Growing up new PostgreSQL developers

Anastasia Lubennikova
Aleksander Alekseev

In this talk

- A few words about us;
- First-person view on developing PostgreSQL;
- Where to find new developers;
- Teaching students in Moscow State University;
- Teaching students in Higher School of Economics;
- Project management gotchas;
- Onboarding & team leading gotchas;
- On distributed teams;
- And stuff like this :)

Disclaimer!

- This talk is completely non-technical. Still we hope you will be interested.
- In this talk we use ‘mentor’, ‘project manager’ and ‘team leader’ as synonyms. We know this is not very accurate but we have our reasons which hopefully will become clear later.

A few words about us: Anastasia

- Graduated with BC in Applied Mathematics and Informatics in 2014;
- Took part in GSoC with PostgreSQL as a student (2014) and a mentor (2017);
- Since 2015 work in Postgres Professional;
- Contribute mostly to index enhancements and pg_probackup;

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ · МИФИ ·
НИЯУ МИФИ

Google SUMMER of CODE
TENTH 2014 YEAR

CYBERTEC
The PostgreSQL Database Company

Postgres PROFESSIONAL

A few words about us: Aleksander

- I live in Moscow, Russia;
- Develop software since 2007;
- Contribute to PostgreSQL since 2015;
- Work in Postgres Professional company;
- Author of zson and pg_protobuf extensions;
- Interests: OSS, functional programming, electronics, distributed systems, blogging, podcasting;
- https://eax.me/ & https://devzen.ru/ ;

On developing PostgreSQL: Anastasia

- Developing PostgreSQL core is my first job.
  So I had no idea that it is considered to be a difficult thing;
- I learned mostly through practical experience;
- Code comments and mailing lists are a great source of information;
  - You see not only the code but also a reasoning behind it;
- The pace can be really frustrating sometimes;
  And results as well. It is not just development. More like R&D;
- The project is really big and it covers many areas of CS.

On developing PostgreSQL: Aleksander

- ~8 years of experience in different area (backends / distributed systems / …) before this job;
- The process is slow (several patches a week vs several patches a year);
- Something is not quite right with abstraction levels (e.g. dynahash and locks, re-using atomics outside of backend, ...);
- Something is not quite right with reusing existing code (home-brewed concurrency primitives, containers, etc);
- Perl 5.8, Autotools :(
- It doesn't feel like there is a lack of *developers.*

Where to find new developers

- Find an experienced RDBS developer who is looking for a job;

Where to find new developers

- Find an experienced RDBS developer *who is looking for a job;*
    - **Oops, there are none!**

Where to find new developers

- Find an experienced RDBS developer *who is looking for a job;*
    - **Oops, there are none!**
- OK, find a regular developer then, preferably with an experience (or at least an interest) in low-level programming;

Where to find new developers

- Find an experienced RDBS developer *who is looking for a job;*
  - **Oops, there are none!**
- OK, find a regular developer then, preferably with an experience (or at least an interest) in low-level programming;
  - **It might work. However many will leave after a year or so.**

Where to find new developers

- Find an experienced RDBS developer *who is looking for a job;*
    - **Oops, there are none!**
- OK, find a regular developer then, preferably with an experience (or at least an interest) in low-level programming;
    - **It might work. However many will leave after a year or so.**
- What about hiring graduates?

Where to find new developers

- Find an experienced RDBS developer *who is looking for a job;*
    - **Oops, there are none!**
- OK, find a regular developer then, preferably with an experience (or at least an interest) in low-level programming;
    - **It might work. However many will leave after a year or so.**
- What about hiring graduates?
    - **It will work**. However:
        - Right graduates are rare;
        - We need corresponding lectures;
        - We need mentors / team leaders;

On teaching students. Theory

- Some great books and courses already exist;
- Clear reasoning behind each component and algorithm;
- A lot of design ideas and trade-offs to discuss;
- A lot of fresh papers to study.

On teaching students. Practice

- Abstract practical tasks are boring :(
- There are not many junior tasks in DBMS development.
- It takes more than just good theory knowledge and code skills
  - Initial design, attention to corner cases,
    code comments, test coverage, thoughtful benchmarking.
- PostgreSQL specific;
  - Building, macroses, development process, etc;
  - TODO-list on wiki doesn’t help much.

