



PhUSE US Connect 2019

Mindful Programming

Kala Shivalingaiah, ICON/DOCS, Durham, USA

ABSTRACT

As the industry shifts towards global programming, companies are forced to expand to meet demand. Teams are distributed all across the world, working across different time zone cooperate for a common goal, and bringing together their strengths and lifting each other's weaknesses.

As growth increases, new challenges such as communication, culture, and regional holidays arise which influences the timeline and quality of the project. There many unanswered questions such as, how do we manage the timeline without compromising the quality? What do these programmers need? In what capacity can managers/mentors fill the gap? Is being good, technically enough within the current model of growth? What else do you need, to be a successful programmer? Finally, how can you mindfully program while keeping the balance between timeline and quality?

In this paper I have incorporate seven highly effective habits, as described by author Stephen Covey, into our day-to-day programming life. We have successfully implemented this process in our current project, which helped up achieve the required results. Clients are immensely satisfied and are asking that this program is extending in to different studies and groups.

INTRODUCTION

The statistical programming team plays an important and critical role in the analysis, reporting, and submission of data. Success in this team is defined by quality of work and a timely deliverables to the cross function team. This can only happen if there is open and transparent communication within a team. The main question that should be focused on is whether communication is a "soft skill" to mentor and coach, and if so how do we do this? I have adopted the teachings of Harvard graduate, Dr. Steven Covey into my role, which have provided immense success. By organizing and showing the seven steps, I will provide examples on how to be more effective.

1. Be proactive

This is important trait for any programmer. In general, programmers don't want to talk and express any concerns. Meetings are mostly one-sided with little to no participation from programmers and little or no timely feedback on their team. These factors are a recipe for disaster and can bring up crucial problems too late into the timeline.

Solutions

- Kick-off meeting – Ask questions about the study, get to know your team and bring in lessons learned from your previous experiences.
- During the study-programming phase – DO NOT assume anything unless you are one hundred percent sure. Ask questions and respectfully bring any suggestions that you believe can improvement the process. Communicate with your team actively; talk about your assignment progress, any roadblocks, new lessons, questions, or concerns that you might have. Feedback is always good since it can either motivate the team or give an opportunity for another person to learn a new skill.
- Post study – Take moment to see what went well and what else could have been achieved. Do you see a process improvement? Did you learn a new skill? Do you see an opportunity to give training to someone else? Did you make connection with a SME? By building off of this project, you can learn new things to implement on the next project you work.

In summary, being proactive as a programmer means we are responsible for our own timelines and quality. Move away from reactive approach and proactively organize, prioritize, communicate, and discuss to improve efficiency. Be active in meetings; voice your thoughts and suggestions to improve the cohesiveness of the team. Give and receive active feedback in timely manner so that mistakes that might have happened won't be repeated again in a future trial. The most successful programmers are those who are proactive, they don't succumb to boredom in their



career, they actively choose a direction and move forwards.

2. Begin with the end in mind

Dived and conquer has been the approach we are taking in this industry. We compartmentalize our job and stay within our study or department. However there is a need to visualize big picture, and have holistic approach. Programmer needs to understand how their program is effecting this study, this therapeutic area and this industry, only then will they be able to communicated and deliver quality.

As I have insisted in past habit, communication is key. For us to effectively communicate we will need to understand the story not a short paragraph or few chapter but the overall story. Keeping end in mind will give you right priority. Many programmers have no idea on important deliverable date, stake holder on the study, other cross functional team, dependencies etc. It is important to know all of this so you can plan on effective communication pathway. And mainly this is part of being proactive. Proactively understanding the end will give you sense of ownership, meaning for you work, motivation to look at the finished product/results, know your prioritization.

3. Put first things first:

Here we are talking about both procrastination and priority. We all face this dilemma; we are caught between urgent and important. Being able to prioritize is extremely difficult goal without manger or mentor support. Able to react to urgent matter but continue to stay proactive about important matter is where quality and timeline will converge again.

We are just adding on here, Priority is very important. On any given day we are juggling between multiple projects and studies, to specie it up we are regulatory and ad-hoc requests. Without being proactive and understanding the end goal we will never be able to prioritize. Having said this procrastination is never a good idea. Will only cause stressful situation, timelines are not met, and may cause quality issues. When we don't plan and prioritize we may end up spending lot of time firefighting without achieving out long-term goals.

When unable to prioritize it is important to reach out to your Study lead. Again proactive communication is key.

4. Seek to understand, then to be understood

Listening is big part within any area. This can be simply sufficed as open communication between Manger-Programmer, Programmer-Vendor, and Programmer-Client etc. When we seek to understand we build strong relationship and in turn better business. Especially with global team it is very important to understand each other's background, culture to have effective communication. If we want to be effective programmers that lift up each other, we need to use empathic listening and genuinely understand our colleagues. This compels others to reciprocate the listening and be willing to be influenced by you. Communication is foundation to success.

5. Win/Win

Whether or not you are the only developer working on a project, you must not let your ego cause you to think only of yourself. This only give superficial contentment but damage the team productivity. Increase in team moral will build trust and ultimately have team who can keep their timeline and quality without much stress.

You should think about mutually beneficial solutions that will ultimately lead to a better long-term resolution, rather than if only one person in the situation gets their way. Cultivate the habit of asking yourself questions like: "What's in it for them that I can also benefit from? How can we both get some portions of what we want without damaging our relationship?"

6. Synergize

Especially in clinical industry there is hardly anything you can achieve without Synergy. The interdependencies with between different department and different programmer within one department is quite obvious and which evolves large teams. It is important to empower very programmer by helping them feel very comfortable expressing different points of view without the fear of embarrassment. In this way, your team can be a lot more productive by taking advantage of the strengths of every member. Teamwork is greater than the sum of its parts.



7. Sharpen the saw

Accordingly to Covey sharpening the saw means renewing in all aspects, for programmer which may include physical, mental, social/emotional and technical. We must master our fundamentals and continue to improve technically. Programmer must challenge themselves and learn more, set stretch goals.

Imagine you look at a colleague who is stuck and stressed out, because they are not able to finish a task and have not slept in the past two days. They do not have a deadline, but they are just very eager to finish their work, so they aren't prioritizing sleep. You advise them to take some rest because you feel sorry for them. "I am too busy with programming," they may respond. This may be an exaggeration, but it isn't that far from the truth. More often than not, we get so caught up in the day-in and day-out business of life that we rarely step back and ponder about our lives. We are trapped in the "busyness" so much that we do not even try to find any time to sharpen the saw.

If you want to become an effective programmer, first you have to decide to do so. As Covey said, "I am not a product of my circumstances. I am a product of my decisions." Excellence is not accidental but comes as a result of careful planning, hard work, and devoted dedication. As Aristotle said, "We are what we repeatedly do. Excellence, then, is not an act but a habit."

Start to practice any of these habits and see for yourself how it goes.

CONCLUSION

These new habits give us something to think about and work towards as we become more effective in our daily activities. These best practices, deployed habitually, have proven themselves time and again across successful studies.

REFERENCES

The 7 Habits of Highly Effective People – By Stephen Covey
Simpleprogrammer.com

RECOMMENDED READING

The 7 Habits of Highly Effective People – By Stephen Covey
Getting Things Done: The Art of Stress-Free Productivity - by David Allen

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Kala Shivalingaiah

Company: ICON DOCS

Email: chandrakala.shivalingaiah@docsglobal.com