The True Cost

There is no universally agreed upon way to measure cost. Cost is a relative term, and it is also contextually sensitive. Cost in a society that uses a barter system is vastly different than cost in a society that relies on currency. There are also occasions when cost becomes irrelevant. For example, a society may value certain pieces of art or historical artifacts and label them “priceless”. The value of such an object when translated into the dynamic of cost cannot be measured. Some things, in other words, have intrinsic value that transcends relative and contextual measures of cost.

For the past twenty to thirty years, libraries have also struggled with the issue of cost and benefit relative to technological advancement[1]. The struggle is not ultimately about the financial aspects of the equation, but rather the value that can be gained, the opportunities taken or lost, and the questions of relevance and service. This struggle is not unique to libraries, especially regarding advancements in technology. Industry, which is often the producer of new technology in collaboration with academic institutions, faces issues of cost benefit analysis in more acute fashion. For industry, cost influences fiscal health, and fiscal health is the measure of vitality for any organization that is a part of capitalist economies.

Libraries, though, are not tied directly to fiscal responsibilities. Certainly, libraries need to be fiscally responsible, but their primary focus is content and services for their stakeholders: the user, patrons, students, faculty that use their content and services. This shapes the burden of cost analysis in a unique way for libraries. Budgetary concerns are of course a part of the analysis – salaries and discretionary line items scope what can be accomplished. However, the true cost, relative to being technologically relevant, is rarely for libraries directly tied to fiscal oversight. This isn’t to understate the value of money – on the contrary budgets are necessary and need to be carefully scrutinized. Rather, the full cost regarding technology is measured in a way similar to the priceless work of art. The reason for this is due to the fact that, for libraries, service to the patron is of inestimable value and is the ultimate goal.

While fiscal responsibility is a part of the equation for measuring cost, there are other almost equally important factors. Time is probably the number one factor in addition to financial cost that is highly valued. This is true for a couple of reasons. First, time is limited. Time translates directly into capacity. Second, because technology advances so rapidly, the window that libraries have to leverage certain technologies in service to patrons is also quite limited. Time is spent analyzing the landscape, planning, and delivering technologies that are relevant. The necessity to view time as a commodity creates somewhat predictable scenarios from an analysis perspective. Regardless of the technology being considered, questions about time to implementation, time to maintain, time budgeted for other projects or services, and others almost always emerge.

Management will sometimes evaluate the cost relative to the amount of time *saved or gained*, the amount of time *lost*, etc. Along with finance as a factor, analysis is sometimes obsessively concerned with matters involving time. Because the time factor can sometimes skew conversations regarding cost, undue pressure can be placed on those responsible for managing time or estimating time. This results in time overestimation if the team or manager is conservative, or time underestimation if the implementation team feels the need to pad their estimation.

Time and financial burden are two of the most important elements of cost estimation. They are important, but they can also deflect from important strategic thinking that needs to take place in libraries regarding technology, and can also mask the importance of other cost factors such as reputation and opportunity. The tension that is present in industry between software specialists and management regarding cost factors is also present in libraries, even if libraries are not employing development teams. The decisions around cost relative to financial and time concerns are still present. For larger academic institutions that do employ development teams, though, the issue of time as a concern is ubiquitous because management is trying to be as responsible as possible regarding the aggregate pool of resources available.

Quality is a cost factor that is often overlooked. Quality emerges at several levels but the aggregate is of most importance to decision makers focused on digital products and services for the library. Providing a quality experience and increasing levels of satisfaction is a primary deliverable for any library. When considering cost, the related costs of quality, opportunity, relevance and reputation seem to greatly outweigh the costs of finance and time. This isn't immediately obvious, though, and the obsession with the immediate concerns of the financial burden coupled with potentially wasted time can eclipse their value. The root of the problem, however, is not that library professionals in management positions can't see the value of these factors. They do, but they are not as immediate and the focus on the other factors obfuscates the more important strategic vision that upper management should be focusing on.

