Creating a more inviting environment for newcomers
New experiences from MoM, SoB, Teammetrics

Andreas Tille
Debian

Heidelberg, 14. August 2015
1 Motivation

2 Teams

3 Mentoring of the Month

4 Sponsoring of Blends

5 Sprints
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass
- Judgment call:
  - Do I write the code myself or should I rather train a newcomer how to write the code
- Important: keep enthusiastic newcomers
- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass

- Judgment call:
  Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers

- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"
  - Reach critical mass
- Judgment call:
  - Do I write the code myself or should I rather train a newcomer how to write the code
- Important: keep enthusiastic newcomers
- Training newcomers ensures survival of the whole project
  - Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

⇒ Reach critical mass

- Judgment call:
  - Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers
- Training newcomers ensures survival of the whole project

⇒ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass

- Judgment call:
  Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers
- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass

- Judgment call:
  Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers

- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass

- Judgment call:
  Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers

- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Teams versus lone fighter

- More and more packages in Debian are team maintained
- In most cases this results in better maintenance
- However: small teams might "vanish"

→ Reach critical mass

- Judgment call:
  - Do I write the code myself or should I rather train a newcomer how to write the code

- Important: keep enthusiastic newcomers
- Training newcomers ensures survival of the whole project

→ Reserve part of your own work for the future of the project
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
No such "lever" like salary increase
No period of notice for leaving a free project
Not even clear whether somebody feels like belonging to a team
Wanted: "compensation" of motivation
Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- **Wanted:** "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
Volunteer versus payed work

- No such "lever" like salary increase
- No period of notice for leaving a free project
- Not even clear whether somebody feels like belonging to a team
- Wanted: "compensation" of motivation
- Attractive projects
  - attractive technique
  - attractive results
  - attractive work environment
When having talks about Debian at random conferences I was told by potential newcomers:

It is hard to find a helping hand for beginners.

While that’s partly true there are some teams where this is not the case. The strategic answer should be to form more teams dealing with user oriented topics.
Waking up in the morning and realising that somebody else has solved your problem from yesterday
Illusion? – Random example

Sat, 8 Feb 2014 00:16  A. Tille
Need some help for Java package

Sat, 8 Feb 2014 08:15  O. Sallou
I gonna have a look

Sat, 8 Feb 2014 08:42  O. Sallou
I was close to get it

Sat, 8 Feb 2014 09:11  O. Sallou
Got this problem, detected new one, need to contact authors

Sat, 8 Feb 2014 09:30  A. Tille
Thanks for your effort, I can give the authors a phone call

Sat, 8 Feb 2014 10:57  A. Tille
There is some other minor problem

Sat, 8 Feb 2014 12:44  O. Sallou
There is a simple workaround

Mon, 10 Feb 2014 11:03  O. Sallou
I asked authors and they solved remaining issue

Mon, 10 Feb 2014 13:51  A. Tille
Thanks. Some last advise needed

Mon, 10 Feb 2014 14:02  O. Sallou
Code updated. problem solved
Example for a good team: Pkg-Perl

Number of maintainers working on a package in pkg-perl team
Maintainer per package graph calculation

- teammetrics database has commits package, team, commiter
- for graphing the following commits are ignored
  1. packages with less than \( x \) commits (\( x = 5 \))
     (exclude fresh or unmaintained packages)
  2. maintainers with less than \( y \) commits (\( y = 2 \))
     (exclude random commits of non-team members)
Maintainer per package graph calculation

- teammetrics database has commits package, team, commiter
- for graphing the following commits are ignored
  1. packages with less than $x$ commits ($x = 5$) (exclude fresh or unmaintained packages)
  2. maintainers with less than $y$ commits ($y = 2$) (exclude random commits of non-team members)
Maintainer per package graph calculation

