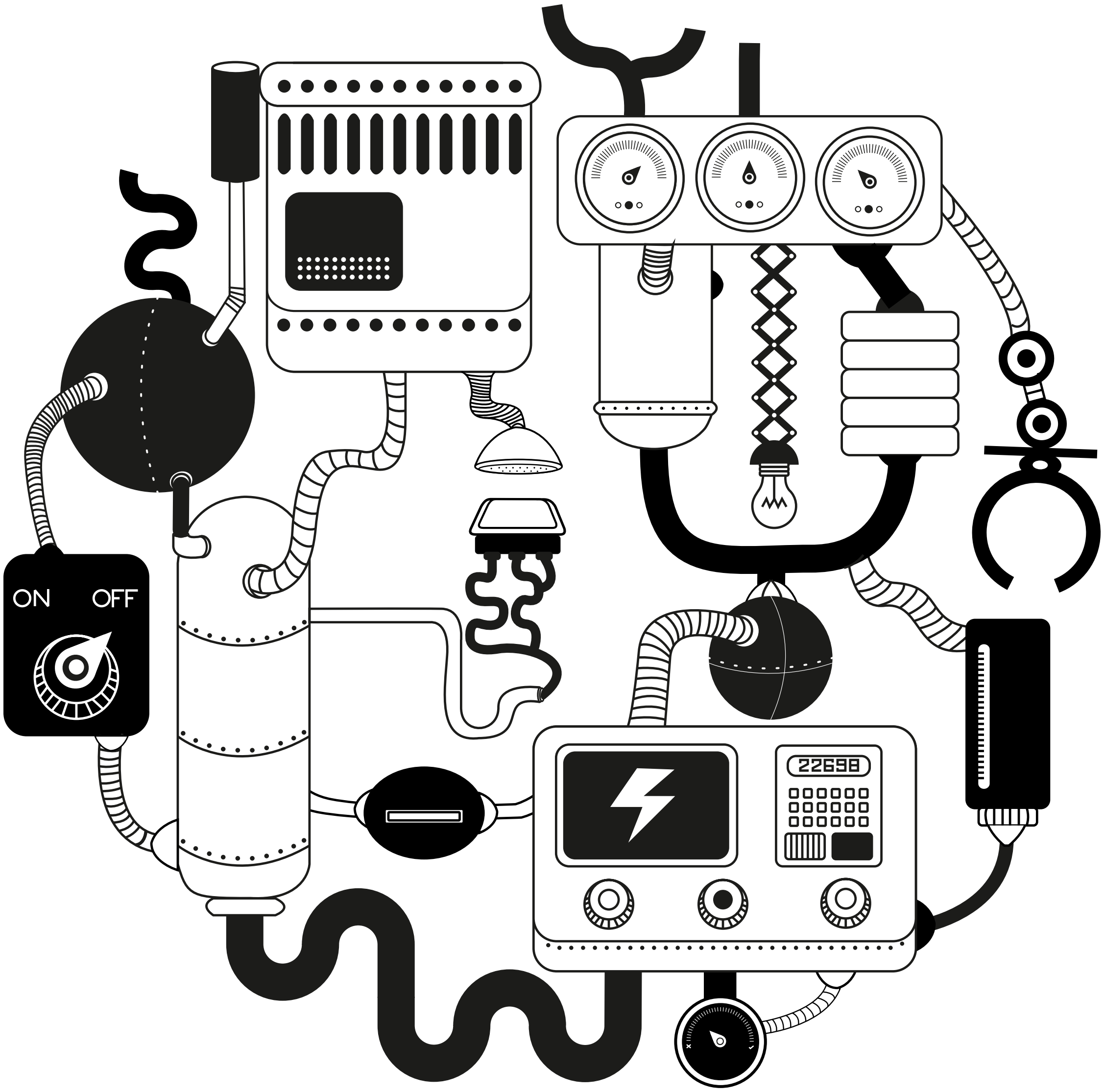
Deloitte.



Management of Programmers

Programmers are a special breed of workers.

June 2016

If you expect a developer to come into work at 8am and leave at 5pm to restart his day the next day then you might be sadly mistaken. Programmers cannot be likened to normal 8am-5pm office workers. Their minds do not switch off at the end of the day nor do they start at 8am.

They are problem solvers where many work through the nocturnal hours to achieve their goals. They are flexible in nature regarding working hours. Managers find that setting shorter core hours e.g. 9am-3pm where both business and tech are around at the same time to answer questions is more useful.