With cost estimation driving strategic concerns related to technology in libraries, it's difficult to be reflective about ways to make a paradigm shift toward a culture that has the concerns placed appropriately and satisfies the concerns of cost analysis. Taking a closer look at a more agile method for analysis and decision-making could offer a set of solutions that could benefit library organizations in making more strategic technology decisions moving forward.

Optimistic Appraisals

Strategic decisions regarding technology can be complicated and involve a myriad of facts and concerns that need to be accounted for. Typically, though, there is an inverse relationship between those factors that should have the

highest influence in decision making at the strategic level, and those factors that should probably be taken into account at the implementation level. As was alluded to previously, the visionary level of management can get bogged down with concerns about time and finance when those concerns are secondary to making the right strategic decision. The most important concerns relate to the opportunities that can be taken advantage of, and positioning library technology so that it is more future proof and compatible with patron needs. Innovation is rarely born out of concerns involving time and finance.

On the other hand, development and implementation teams are also stymied by those concerns as well. There are a couple of reasons for this. If an implementation team has come to the conclusion that a solution will be expensive, they will feel obligated to meet the expectations of management and under or over estimate the cost. They may also feel compelled to ignore the best solution in favor of the merely cost effective solutions. Also, the total financial and time cost estimates are usually inaccurate due to a variety of factors. The implementation team is then forced to provide inaccurate information to a management team who will use that information to make strategic decisions. In the end, it becomes a lose-lose scenario, the severity of which is dependent on the level of inaccuracy involved.

A possible method of alleviating this situation is to consider how roles are assigned relative to product development and implementation. If a solution has been given the green light by upper management, quite often the implementation team (development team or application team) will be assigned several responsibilities. They will need to become the champion of the product or solution and they will also be tasked with communication to the organization about the solution, which could involve a large number of stakeholders. They may also be responsible for making sure that the timeline for implementation is followed, that the requirements are met, that testing occurs, and that the solution is implemented in a sustainable fashion. This burdens an implementation team with far too many responsibilities and dilutes the ability of team members to apply their appropriate expertise to solution implementation.

An approach that is being widely adopted by software development teams as well as a variety of other sectors called “agile scrum” recommends that at least three discrete roles be determined and empowered when a product or service is in development[2]. None of the roles involve management, but instead focus on individuals who do the work of implementation[3]. Here’s a brief explanation of each of these roles.

1. **Development or Implementation Team** - This includes anyone doing the work. Software specialists, designers, UX analysts, content creators, application administrators, systems administrators, dev/ops, data analysts - anyone who is a part of the team **doing the work**.

2. **Product Owner** - This should be a single person who can represent the needs of the customers, who understands and can explain the requirements and desired outcomes, and who can provide sign off. This person needs to be accessible and able to answer questions posed by the development team.
3. **Scrum Master** – If the organization supports a project management group, it's important that a project manager fill the role of Scrum Master. Another designated person could fill this role. The Scrum Master is not a part of the implementation team, and should not be a person who is involved in doing any of the work. Their responsibilities are to run the daily scrums, track progress the of user stories, and remove any roadblocks that development team members encounter. The Scrum Master should also schedule sprint reviews.

These assignments add two important elements to the analysis of how to solve problems regarding technology and the situations libraries currently face. First, the implementation team is focused on doing the work necessary to meet all technical requirements associated with a product or service. Second, the product owner handles all communication with stakeholders, subject experts, consumers or patrons, as well as all other external factors that influence requirements. In addition, the scrum master is the glue between the implementation team and the product owner. This role is responsible for removing roadblocks that would otherwise be very time consuming for the implementation team, and making sure that things stay on track regarding the commitments that have been made.

Regarding cost analysis, the central element of agile scrum is *estimation*. The method for doing estimation is critically important and is the hub upon which everything else turns[4]. The reason for this is simple: estimation in agile scrum is about *actual output* not an arbitrary measure of time or other cost measures. Estimation of work is, from the planning perspective, the most important aspect of any process employed by teams. But it's important to be clear about *what* is being estimated. We sometimes refer to estimation as estimation of *time*, estimation of *effort* or estimation of *work*. For capacity planning purposes though, we're not estimating any of these factors. In order for estimation to be meaningful, the factor that requires estimation is **team output**. In other words, what can the team deliver in a given time frame - managers, scrum masters, product owners, and stakeholders aren't interested in how much work or effort is required to complete a task. They need an approximate but fairly reliable way to measure team output over time.