- teammetrics database has commits for package, team, commiter
- for graphing the following commits are ignored
  1. packages with less than \( x \) commits \((x = 5)\) (exclude fresh or unmaintained packages)
  2. maintainers with less than \( y \) commits \((y = 2)\) (exclude random commits of non-team members)
teammetrics database has commits package, team, committer

for graphing the following commits are ignored

1. packages with less than $x$ commits ($x = 5$) (exclude fresh or unmaintained packages)
2. maintainers with less than $y$ commits ($y = 2$) (exclude random commits of non-team members)
Team with room for enhancement: Debian Med

Number of maintainers working on a package in debian-med team

Andreas Tille (Debian)
Creating a more inviting environment for newcomers
Heidelberg, 14. August 2015
Example for one person "team": openstack

Number of maintainers working on a package in openstack team

Andreas Tille (Debian)  Creating a more inviting environment for newc   Heidelberg, 14. August 2015
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- **Write good documentation**
- **Write good documentation**
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- **Write good documentation**
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
How to make your team attractive

- Write good documentation
- Write good documentation
- Do not let potential newcomers wait longer than 24h (better 8h) for an answer
- Be patient - every newcomer might be different
- Insist on open discussion
- Write good documentation
- Point newcomers to the documentation
- Establish methods to involve newcomers
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
Three levers to move newcomers in

MoM | Offering time for education about packaging
SoB | Offering sponsor service if newcomer considers Blends
Sprint | Meet the team members in fruitful environment
Three levers to move newcomers in

MoM  Offering time for education about packaging
SoB  Offering sponsor service if newcomer considers Blends
Sprint Meet the team members in fruitful environment
Three levers to move newcomers in

- **MoM** Offering time for education about packaging
- **SoB** Offering sponsor service if newcomer considers Blends
- **Sprint** Meet the team members in fruitful environment

Andreas Tille (Debian)
Mentor dedicates a part of spare time to a newcomer
- Guide the student kindly into all secrets of Debian packaging
- Learning per relevant example
- Student is free to pick the package (mentor has a vetoing right)
Mentor dedicates a part of spare time to a newcomer

Guide the student kindly into all secrets of Debian packaging

Learning per *relevant* example

Student is free to pick the package (mentor has a vetoing right)
Mentoring of the Month (MoM)

- Mentor dedicates a part of spare time to a newcomer
- Guide the student kindly into all secrets of Debian packaging
- Learning per *relevant* example
- Student is free to pick the package (mentor has a vetoing right)
Mentoring of the Month (MoM)

- Mentor dedicates a part of spare time to a newcomer
- Guide the student kindly into all secrets of Debian packaging
- Learning per *relevant* 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.
Rules

- Responsiveness
  - Confirm understanding
  - Confirm ability to built
  - Open communication
  - Report
Rules

- Responsiveness
- **Confirm understanding**
- Confirm ability to build
- Open communication
- Report
Rules

- Responsiveness
- Confirm understanding
- Confirm ability to build
- Open communication
- Report
Rules

- Responsiveness
- Confirm understanding
- Confirm ability to build
- Open communication
- Report
Rules

- Responsiveness
- Confirm understanding
- Confirm ability to built
- Open communication
- Report
Summary of MoM formalism

- MoM is work
- Work is also fun
- Mentor trades 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*
- Work is also *fun*
  - Mentor trades 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*
- Work is also *fun*
- Mentor trades 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 \textit{work}
- Work is also \textit{fun}
- Mentor trades 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*
- Work is also *fun*
- Mentor trades 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
MoM for women

- Motivated by discussion about MiniDebConf by debian-women
- Dropped restriction of topic Debian Med
- Woman should pick a package and we find the according team
Motivated by discussion about MiniDebConf by debian-women
Dropped restriction of topic Debian Med
Woman should pick a package and we find the according team
MoM for women