The world of software development has evolved to the point where projects comprise of virtual teams who reside across the world. This gives rise to the use of skillsets that may be limited in a country, ability to do 24 hours development (Coder's checks in his code and a coder in a different time zone can pick up this code and continue coding) or the reduction in cost of the resource.

The Organisational culture needs to embrace flexibility and the concept of virtual teams. Rather than trying to micro-manage these teams, the manager must use tools to track progress e.g. VersionOne. Effective communication tool are essential e.g. Slack, Skype, Lync. These tools are used for collaboration.

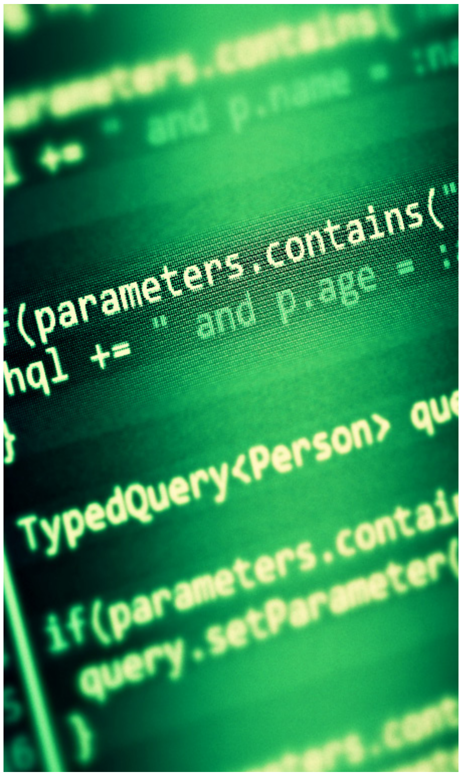
> They are problem solvers where many work through the nocturnal hours to achieve their goals.

When managing programmers in a project, it is necessary for the manager/project manager/programme manager/ delivery manager to foster an environment of collaboration. Collaboration is not just between developers. Collaboration needs to be nurtured from the beginning of the project. In successful projects, it is important to involve the programmers early in the project cycle to ask the right questions, include thought processing around integration and non-functional requirement and most importantly understand the business requirement from the beginning.

After interviewing many developers, I asked the question of when is code completed. And truthfully, it is never complete. It may work and fulfil the requirements; it will adhere to the principles, patterns and standards required; however there are still ways to optimise the code written. There are always better ways to do a piece of work. With saying this, it is very important to have deadlines else a developer will continually try to optimise and build better code.

Deadlines are essential. These should be decided collaboratively between the programmer and the manager. If a programmer is left to decide the time on his own, it may not be optimal as he may develop for as long as he deems necessary.

Requirements need to be clear. If you are expecting a 'Tazz' requirement and have not detailed this clearly in the requirements definition then a programmer might end up building a 'Ferrari' requirement which may be more costly and complex.



Programmers within themselves have different personalities that require a change in management style. One size of management is not fit for the entire programming team. Programmers in general do not like to manage a project or be involved in the detail of management. They are either:

- **Self-Managed Programmers**

 With self-managed programmers, they are able to own and deliver a piece of work without the requirement for continuously checking if they have completed a task. They are willing to take on new work and converse freely with other developers in a cross functional team.

- **Pure Coders**

Pure coders tend to code in a corner. They are not likely to work well in cross-functional teams and will shy away from taking responsibility for anything other than their own code.

- **Programmers who require micro-management**

There are two types of programmers that require micromanagement. Those that are not able to code well and those that are brilliant at coding but cannot stay within the requirement. Such micro-management can be handled as a daily stand up. When managing programmers who are junior or those individuals who are not good programmers, peer mentorship works well.

As a manager, it is wise to create a technical lead/s to review code and design. This allows for improved quality through iterative learning. It is useful for a manager to have a conceptual understanding of the technical environment. When a manager understand the complexity of the technical environment, then he is also able to engage with the programmer better.

Programmers are motivated when challenged and given the opportunity to expand their capabilities, are respected and trusted to technically design and implement the requirements and are allowed to work flexibly.

Programmers are highly demotivated when requirements continuously change with no reason, business dictates the technology design, insufficient executive support in technology, insufficient time for development and testing, insufficient requirements provided and when post implementation reviews become a finger pointing exercise.

One of the components disliked by programmers is administration. Compilation of technical documents is a weakness amongst programmers. Managers need to ensure that these documents are completed where required.

For managers to manage programmers effectively, managers must hire the right type of personality and skillset to fit the culture and ethos. I would not hire a legacy developer to do web development.
Most projects are adopting an agile approach. In doing so, developers need to be self-managed and cope with the uncertainty of the unknown. “We don’t know, what we don’t know!”.
The role of the manager is to remove the obstacles facing the programmer so that he can concentrate on delivering a requirement.