In order to measure output, it's necessary to have a ratio that isn't composed of only time. A factor is required to measure *against time*. The scale of time that's relevant to those who need to make strategic decisions about priorities at the macro and micro level will vary. Product owners will be measuring output over weeks, and perhaps a month or two. Managers, directors, and upper management will want to measure output at a higher level, over months if not

years. Implementation teams, if they're being accurate to any degree, will probably try to estimate time in units much smaller than is relevant to either of those groups. While it's theoretically possible to aggregate work hours into weeks and months, the value of that aggregate is increasingly small the larger the project becomes.

Estimating tasks according to *time* isn't feasible for many reasons. First, as was mentioned, it's impossible to reliably measure output when time is the only factor because then the ratio for measurement wouldn't make sense. A ratio of time over time isn't reasonable. The only way for time to possibly be effective as a ratio is if the estimations become very granular, for example at the *hourly* level. Experience and study have shown that hourly estimates are *almost always* highly inaccurate and counterproductive[5]. Using hours to estimate tasks will result in one of many undesirable situations. If a specialist is held accountable for the hourly estimate, they will be pressured to complete a task - regardless of quality - in the time frame originally estimated. Or, more commonly, the specialist will take longer to complete the task than was originally estimated. Both scenarios could result in reputation cost for the organization as estimations slip over the course of a project.

Another problem with using actual duration as an estimating factor is that time, regardless of granularity, is an **input** measure not an **output** measure. For example, a specialist might estimate a task at 2 hours and let's say for example that they in fact complete the task in 2 hours. The next work iteration they work on the same task (or type of task), and they complete it in 30 minutes because they learned from their previous experience. Did they accomplish 25% of what they accomplished the last sprint because it only took the ¼ of the time? No, the output was the same. Looking at another example, if one specialist completes a task in 4 hours, but another specialist takes 2 hours to complete the same task, did the second specialist accomplish half as much as the first specialist? No, they both had exactly the same output. What's important from a planning perspective is not the time taken to complete a task, but the output of the team. Estimations using time are by nature an input measure. The primary problem with estimating according to duration is that it's almost always inaccurate. This has been proven in the software industry through many studies that have been done. Research into time estimates in industry began in the first half of the 20th century by the Department of Defense who hired the Rand Corporation to study the issue in depth. That study resulted in the point system of relative estimation. Other companies such as Microsoft have reproduced the Rand study recently and their results were nearly identical[6]. As an alternative to time, using a point system such as the Fibonacci sequence to estimate tasks results in estimates that are relatively unbiased and far more accurate. The evidence against the reliability of absolute values for task estimation is substantial, regardless of what absolute standard is used (e.g. - time, quantity of work, etc.). A 2006 IEEE study looked at over 100 development projects that they had engaged in along with more from an independent study[7]. The author found that the majority of estimates using an absolute measurement such as time are almost always inaccurate.

In order for historical data measuring the quantity of output over time to be valid, it's also important to establish **fixed reference points** for estimation. When a team initially begins estimation, they should establish a reference task at about 3 points of complexity. The reason a reference point is valuable is that it gives a standard by which other user stories can be estimated. The estimation team can then look at that reference story that they've agreed upon, and the **relative** complexity of other stories can be measured against the fixed point. Let's say we had to give an estimate of how long a room is, but we didn't have a measuring tape. If we estimate a certain distance from the wall as being about a foot, or three feet, for example, then we could look at any other arbitrary length along the floor and give a fairly accurate estimate of how far along a straight line that point is. Without an established reference point, the accuracy of that estimate would be much less trustworthy.