On teaching students in MSU

- It was an *elective* course for all faculty, for students with various background;
  - **Elective - turned out it was a bad option :(**
- Students tend to choose simpler courses to get their grades;
- Chosen course was not fixed in the beginning of the semester;
- Many students visited lectures out of interest, but didn’t do homework and didn’t take exams;

On teaching students in HSE

- *Mandatory* course for students of distributed systems department, strong CS background;
  - **Mandatory - is good :)**
- Cumulative grade system, modules instead of semesters
  - One module = ½ semester, semester = half a year;
- Surprisingly there was no system programming (C, *nix) course before our course;

On teaching students. Tasks and homework

- ‘Papers we love’ style seminars;
    - This was a good move.
- Work in small groups;
    - Bad idea. See the slide about MSU and leaving students.
    - There should be a separate course on “mature” software development;
- Patches and contribs for PostgreSQL;
    - Hard to find suitable tasks;
    - Requires a lot of mentorship (time, code review, etc);
    - Turned out to be quite challenging for some of our students;
- Patches for educational DBMS HurmaDB; (*)
    - A lot of tasks;
    - Less enthusiasm for a “toy” project comparing to PostgreSQL;
    - Little benefit from the company’s point of view;

(*) Hurma (хурма) means ‘persimmon’ in Russian. Second meaning: something weird.

We got students with strong* DBMS background.

What’s next?

What’s next?

- Providing diploma projects;
    - Requires a dedicated mentor!
        - Not everyone can be a good mentor, see next slides;
    - Worked well for some of our colleagues;
- Internship;
- Job offer, trial period;
- Gotcha: you raised a valuable expert.

  Such experts are prone to get good job offers from other companies.

Onboarding gotchas

- Write articles and record videos for new employees
  - How to build PostgreSQL
  - On PostgreSQL internals
  - etc
- Don’t allow new employees to work too hard
  - Watch out impostor syndrome!
  - Some employees might think that they are not working good enough and start to work on holidays
  - Don’t allow this! There is little use of exhausted employees. Explain it.
- Experienced new employees also need onboarding

It would be quite challenging for a regular PM to manage DBMS development.
(We’ve tried this approach :)

Thus PMs most likely will grow up from your developers.

On transitional process

- The transition from developer to PM might be depressing;

On transitional process

- The transition from developer to PM might be depressing;
- You spent years to develop technical skills;

On transitional process

- The transition from developer to PM might be depressing;
- You spent years to develop technical skills;
- And now suddenly you need soft skills;

On transitional process

- The transition from developer to PM might be depressing;
- You spent years to develop technical skills;
- And now suddenly you need soft skills;
- Which you are probably not very good at since you didn’t develop these skills, at least intentionally;

On transitional process

- The transition from developer to PM might be depressing;
- You spent years to develop technical skills;
- And now suddenly you need soft skills;
- Which you are probably not very good at since you didn’t develop these skills, at least intentionally;
- Personally I consider this a new stage in my career and also an opportunity to evolve further;

On transitional process

- The transition from developer to PM might be depressing;
- You spent years to develop technical skills;
- And now suddenly you need soft skills;
- Which you are probably not very good at since you didn’t develop these skills, at least intentionally;
- Personally I consider this a new stage in my career and also an opportunity to evolve further;
- You start to look at many things from a different perspective;
    - On development, business needs and people

Change of perspective

- What is the priority of this feature? Did our clients even ask for it?

Change of perspective

- What is the priority of this feature? Did our clients even ask for it?
- Does it worth an effort to optimize this feature? Did our clients complain on it performance? Do our clients even use this feature? How did you look for a bottleneck?

Change of perspective

- What is the priority of this feature? Did our clients even ask for it?
- Does it worth an effort to optimize this feature? Did our clients complain on it performance? Do our clients even use this feature? How did you look for a bottleneck?
- Perfect is the enemy of good;

Change of perspective

- What is the priority of this feature? Did our clients even ask for it?
- Does it worth an effort to optimize this feature? Did our clients complain on it performance? Do our clients even use this feature? How did you look for a bottleneck?
- Perfect is the enemy of good;

etc… You will be surprised how often great technical experts don’t think on such things.

Project management gotchas

- Your employer might think that she doesn’t need PMs;
  - PostgreSQL community style;
  - However you do need PMs to have a predictable and sustainable development process (with right estimates, priorities, etc)
  - Processes are very important!

Project management gotchas

- Your employer might think that she doesn’t need PMs;
    - PostgreSQL community style;
    - However you do need PMs to have a predictable and sustainable development process (with right estimates, priorities, etc)
    - Processes are very important!
- Hiring and firing people is part of the job;

Project management gotchas

- Your employer might think that she doesn’t need PMs;
  - PostgreSQL community style;
  - However you do need PMs to have a predictable and sustainable development process (with right estimates, priorities, etc)
  - Processes are very important!
- Hiring and firing people is part of the job;
- Delegate!

Project management gotchas

- Your employer might think that she doesn’t need PMs;
  - PostgreSQL community style;
  - However you do need PMs to have a predictable and sustainable development process (with right estimates, priorities, etc)
  - Processes are very important!
- Hiring and firing people is part of the job;
- Delegate!
- Bus factor;

Project management gotchas

- Your employer might think that she doesn’t need PMs;
    - PostgreSQL community style;
    - However you do need PMs to have a predictable and sustainable development process (with right estimates, priorities, etc)
    - Processes are very important!
- Hiring and firing people is part of the job;
- Delegate!
- Bus factor;
- Some colleagues might have bad experience with SCRAM before. Don’t make rush changes.Try to take only best from agile methodologies and adjust them to your team.

