Understanding the Impact of Technical Debt on the Capacity and Velocity of Teams and Organizations

Viewing Team and Organization Capacity as a Portfolio of Real Options

Ken Power
Collaboration Technology Group
Cisco Systems
Galway, Ireland
ken.power@systemagility.com

Abstract—Understanding the impact of technical debt is critical to understanding a team’s velocity. For organizations with multiple teams and products, the impact of technical debt combines non-linearly to impact the organization’s velocity. We can think of the capacity of a team as a portfolio. Not all of that capacity can be invested in new features or defect fixing, without incurring negative consequences. A portion of the team’s capacity needs to be invested in the ongoing management and reduction of technical debt. This paper describes a simple technique for visualizing, quantifying and tracking a team’s technical debt as a portion of their overall capacity investment. The knowledge and insights gained through this technique help with better capacity planning, improved forecasting, and helps to justify the business case for investing in managing and reducing technical debt.

Index Terms—Technical debt, visualization, tracking, capacity, team capacity, organization capacity, portfolio thinking, real options, velocity, team velocity, organization velocity.

I. INTRODUCTION

This paper is part of ongoing research to understand the causes and impacts of technical debt on product development teams and organizations. The context of the research is a large technology organization developing products and systems for real-time voice, video and messaging communications. All of the teams in this organization use an agile approach based on Scrum, Kanban, XP, or some combination of these. This organization, like most, is impacted by technical debt. What is needed is a pragmatic and useful way to quickly understand how much technical debt they have, how much they can afford to invest in debt reduction, and the positive or negative impacts of that investment. This paper provides an insight into one aspect of the organization’s strategy for managing and reducing technical debt.

II. TECHNICAL DEBT

Technical debt is a metaphor introduced by Ward Cunningham to explain the need for refactoring, and the impact of design choices on a software product [1]. Authors including McConnell [2], Fowler [3], Gat [4] and Sterling [5] have expanded the metaphor since.

III. CAPACITY AS A PORTFOLIO OF OPTIONS

A. Portfolio Thinking

A product development team has a finite amount of time to invest. An organization (whether business unit or entire firm) has a finite amount of capacity to invest in its overall product development efforts. It is useful to think of how we invest that time as a portfolio of options. Other authors have also considered a portfolio approach to managing technical debt [6, 7]. This paper does not directly draw on their work, but future publications based on ongoing research and practice will draw comparisons and show relationships.

B. Real Options Theory

Investopedia defines a Real Option as “An alternative or choice that becomes available with a business investment opportunity. ... Taking into account real options can greatly affect the valuation of potential investments” [8]. Weaver notes that a “firm may acquire real options by the learning developed from embarking on new areas of activity” [9].

It has been shown that Real Options can be used to map options to projects [10]. In his work, Luehrman presents a framework for understanding, evaluating and selection options in projects. The work in this paper does not present any of the mathematical foundations and analyses of Real Options. Ongoing research by this author is looking at ways to make that relevant to the capacity allocation of teams, while considering debt reduction as a real option for investment purposes.

C. Team Capacity as a Portfolio of Options

Consider the capacity of a team as a portfolio of options. For example, the time could be invested in many different activities including research, developing new feature, planning, fixing defects, design, architecture, customer support, documentation, training or other activities. Like a financial portfolio, if the team’s capacity is not invested wisely then the return will be lower than desired. Many teams fail to invest adequately in managing and reducing technical debt. This paper proposes that teams and organizations can benefit from
viewing technical debt management and reduction as a vital component in a healthy portfolio.

**D. Organization Capacity as a Portfolio of Options**

Few organizations consider or quantify their organization capacity and organization velocity. Organization velocity is a measure of how much value an organization is delivering over its entire product range. The timeframe is usually quarterly rather than at the end of each Sprint or iteration. By examining how teams across the portfolio are investing their capacity, the organization can get a better understanding of problem areas, investment levels in debt reduction, and generally how it is investing its money.

**IV. CHALLENGES AND APPROACHES**

This section presents some challenges surrounding technical debt management, and how we approach them.

**A. Agreeing what Technical Debt is**

Many people confuse technical debt with quality debt, or other forms of debt. There is a risk that teams will classify defect fixing as technical debt reduction, and therefore think they are investing in technical debt. As part of our work with teams we clearly differentiate between technical debt and other forms of debt, and clarify the investment in each distinct area. This organization uses ‘quality debt’ to refer to defects.

**B. Quantifying Technical Debt**

Teams need a quick and easy method of quantifying technical debt. For teams that are already under pressure we need a method that is non-intrusive yet easily relatable. We have found team capacity to be a very effective way of quantifying the level of investment in technical debt. Team capacity and velocity are useful ways to describe the impact of technical debt.

**C. Visualizing Technical Debt**

Pie charts and bar charts that show how much capacity the team is investing in reducing and managing technical debt are very useful. They quickly show the relative effort in proportion to the other areas the team is spending their time.

The following diagram in Figure 1 shows a healthy investment portfolio. In this example the team is investing its capacity not only in new feature development, but also in technical debt, quality debt, spike tests, research and planning activities.

