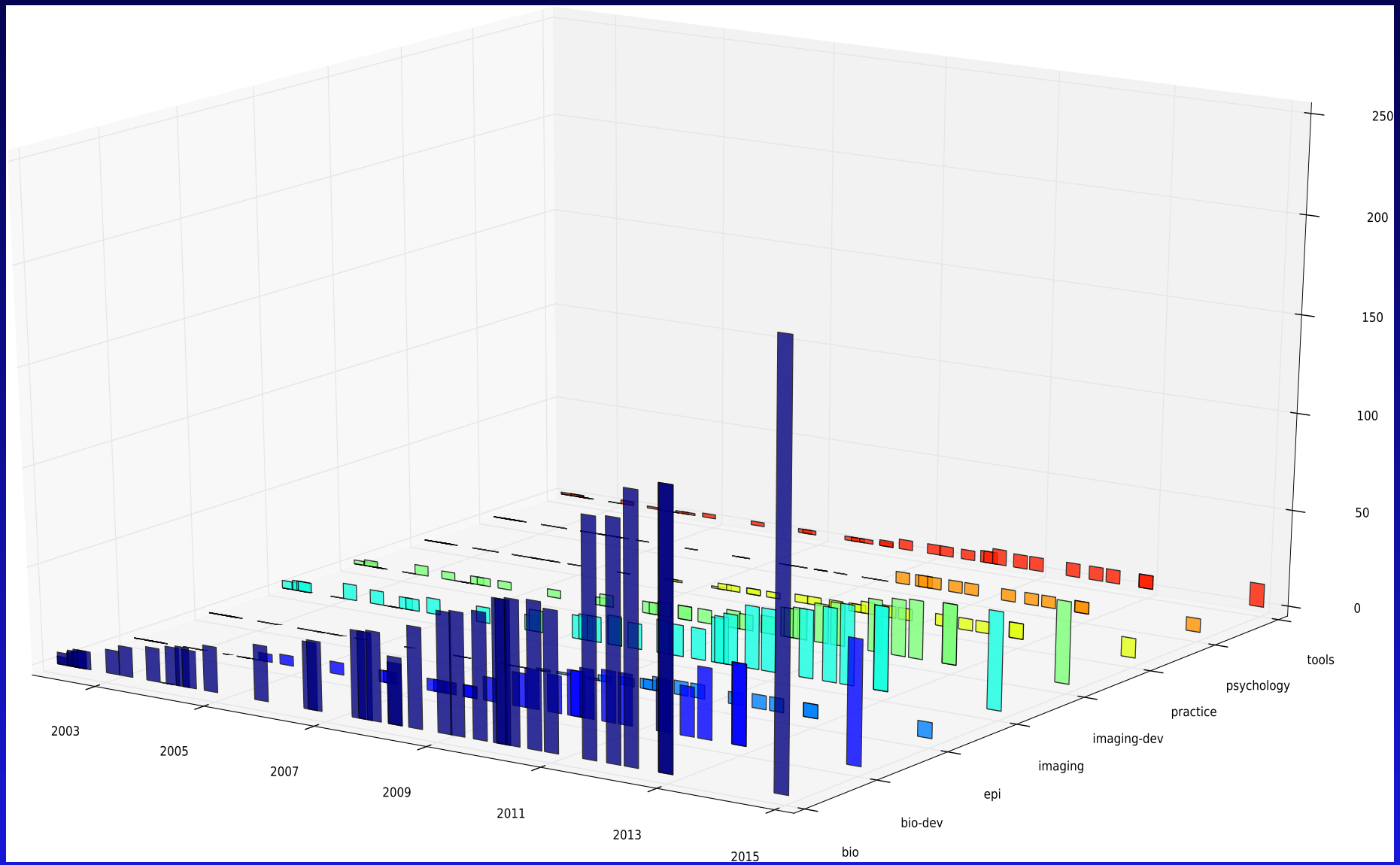
Top 10 commiters to Debian Med VCS



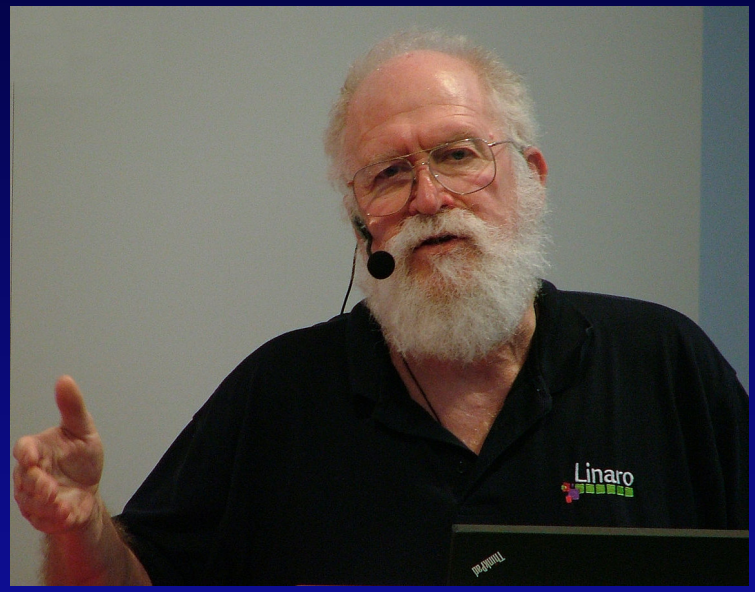
Maintainer per package relation in Debian Med



Number of Packages in selected tasks



Lesson from DebConf 15 Heidelberg

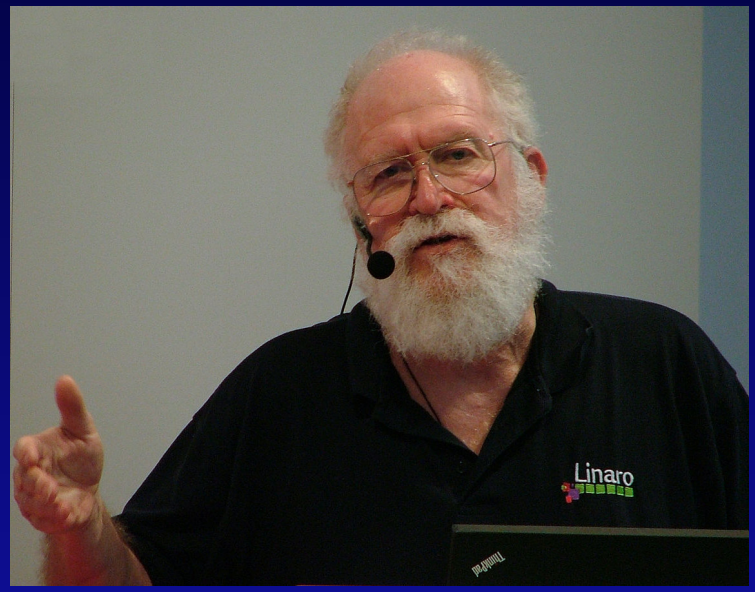


I After 15 years people are starting to use what I'm working on.

John Only 15 years? You young guys should be more patient. I should hurry up since after 15 years I might be dead but you have so much time.

Image by Olaf Kosinsky CC BY-SA 3.0

Lesson from DebConf 15 Heidelberg



I After 15 years people are starting to use what I'm working on.

John Only 15 years? You young guys should be more patient. I should hurry up since after 15 years I might be dead but you have so much time.

Image by Olaf Kosinsky CC BY-SA 3.0

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	yes
03.2015	Yves Martelli	dwvexplorer	
04.2015	Malihe Asemani	manila	-
05.2015	Afif Elghraoui	kmer-tools	yes
06.2015	Alba Crespi	r-cran-fastmatch	yes
07.2015	Julien Lamy	dcmtk++	yes
08.2015	Malihe Asemani	king-probe	yes
12.2015	Martin Uecker	bart	yes

Debian Med has attracted one developer per year

According to a *questionnaire* in Wiki

- Debian Med has 30 DDs+DMs (not all active any more)
- 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 30 DDs+DMs (not all active any more)
- 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 30 DDs+DMs (not all active any more)
- 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 30 DDs+DMs (not all active any more)
- 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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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

Team

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

Todo list for this sprint

- So **all** team packages can and should be toched by **you**
- There is a *todo list* for this sprint everybody without personal agenda could work on

Todo list for this sprint

- So **all** team packages can and should be toched by **you**
- There is a *todo list* for this sprint everybody without personal agenda could work on

This talk is available at

http://people.debian.org/~tille/talks/

Andreas Tille <tille@debian.org>