Using a relative method of estimation frees the development team of needing to be concerned with time as a bounding principle, and shifts the focus to quantity and quality of output. Instead of needing to worry about the overhead costs, the team can focus on *individuals, interactions, working software, and customer collaboration.*

The Road to Results

When the previously mentioned pieces are sewn together, though, how does it result in a win-win scenario for upper management who stewards the vision for the organization, for those responsible for project and team management, and for the implementation and development teams? Agile scrum is counter-intuitive for many organizations because they are wedded tightly to measuring cost by time and financial concerns. It's difficult to conceive of product implementation as a whole when the requirements are broken down into digestible chunks. This can make it difficult to do long term planning at first because it is a shift from thinking about products and services as cost centers to thinking about them as strategic concerns that have opportunity and reputation costs. The crux of the problem is that decision makers at the upper management level want to avoid risk as much as possible, and would rather have guarantees or leverage to achieve results that are reliable. Risk management is present when making any strategic decision regarding technology in libraries but it can also be another dominating factor when the focus for visionary leadership should be on prioritizing digital products and services that best serve patrons[8].

To the chagrin of library leadership, change relative to technology can occur at such a rapid pace that it make reprioritizing or changing direction difficult. For example, longevity assumptions could be made about social media or mobile applications that effect planning that is scheduled for multiple years. Those assumptions could easily be undermined by any number of factors including technology advances, deprecated hardware features, ephemeral user interest and other unpredictable circumstances. In other words, change is inevitable.

As a thought experiment if we treated libraries in an analogous way to commercial entities, a goal of leadership would be to do strategic planning that would anticipate and outpace competitors. But in order to make that type of planning effective, flexibility also needs to be factored in. To be relevant and in tune with patron expectations, library organizations need to be nimble enough to change strategies fairly quickly. Thus, in order to avoid opportunity mishaps, change needs to be embraced and in a sense, *planned for.* This is possible using agile scrum because change is entailed in the process. The basic contract between the implementation team and the product owner is to accomplish a certain quantity of work over the course of an iteration that may last one, two or more weeks. As long as that contract is honored, then expectations are met. If upper management chooses to shift focus it can be done rapidly if estimations have been done on actionable items that in aggregate represent a project. Tasks that need to be temporarily abandoned can later be picked up in a future iteration if the focus switches back to that product or service.

In addition to flexibility, agile scrum requires an organization to treat time as a budgeted commodity instead of a primary decision metric. Once priorities have been determined and strategic concerns defined, product owners can work with management to negotiate how time should be expended. If opportunities arise that take precedence over existing projects, the decision will be based on what is more strategically important, and what will have a greater opportunity or reputation cost for the library. Instead of focusing inward on budgetary issues and dealing with unmanaged expectations, leadership can focus on the more important areas of content and service improvement. And this, in the final analysis, is the metric against which the true cost should be measured.

---

[1] P. Pandey, "Analysis of the Techniques for Software Cost Estimation," *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on*, Rohtak, 2013, pp. 16-19.
[2] B. L. Romano and A. D. d. Silva, "Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise," *Information Technology - New Generations (ITNG), 2015 12th International Conference on*, Las Vegas, NV, 2015, pp. 774-776.
[3] R. Ramanujam and I. Lee, "Collaborative and competitive strategies for agile scrum development," *Networked Computing and Advanced Information Management (NCM), 2011 7th International Conference on*, Gyeongju, 2011, pp. 123-127.
[4] S. Grimstad, "Understanding of estimation accuracy in software development projects," *11th IEEE International Software Metrics Symposium (METRICS'05)*, Como, 2005, pp. 2 pp.-42.
[5] D. V. Ferens, "The conundrum of software estimation models," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 14, no. 3, pp. 23-29, Mar 1999.

[6] L. Williams, G. Brown, A. Meltzer and N. Nagappan, "Scrum + Engineering Practices: Experiences of Three Microsoft Teams," *2011 International Symposium on Empirical Software Engineering and Measurement*, Banff, AB, 2011, pp. 463-471.
[7] T. Little, "Schedule estimation and uncertainty surrounding the cone of uncertainty," in *IEEE Software*, vol. 23, no. 3, pp. 48-54, May-June 2006.
[8] B. Gold and C. Vassell, "Using risk management to balance agile methods: A study of the Scrum process," *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, 2015, pp. 49-54.