On distributed teams

- Text is prone to be received as an aggressive message;

On distributed teams

- Text is prone to be received as an aggressive message;
- Thus be brief and use voice whenever possible;

On distributed teams

- Text is prone to be received as an aggressive message;
- Thus be brief and use voice whenever possible;
- For stand-up meetings we choose Mumble;

On distributed teams

- Text is prone to be received as an aggressive message;
- Thus be brief and use voice whenever possible;
- For stand-up meetings we choose Mumble;
- Check the status of remote employees tasks from time to time (in the middle of the SCRAM sprint)
  - Just ask “what’s up” :)

Team leading gotchas

- You work with both people who are completely new to the area and with extremely experienced rockstars.
- Because of the project complexity, you won't be able to keep everything in your head. You need to delegate and trust people a lot.
- Quite a few people (if any) can be experts in all the components.
  And vice-versa, almost every developer will be good at some area [1].
  Get ready to manage a lot of discussions.
- If there is no consensus you have to make a decision on your own.
  Be prepared to upset someone.

[1]: see also https://ferd.ca/the-hitchhiker-s-guide-to-the-unexpected.html

Three roles of a team leader

The roles are:

- Psychologist (people)
- Logistic (processes)
- Expert (development)

No one is good in all 3 roles simultaneously, thus:

- Gather a team specifically for a given team leader or
- Use two persons as a single team leader

https://felixit.blog/2018/03/31/timlid-v-trioh-licah/ -- the article is in Russian but Google Translate should manage

Three roles of a team leader

The roles are:

- Psychologist (people)
- Logistic (processes)
- Expert (development)

Classical PM

No one is good in all 3 roles simultaneously, thus:

- Gather a team specifically for a given team leader or
- Use two persons as a single team leader

https://felixit.blog/2018/03/31/timlid-v-trioh-licah/ -- the article is in Russian but Google Translate should manage

Three roles of a team leader

The roles are:

- Psychologist (people)
- Logistic (processes)
- Expert (development)

Team leader

No one is good in all 3 roles simultaneously, thus:

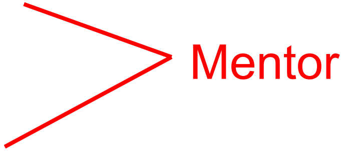
- Gather a team specifically for a given team leader or
- Use two persons as a single team leader

https://felixit.blog/2018/03/31/timlid-v-trioh-licah/ -- the article is in Russian but Google Translate should manage

Three roles of a team leader

The roles are:

- Psychologist (people)
- Logistic (processes)
- Expert (development)



Mentor

No one is good in all 3 roles simultaneously, thus:

- Gather a team specifically for a given team leader or
- Use two persons as a single team leader

https://felixit.blog/2018/03/31/timlid-v-trioh-licah/ -- the article is in Russian but Google Translate should manage

Links

- https://postgrespro.com/
- https://github.com/postgrespro/
- https://www.msu.ru/en/
- https://www.hse.ru/en/
- https://wiki.mumble.info/wiki/Main_Page
- https://github.com/afiskon/hurmadb
- http://postgres-edu.blogspot.ru/search/label/Hacking%20PostgreSQL

The time of this talk is limited, thus we couldn’t tell everything we wanted. Fortunately we are available after the talk.

Thank you for your attention!

- a.lubennikova@postgrespro.ru
- a.alekseev@postgrespro.ru

Bonus slides!

What does SENIOR mean?

Software developer’s evolutionary path:

1. You get tasks and write code. Tasks on input, patches on output.
2. You start to think of bigger **technical** problems, split them into the tasks, plan the tasks, consider dependencies, etc.
3. You start to look for **business** problems and try to solve them.

https://www.youtube.com/watch?v=cvHK5tRUCrs -- the video was removed :(

Some of recommended white papers (1 / 2)

- http://www.vldb.org/pvldb/vol10/p385-yu.pdf
- https://www.allthingsdistributed.com/files/p1041-verbitski.pdf
- https://people.inf.ethz.ch/zistvan/doc/vldb17-caribou.pdf
- http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D90D2CAD79DE47115B96EFD947A6427B?doi=10.1.1.395.6493&rep=rep1&type=pdf
- https://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf
- https://raft.github.io/raft.pdf
- https://arxiv.org/pdf/1802.07000.pdf
- https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/36726.pdf

Some of recommended white papers (2 / 2)

- https://static.googleusercontent.com/media/research.google.com/en//archive/spanner-osdi2012.pdf
- https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/46103.pdf
- http://db.cs.cmu.edu/papers/2016/pavlo-newsql-sigmodrec2016.pdf
- https://arxiv.org/pdf/cs/0701157.pdf
- See also:
    - http://www.redbook.io/
    - https://blog.acolyer.org/
    - https://t.me/databasss