


Chapter 2. I'm only 11 and I already did this...


I'm only 11 and I already did this implementation of function nobody asked for.

Having a lot of people in team will require good communication skills or rules. In my opinion, team lead should focus only on PRs and architecture, shifting focus more to the architecture when it's needed. And team members should focus on solving tasks as close as team lead see them to be implemented.



I had this problem. As team lead I was developing as well, but what is worse I didn't share my development vision with others. Because of this, some of my team members could stay with no work to do, while I had to pay more attention for checking PRs of others, they stayed in review for too long. Consequences came immediately, some team members found what to do instead of KATH project and lost motivation, while others were more on their own with their own ideas and visions of task implementation. It led to lost of 1-2 people and some strange things in code: different comments notation, shitty implementation of some tasks, unused functions...

On the other hand, my other team members were so motivated that were doing more than I was asking, leaving others with no task as well. And here when communication skills should come in.

 If you are developer, make sure to tell about problems, ideas, wishes you have to your team lead. He will help to solve your problem, will tell if your idea good enough to implement it now or later. You are hands and team lead is head. When head doesn't know what hands do, it causes troubles.

 If you are team lead, make sure to clearly define tasks before giving them. Plan it, make a diagram, define expected input and output. Don't hesitate to ask opinion from your members, ask them for a progress, ask them for a problems. When head doesn't know what hands do, it causes troubles.

I became a team lead for the first time, and at that time nobody knew what to do with Harvard, nobody could imagine what sort of problems we will have and what architecture it will require. It was more like a research for us. All of us had different stack of technologies, and people were bringing different habits from other programming languages which caused even more problems.

 I don't want to be excused. It's a team lead's task to think about such things. So, I will share some guidelines here  Oh, yeah. I'm fucking good KATH member . I hope it will help to keep project consistent.