Filip Falinski

A report on

**TECHNICAL DEBT**

**What is Technical Debt?**

The Term “Technical debt” originated from Ward Cunningham who first used the term when describing why recourses need to be budgeted for refactoring to stakeholders. Technical debt is very similar to financial debt in that to the person borrowing the funds it is more important to have the item now rather than save up for it themselves and purchase it at a later stage. In software Development, technical debit ( also know as “tech debt” or “code debt”) is when a company needs to deliver a product fast in order to meet the deadline. This approach is neither bad or good, it solely depends on how it is dealt with, Some companies take tech debt intentionally and strategically meaning that they are aware of the consequences and are prepared to deal with them this is referred to as “true Tech Debt”. While other companies might be pressured by deadlines and competition and take the debt in order to go the easy way rather than taking the long way with releasing poor code in order to achieve gains. Now technical debt does not necessarily mean that the code or the project will be a mess, like mentioned before if it is approached with the right plan it can be risky but beneficial to the company.

**Minimizing Technical Debt.**

Companies using agile development cannot avoid technical debt but the can minimize it so that it does not hinder them in the future. The first step to minimize tech debt is to identify the source of the debt and its cause. This needs to be acknowledged by the team and they need to adjust to it. When trying to minimize the debt it is not always possible to fix all the issues that cause the debt, meaning that the highest risks must be prioritized first. When the high risks have been identified, the second part comes into play which is to execute the risks. Once all the issues have been prioritized they need to be executed and adjusted to meet the business. Another method that can help reduce debt is to use automated testing as it can identify code defects in early stages. After the issues have been identified and solved the team needs to plan out for the future aspects of the development and how they can adjust so that similar problems don’t occur and they can work efficiently without worrying about more technical debt.

**Technical Debt Management.**

Management of technical debt is the work that builds when quicker and short term solutions are implemented instead of optimal solutions. Like mentioned before technical debt isn’t always bad it can really beneficial to a company when they want to get ahead of other companies and send their applications to the public market but this is only when it is managed correctly. It might not be the prettiest solution in terms of code and other software terms but it can help generate revenue which than will help to sort out a schedule and address and rework the issue the were present on launch. One of the major way to manage this debt is to talk about it during every meeting with the developing team, it is more than likely that they are aware of the debt and can provide their own input on it which will help manage it. This will help in the long term as it will help the manager know when to take on more debt or when to slow down on it to sort out current issues.

Being organised is very important in managing tech debt, similar to talking about debt at every meeting it is also crucial to talk about what happens if the technical debt is not addressed and the problems it will cause in the future. That is why creating a schedule in the developing cycle that has area that are dedicated to testing, bug fixes and running maintenance and dealing with the technical debt insures that the team is dealing with the faults and weaknesses while maintaining to the work schedule. Giving the task of tracking the technical debt to the development team is also a good practice, making the team list the debt to the backlog makes sure that they know that it is as important to deal with as it is to implement new features.

The last way to manage any tech debt is to plan and create reasonable sprints. Asking too much of the team is the easiest way to build up code debt, so by communicating with the team and planning out sprints that will produce efficient amount of work is a good way to avoid more debt. Sometimes it will be inevitable to build it up but if the rules listed above are followed that debt can be dealt with along the way.

**What Causes Technical Debt?**

It is almost inevitable to not encounter tech debt, simply because Technical debt occurs when the design and implementation collide with the business aspect of the project and deadlines. This means that there are many causes which can lead to tech debt, it can happen when certain development work needs to be cut short by the developing team such as writing clean code or writing proper documentation. Some of the most common issues that develop into technical debt are, ‘Business pressure’ this is when a company is under pressure from stakeholders with meeting their deadlines or meeting their contract agreements which leads the company to make decisions such as producing poor code, reduce testing and skip other aspects of the development which will lead to the company loosing productivity. Another cause is ‘Incompetence’ this is when the developer has no idea how to write proper code and proceeds to take the easy way out and rush production with poor and messy code. The list of what causes code debt is constant which is why it is inevitable to avoid it. Identifying the issue in the early stages of development is the best solution as it can be dealt with but more the project mature it becomes more difficult to manually identify and manage the technical debt.