

Startups and Technical Debt

Managing Technical Debt with Visual Thinking

Marcos Chicote

Llamando al Doctor
Buenos Aires, Argentina
chicotemarcos@gmail.com

Abstract—This position paper addresses the issue of startups and technical debt. Early stage startups condition makes creating technical debt an almost mandatory decision. Not managing technical debt can be deadly for a startup as fast product iteration cycle is necessary. We here introduce a technique for managing technical debt based on Visual Thinking. The technique addresses the problem of knowing how much debt is in place and how it is affecting the development cycle.

Technical debt; startups; visual thinking;

I. INTRODUCTION

Technical debt is a well-known phenomenon in software engineering. When technical debt is issued, a compromise is made while performing a technical task. Fulfillment of the task shall be accomplished in the shortest time possible at expense of a below standards solution. Such compromise is likely to have consequences in the future as dealing with messy solutions makes new changes harder.

Early stage startup context makes issuing technical debt a common decision. However, not managing such decision could result in harmful outcome such as the death of the startup itself.

We here present a technique for managing technical debt based on Visual Thinking. The technique provides a way to keep track of how much debt has been issued and how it is affecting the development process.

II. TECHNICAL DEBT

As far as we can tell the concept of technical debt was introduced by Ward Cunningham in 1992. Cunningham first presented the concept in the experience report titled *The WyCash Portfolio Management System* [1]. In the report he stated that “every minute spent on not-quite-right code counts as interest on that debt”

Martin Fowler gives one of the best definitions on technical debt in his article Technical Debt [2], presenting it as follows “You have a piece of functionality that you need to add to your system. You see two ways to do it, one is quick to do but is messy - you are sure that it will make further changes harder in the future. The other results in a cleaner design, but will take longer to put in place. (...) Doing things the quick and dirty way sets us up with a technical debt, which is similar to a financial debt.”

In order to gain deeper understanding of technical debt let us consider a few examples:

- As the time to reconcile two databases is longer than available you decide to write glue code to make them work together.
- You receive code by third parties that does not meet standards and you decide to fix it later.
- You decide there is not enough time to write tests and that you will write them after the release.

The technical-financial debt analogy applies perfectly to the above examples when examined in the light of interests. In financial debt interest is the charge for the privilege of borrowing money. When considered technical debt, interests are the charge for the privilege of releasing new features early, skipping a check or, generally speaking, doing things the easy way. Often technical debt interests are expressed as an increment in development effort. Technical debt interests can hurt companies and have important consequences, as we will discuss.

A final note on technical debt should be made. Debt can be generated both unintentionally and intentionally. Unintentional debt is the non-strategic result of doing a poor job. Intentional debt is a cost of doing business, is a mean to an end.

III. STARTUPS CONTEXT

By Paul Graham's definition a startup is a company designed to grow really fast [3]. One of the first problems one has to solve when creating a startup is which idea to pursue in a fairly efficient market. When the price is big, the competition is ferocious.

As the core idea is defined, there is only one thing that matters for an early stage startup: finding product-market fit [4]. Product-market fit is the only thing all members of a startup should concentrate on and has been well studied.

When considered the Lean Startup methodology [5] searching for product-market fit means building fast, testing and iterating. Under this scheme, technology is meant to support an experiment, a way to test a concept. Experiments are expected to be small and to the point. In worst case scenarios, technology could be thrown away and rebuilt. It is not worth doing things right until you know you have solved a problem that a large number of people have.

Furthermore, considering that startups usually have scarce resources, taking the easy path, making things the easy way is often the chosen direction.

IV. STARTUPS AND TECHNICAL DEBT

As we previously discussed and agreed upon, early stage startups are driven by product-market fit. Technology is just seen as a mean to an end which is testing the product. In this context companies tend to generate large amounts of technical debt as the goal is releasing as often as possible.

Nonetheless, issuing technical debt is not without cost even for startups [6]. Even more, despite the lack of scientific studies, some might argue that it can be even more dangerous than for established business as a startup could fail because of technical debt as is the case of Friendster [7]. Another famous case of technical debt is those that Twitter and Instagram suffered [8]. Companies have also informally stated that they have stopped acquisitions because of technical debt involved in the purchase.