**Figure 1** shows approximately 15% of the team’s capacity is invested in managing and reducing technical debt. In our experience, if teams neglect technical debt for the sake of short-term gains in feature velocity then their feature velocity will reduce over time anyway, and when they do come to pay back the debt their investment needs to be higher, as shown in Figure 2.

![Figure 2: Impact of not investing in technical debt reduction](image)

Both Figure 1 and Figure 2 apply to the same team and relate to investment in successive releases. Figure 2 shows a typical scenario where the team did not invest enough in technical debt reduction in release 1, which contributed to both increased quality debt and a higher technical debt. This resulted in overall lower feature investment, from about 60% to about 40%. Very often, it is just the feature investment that is visible. Stakeholders observe a reduced feature velocity but do not see where the team’s efforts are actually spent. Using these methods provides transparency and allows for more informed decision-making.

**D. Tracking Technical Debt Over Time**

It is necessary but not sufficient to understand the impacts of technical debt, and the level of investment in technical debt management, at a point in time. It is also necessary to track this over time. This allows teams to compare their planned investment versus actual investment. For example, they may have been forced to deal with unplanned or unknown technical debt that was discovered during a Sprint or iteration. Figure 3 shows how some teams choose to highlight this difference in a Sprint review.

**Figure 3**: Planned versus unplanned technical debt

**E. Impact of Neglecting Technical Debt Over Multiple Releases**

If neglected or simply underinvested in, technical debt will have negative impacts on a team’s feature velocity. A team might get away with ignoring technical debt for a first release, or even a second release, but this is not a sustainable strategy.
The bar graph in Figure 4 shows the effects in a typical case over four successive releases. Eventually the technical debt catches up with the team. In extreme cases we have seen teams forced to spend an entire release cycle almost exclusively on trying to reduce technical debt and quality debt.

This can also be represented as a Line Graph, as shown in Figure 5.

G. Understanding the Cost of Delay

Jeff Sutherland highlights some of the dangers for Scrum teams if they neglect technical debt [11], writing “If some people think the work is done at the end of the Sprint, but it really isn’t done, people are going to have to go back and re-do that work. We know that if you have to do the re-work it will take you at least twice as long to do it, and we’ve seen it take as much as twenty-four times as long”. For this reason we encourage teams to include technical debt management as part of their Definition of Done for user stories, Sprints and releases.

V. Examples

This section considers two examples of product teams in a large communications technology company, with real data.

A. Team 1: Actively Investing in Debt Reduction

Figure 6 shows output of a team’s planning session where they actively choose to invest in technical debt management.

This shows the team’s feature velocity is falling rapidly over successive releases, while its technical debt and quality debt are rising. More time is invested in technical debt reduction to deal with the mounting backlog of debt items.

F. Identifying Technical Debt as a Root Cause of Defects

In the rush to fix defects it can be tempting to come up with a fix that appears to solve the immediate problem. However, technical debt incurred earlier in a product’s history may be the actual root cause of the defect. In fact, a specific technical debt item might be the root cause of several defects. Ignoring the root cause can lead to the same defects getting re-opened later after the team thought they were fixed.
Taken together, these show the team allocating 16% of their capacity to debt reduction, and planning to repay the debt between end of February and April.

B. Team 2: Neglecting to Invest in Debt Reduction

Teams review the planned versus actual capacity allocation at the end of a Sprint or other formal review period. In this example Team 2 deliberately chose to not invest significantly in planned technical debt reduction, opting instead to target new features and defect fixing. What they failed to realize in advance is that many of the defects were caused by longstanding technical debt that was starting to destabilize the system. Figure 8 shows the team’s actual capacity allocation after their Sprint 8.

![Figure 8: Capacity allocation shows impact of low technical debt investment](image)

Figure 8: Capacity allocation shows impact of low technical debt investment

Figure 9 shows the options the teams chose to invest in more detail. The figure highlights the amount of effort the team planned (in points), the committed work that was accepted (in points), and the total number of points accepted. The distinction here is that the team was forced to take on new unplanned work during the Sprint. So, even though they were underinvesting in technical debt reduction already, they came under pressure to reduce that investment even further once the Sprint was underway.

![Figure 9: Planned versus actual investment in debt reduction and other areas](image)

Figure 9: Planned versus actual investment in debt reduction and other areas

Later retrospectives revealed that many of the root causes that contributed to the performance and robustness issues stemmed from accumulated technical debt. These charts, which show planned versus actual investment over time, help provide transparency, and show the consequences over time of under-investing in technical debt reduction.

VI. CONCLUSIONS

This paper described a technique for considering the capacity of your team as an investment portfolio. Investing in technical debt management and reduction needs to be a part of a healthy portfolio. If neglected, a team’s Technical debt will mount over time and impact their feature velocity. Consider the different ways a team could invest its time as Real Options. Make investments in debt reduction explicit and visible, and track actual investments at regular periods. Taken to an organization level, the organization needs to be aware of the amount of technical debt it has, and the overall strategy for investing in managing and reducing that debt.

ACKNOWLEDGMENT

This paper is derived from the ongoing work of Cisco System’s Agile Office, part of the Clients and Mobility Business Unit (CMBU). The author, along with Tom Lambert and Todd Weber, are co-founders and core members.

REFERENCES