When managing programmers, it is wise to have an open plan office to foster collaboration. It is essential that silo styled environments are broken. Part of the office layout should include a ‘games corner’ where the programmer can take a break and refocus his thoughts. Ironically collaboration and team building happens over coffee, smoke-break, lunch and while playing games. As programmers are not great at administration items, managers must find alternative solutions to assist the programmer with his personal admin e.g. paying license disk.

When programmers are distracted from their work focus, it takes a while to re-engage that focus.

It is imperative that managers also ensure that developers are not continuously disturbed. A good example is when a programmer has his headphones on, means he does not like to be disturbed. Protect your programmers from business and other teams. Ensure that disturbances are kept to a minimum for the programmer. In the case where business related questions need to answered, a manager should be the first line of support before the programmers are engaged.

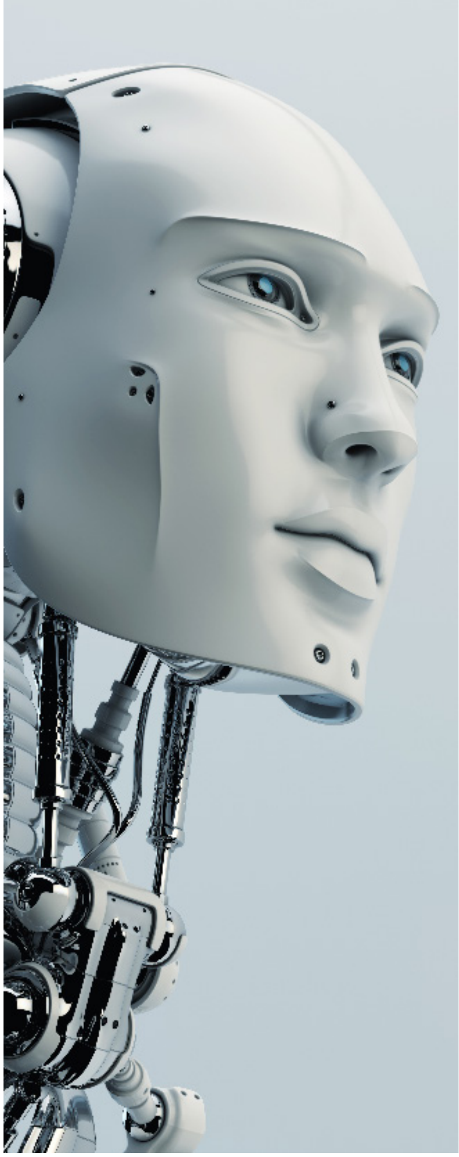
With the changes in technology, it is important for programmers to become cross-functional. Managers can guide this process by linking training and learning to KPI’s. As part of time management on projects, managers must allocate time for the programmers to investigate solutions, learn new knowledge and train others.

A good manager also understands the culture of his team whether local or remote. Impact can be specific periods of time when this person is not contactable e.g. Muslims have prayer time on Fridays over lunch.
He should have an open door policy allowing the programmer to share his problems and frustrations and where possible resolve these for him.

This document references (Biser, n.d.) and (GOTTERER, 1969).

References

Biser, S. (n.d.). developers. Retrieved April 21, 2016, from TOPTAL Web Site: https://www.toptal.com/remote/how-not-to-manage-your-remote-developers
GOTTERER, I. H. (1969). Management of computer programmers. Computer History Museum, 419-424.



Contact us

Kaajal Maharaj
Technology Strategy and Advisory Manager | Deloitte Consulting | Technology
Tel: + 27112096712
Email: kmaharaj@deloitte.co.za

Disclaimer: Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee (DTTL), its network of member firms and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. Please see www.deloitte.com/about for a more detailed description of DTTL and its member firms. Deloitte provides audit, consulting, financial advisory, risk management, tax and related services to public and private clients spanning multiple industries. With a globally connected network of member firms in more than 150 countries and territories, Deloitte brings world-class capabilities and high-quality service to clients, delivering the insights they need to address their most complex business challenges. Over 210 000 professionals of Deloitte are committed to becoming the standard of excellence. This communication contains general information only, and none of Deloitte Touche Tohmatsu Limited, its member firms or their related entities (collectively, the "Deloitte Network") is, by means of this communication, rendering professional advice or services. No entity in the Deloitte network shall be responsible for any loss whatsoever sustained by any person who relies on this communication.