Debian Teams Activity Metrics

Andreas Tille & Sukhbir Singh

DebConf 11

Banja Luka, July 29, 2011
Motivation

- Who is inside the team?
- Who left the team?
- Does a team have enough members?
- Is the team growing or shrinking?
Quite simple start: Mailing list activity
Enhancing the observation needed

- We do not only want to know who is “chatting”
- There should be a “fair” evaluation
- More flexibility
- Technical enhancements
Additional channels / means to consider

• VCS commits
• Uploaded packages (from the UDD)
• Other ideas (IRC, . . . ?)
Implementation

- We reinvented the wheel and wrote everything from scratch.
- ... which makes it very easy for us to add new metrics or change existing ones.
Implementation

- We reinvented the wheel and wrote everything from scratch
- ... which makes it very easy for us to add new metrics or change existing ones
- The process is optimized because we perform remote operations for fetching repositories by SSHing into Alioth and performing them locally
Measure of Communication Activity

- Every project has a mailing list
- We measure who are the most active contributors
- Quantity is not the only metric because quality matters
- We handle spam
Verifying new code with ‘soc-coordination’ list

Andreas Tille & Sukhbir Singh (DebConf 11) Debian Teams Activity Metrics Banja Luka, July 29, 2011 8 / 14
Metrics for Measuring Communication Activity

- Frequency of posting
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
  - the length of the body *excluding*
    - blank lines
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
  - the length of the body *excluding*
    - blank lines
    - blank lines and quotes
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
  - the length of the body *excluding*
    - blank lines
    - blank lines and quotes
    - blank lines, quotes and signatures
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
  - the length of the body *excluding*
    - blank lines
    - blank lines and quotes
    - blank lines, quotes and signatures
Metrics for Measuring Communication Activity

- Frequency of posting
- Message body metrics
  - the raw length of the message body
  - the length of the body *excluding*
    - blank lines
    - blank lines and quotes
    - blank lines, quotes and signatures
- Anything more that we can add?
Metrics for VCS

- Frequency of commits for Git and SVN repositories

But again, quantity ≠ quality

So we also measure the number of lines:

(+): added
(-): deleted

Andreas Tille & Sukhbir Singh (DebConf 11)

Debian Teams Activity Metrics

Banja Luka, July 29, 2011
Metrics for VCS

- Frequency of commits for Git and SVN repositories
- But again, quantity != quality
Metrics for VCS

- Frequency of commits for Git and SVN repositories
- But again, quantity \(\neq\) quality
- So we also measure the number of lines:
  - (+) added
  - (-) deleted
Noooooo!

“I find your using ‘lines of code committed’ as a metric disturbing.”
Noooooo!

“I find your using ‘lines of code committed’ as a metric disturbing.”

Yes, Lord Vader, but then we don’t have many other metrics, so...
Challenges

- We don’t have the mbox archives for lists.debian.org
Challenges

- We don’t have the mbox archives for lists.debian.org
  - So we fetch the archives through Gmane via NNTP, create mbox archives and then parse them
Challenges

- We don’t have the mbox archives for lists.debian.org
  - So we fetch the archives through Gmane via NNTP, create mbox archives and then parse them
- How do we separate upstream commits for Git repositories on Alioth?
Challenges

- We don’t have the mbox archives for lists.debian.org
  - So we fetch the archives through Gmane via NNTP, create mbox archives and then parse them
- How do we separate upstream commits for Git repositories on Alioth?
- We need more metrics for measuring performance
Challenges

- We don’t have the mbox archives for lists.debian.org
  - So we fetch the archives through Gmane via NNTP, create mbox archives and then parse them
- How do we separate upstream commits for Git repositories on Alioth?
- We need more metrics for measuring performance

Please share your thoughts!
Status

- Complete
  - Mailing list (Alioth, lists.debian.org)
  - Repository (Git, SVN)

- Incomplete
  - Package upload data from UDD
  - Metrics you suggest...
  - Presenting these statistics
Questions? Suggestions?

The more we discuss, the better we can get!

http://teammetrics.alioth.debian.org/