- Motivated by discussion about MiniDebConf by debian-women
- Dropped restriction of topic Debian Med
- Woman should pick a package and we find the according team
<table>
<thead>
<tr>
<th>Month</th>
<th>Student</th>
<th>Package</th>
<th>Upload</th>
<th>Visible?</th>
</tr>
</thead>
<tbody>
<tr>
<td>02.2012</td>
<td>Luis Ibanez</td>
<td>fis-gtm</td>
<td>yes</td>
<td>handover</td>
</tr>
<tr>
<td>03.2012</td>
<td>Scott Christley</td>
<td>libswarm</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>06.2012</td>
<td>Eric Maeker</td>
<td>libquazip</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>11.2012</td>
<td>Tomás Di Domenico</td>
<td>python-csb</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>02.2013</td>
<td>Sukhbir Singh</td>
<td>hunspell-en-med</td>
<td>yes</td>
<td>other</td>
</tr>
<tr>
<td>12.2013</td>
<td>Lennart C. Karssen</td>
<td>probabel</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>01.2014</td>
<td>Jorge S. Soares</td>
<td>snp-sites</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>03.2014</td>
<td>Stephen Smith</td>
<td>phyutility</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>06.2014</td>
<td>Ian Wallace</td>
<td>openemr</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>09.2014</td>
<td>Corentin Desfarges</td>
<td>camp</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>02.2015</td>
<td>Paul Novotny</td>
<td>opensurgsim</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>03.2015</td>
<td>Yves Martelli</td>
<td>dwvexplorer</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>04.2015</td>
<td>Malihe Asemani</td>
<td>manila</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>05.2015</td>
<td>Afif Elghraoui</td>
<td>kmer-tools</td>
<td>yes</td>
<td>active</td>
</tr>
<tr>
<td>06.2015</td>
<td>Alba Crespi</td>
<td>r-cran-fastmatch</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>07.2015</td>
<td>Julien Lamy</td>
<td>dcmtk++</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
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
Sponsoring of Blends (SoB)

- Blends concept remains widely unknown amongst newcomers (but also amongst DDs)
- Newcomers might desperately seek for sponsors and simply do not know how to find one
- Kill two birds with one stone: Get the package sponsored after proving that you understood the Blends techniques
Sponsoring of Blends (SoB)

- Blends concept remains widely unknown amongst newcomers (but also amongst DDs)
- Newcomers might desperately seek for sponsors and simply do not know how to find one
- Kill two birds with one stone: Get the package sponsored after proving that you understood the Blends techniques
Sponsoring of Blends (SoB)

- Blends concept remains widely unknown amongst newcomers (but also amongst DDs)
- Newcomers might desperately seek for sponsors and simply do not know how to find one
- Kill two birds with one stone: Get the package sponsored after proving that you understood the Blends techniques
SoB rules

1. You confirm to have understood the Blends principle (have read the Blends documentation, are member of some team on alioth, reading the relevant mailing list)

2. Your package is maintained in a VCS used by the Blends team

3. Your package is listed on the Blends tasks pages (which is either because the package is inside Debian or in any Blends VCS)

4. You are unable to find a sponsor on the specific Blends list even after posting there at least twice
SoB rules

1. You confirm to have understood the Blends principle (have read the Blends documentation, are member of some team on alioth, reading the relevant mailing list)

2. Your package is maintained in a VCS used by the Blends team

3. Your package is listed on the Blends tasks pages (which is either because the package is inside Debian or in any Blends VCS)

4. You are unable to find a sponsor on the specific Blends list even after posting there at least twice
SoB rules

1. You confirm to have understood the Blends principle (have read the Blends documentation, are member of some team on alioth, reading the relevant mailing list)

2. Your package is maintained in a VCS used by the Blends team

3. Your package is listed on the Blends tasks pages (which is either because the package is inside Debian or in any Blends VCS)

4. You are unable to find a sponsor on the specific Blends list even after posting there at least twice
SoB rules

1. You confirm to have understood the Blends principle (have read the Blends documentation, are member of some team on alioth, reading the relevant mailing list)

2. Your package is maintained in a VCS used by the Blends team

3. Your package is listed on the Blends tasks pages (which is either because the package is inside Debian or in any Blends VCS)