Even if we spare the need to be fatalistic, technical debt can have counterproductive consequences. The worst for a startup would be to slow down the lean cycle discussed before. A startup that cannot implement new features quickly because their code base has technical debt is less likely to succeed.

V. MANAGING TECHNICAL DEBT

As technical debt cannot be eliminated we need a way to manage it. Previous work has presented techniques to identify, measure [9] and manage [10] technical debt but none has proven to be able to influence decision makers.

One of the tricky things about technical debt is its impossibility to measure effectively. Visual thinking is defined as a set of tools for making intangible or complex ideas visible. Considering that strategies such as Visual Thinking, Human Centered Design and Design Thinking have gain more and more relevance in the startup ecosystem we need to consider solutions based on these methods.

We would like to introduce a technique based on these methodologies that we named duct taped technical debt. This technique basically consists of taking a duct tape roll and cutting a piece every time you issue technical debt. In the physical world one fixes things duct taping them knowing the solution is only temporal. Each piece of duct tape will represent some part of your code that was developed or fixed in a messy way.

The size of the piece you cut should correspond to how big of a debt you are issuing. Much as story points in Scrum there is no correspondence to an objective measure as we cannot objectively measure technical debt. However, relativity between the piece's size will give you all you need.

Up to this point we have shown how to represent technical debt present on your code. However, the amount of debt you have is not important if consequences are not considered.

As mentioned before the worst non-fatalistic consequence of technical debt is slowing the lean cycle,

which in development terms means hurting team's productivity. We need to represent this.

Imagine yourself in the physical world trying to fix a broken bicycle. The first piece might have aesthetic implications. The second one will slow you down. The third one will reduce your speed significantly. The same analogy applies to technical debt. Each time you feel that the development team is going slower than they should because of technical debt, take a previously cut piece of tape and stick it somewhere it can be seen by everyone, a duct tape board we call it. Whenever you see your Velocity chart go down, assuming you are using Scrum, check your duct tape board and you might find the reason. Following this steps you will identify productivity damage produced by technical debt.

Implementing the whole technique will allow you to see how much technical debt you have issued and how much interests you are paying.

VI. CONCLUSION

We have reviewed basic concepts regarding technical debt and startups context. An analysis of how technical debt impacts early stage startups was presented. Finally, stating that technical debt cannot be eliminated we proposed a methodology to manage it using visual thinking technique.

Future work includes case reports on the usage of the methodology on startups on different stages. Incorporating duct taped technical debt into development process in companies other than startups could also be considered as opportunities for future work.

ACKNOWLEDGMENT

We thank anonymous entrepreneurs from Argentina who provided insights and expertise both when designing the technique proposed in this article and writing this article.

We would also like to show our gratitude to the Centro de Innovación Tecnológica, Empresarial y Social (CITES) for the opportunity to write this position paper.

REFERENCES

- [1] W. Cunningham, "The WyCash Portfolio Management System", Experience Reporte in OOPSLA '92, March 1992.
- [2] M. Fowler, "Technical Debt", Martin Fowler's Blog
- [3] P. Graham, "Startup = Growth," in Paul Graham's Blog, September 2012.
- [4] M. Andressen, "The only thing that matters," in The Pmarca Guide To Startups, June 2007.
- [5] E. Ries, "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses," Crown Business, September 2011.
- [6] S. Blank, "Startup Suicide – Rewriting Code," in Steve Blank Blog, January 2011.
- [7] S. Altman, tweet:
<https://twitter.com/sama/status/630782913009725440?lang=en>
- [8] HackerNews thread: <https://news.ycombinator.com/item?id=9949664>
- [9] N. Zazworka, and Carolyn Seaman, "Identifying and Managing Technical Debt", webinar presented by IEEE and Boeing, June 2012.
- [10] H. Nilsson, and L. Petersson, "How to Manage Technical Debt in a Lean Startup", October 2013.