4. You are unable to find a sponsor on the specific Blends list even after posting there at least twice
SoB results

- SoB was heavily used by Debian GIS team - now sponsees are DDs/DMs themselves
- Watching Wiki page and take mostly immediate action
- Lurking on relevant mailing lists as well on debian-mentors to catch maintainers in need of a sponsor
- If you notice a candidate for SoB please redirect them to me
SoB results

- SoB was heavily used by Debian GIS team - now sponsees are DDs/DMs themselves
- Watching Wiki page and take mostly immediate action
- Lurking on relevant mailing lists as well on debian-mentors to catch maintainers in need of a sponsor
- If you notice a candidate for SoB please redirect them to me
SoB results

- SoB was heavily used by Debian GIS team - now sponsees are DDs/DMs themselves
- Watching Wiki page and take mostly immediate action
- Lurking on relevant mailing lists as well on debian-mentors to catch maintainers in need of a sponsor
- If you notice a candidate for SoB please redirect them to me
SoB results

- SoB was heavily used by Debian GIS team - now sponsees are DDs/DMs themselves
- Watching Wiki page and take mostly immediate action
- Lurking on relevant mailing lists as well on debian-mentors to catch maintainers in need of a sponsor
- If you notice a candidate for SoB please redirect them to me
Discussion in Debian GIS after SoB started

Andreas Tille (Debian)  Creating a more inviting environment for newc  Heidelberg, 14. August 2015  25 / 37
Commits of Debian GIS after SoB started

Andreas Tille (Debian)  
Creating a more inviting environment for new contributors  
Heidelberg, 14. August 2015
Face to face meetings with users and upstream developers

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

Thanks for supporting sprints which we have proven to be successful.
Top 10 Uploaders of Debian Med team

- Andreas T (1701)
- Charles P (528)
- Steffen M (342)
- Mathieu M (320)
- Olivier S (220)
- Steve M. R (137)
- Thorsten A (109)
- Laszlo K (98)
- Aaron M. U (96)
- Alexandre M (54)
Top 10 discussion on Debian Med user mailing list

- Andreas T 4986
- Charles P 1351
- Karsten H 638
- Mathieu M 446
- Steffen M 441
- Olivier S 415
- David P 315
- Eric M 306
- Yaroslav H 266
- Sebastian H 210
Top 10 discussion Debian Med developer mailing list
Top 10 bug hunters of Debian Med packages

Andreas T 717
Mathieu M 218
Charles P 200
Olivier S 119
Steve M. R 82
Aaron M. U 44
Steffen M 35
Laszlo K 32
Dominique B 29
Thorsten A 21

creating a more inviting environment for newcomers
Number of Packages in selected tasks

Andreas Tille (Debian)  Creating a more inviting environment for newcomers  Heidelberg, 14. August 2015  33 / 37
Summary

- Care nicely for newcomers
- Trade your time to involve newcomers
- Rank newcomer issues higher than your own coding
- Do in-person meetings
Summary

- Care nicely for newcomers
- Trade your time to involve newcomers
- Rank newcomer issues higher than your own coding
- Do in-person meetings
Summary

- Care nicely for newcomers
- Trade your time to involve newcomers
- Rank newcomer issues higher than your own coding
- Do in-person meetings
Care nicely for newcomers
Trade your time to involve newcomers
Rank newcomer issues higher than your own coding
Do in-person meetings
**Teammetrics graphs**

**Mentoring of the Month**

**Sponsoring of Blends**

**Debian Med policy**
Links

- **Teammetrics graphs**
- **Mentoring of the Month**
- **Sponsoring of Blends**
- **Debian Med policy**
Links

- Teammetrics graphs
- Mentoring of the Month
- Sponsoring of Blends
- Debian Med policy
Links

- Teammetrics graphs
- Mentoring of the Month
- Sponsoring of Blends
- Debian Med policy
This talk is available at

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

Andreas Tille <tille@debian.org>