In a McKinsey article published in July 2016, “Adapting your board to the digital age,” Hugo Sarrazin and Paul Willmott reported that “less than one in five directors fully understand how the industry dynamics of their companies are changing.” They quote Marc Andreessen’s light-hearted but pertinent observation that, “Software is eating the world.” They also reported an expectation among one in three McKinsey clients that their business model will be disrupted in the next 5 years. Sarrazin and Willmott go on to assert that, “Board members need better knowledge about the technology environment, its potential impact on different parts of the company, and its value chain.” Hunter and Westerman (2009) point out that, “Ensuring that value is delivered is a process, and a process must be managed. After IT solutions are delivered and business change is complete, the project steering group is replaced by a harvest steering group.”

I suggest that what the board and the company need is a software value management office (SVMO), responsible for measuring and monitoring the value delivered by information technology (IT) and specifically software development. Why specifically software development? Because IT is increasingly a utility—it needs to be properly designed, implemented, and managed, and technical expertise is needed to do all of that well. However, I suggest that most of the company’s differentiated IT value will come from its software.

Where Does the SVMO Fit in the Organization?

The concept of a value management office (VMO) is not new. Before I explain where the SVMO fits in the organization, I want to consider some of the other concepts for VMOs so that we can see if the SVMO would fit within an existing VMO. I have chosen a small, representative sample of VMO ideas that are out there.

In September 2015, Tom Pisello* wrote a blog post about his ideas for a VMO to be “the center of excellence aligning value marketing, selling and consulting,—with the mission of proposing, proving and improving value to customers.” Pisello is following a similar thought process to me (or actually I to him) insofar as he is focusing on the importance of delivering value to customers. Hence, the philosophy of Pisello’s VMO is well-aligned with the SVMO, but I’m guessing that Pisello’s VMO would prefer not to engage with software development. That being said, Pisello’s VMO would make an excellent customer for the SVMO. Therefore, while I believe that there are definite organizational benefits that need to be considered in Pisello’s ideas, I don’t recommend including SVMO within Pisello’s VMO.

Kaplan et al. (2015), writing in the Harvard Business Review in 2015, floated the idea of a VMO in the context of health-care organizations,

A ‘value management office’ can greatly enhance an institution’s ability to improve outcomes and costs across the enterprise. At a minimum, it can serve as a center of excellence to assist decentralized clinical units in outcomes and cost measurement and management, set priorities for continuous improvement projects, facilitate the creation of value-based payment models with insurers and employers, and ensure that new information technology platforms are aligned with the value agenda.

This description is much closer to my idea of an SVMO in that it is a center of excellence that considers the end-to-end value delivered by the organization both in terms of models and measurement. It recognizes the needs of multiple stakeholders—insurers and employers in this case. It even takes responsibility for ensuring that the IT activities are aligned, which reinforces my position that the business units and IT organizations need to collaborate to drive the greatest value from their software.

Kaplan’s article gives two examples of health-care organizations that have implemented VMOs. I think that the SVMO could be easily accommodated in Kaplan et al.’s VMO, but more specificity about the role of the SVMO is needed. That is the purpose of this chapter.

The Office of Budget and Management (OBM) of the State of Ohio in the United States has a VMO with an excellent explanation of its motivation, which is noted as follows:†

---

† http://www.obm.ohio.gov/vmo/faq.aspx
What Is Value Management?
A structured approach similar to project management. Value management
- Extends across the lifecycle of an initiative, from conception to implementation and evaluation
- Helps organizations achieve the intended outcome of a program or an initiative
- Measures financial and nonfinancial benefits of initiatives
- Looks across agencies and silos to identify enterprise results and efficiency

What Value Management Is Not
Value management is not an “add on” to the business process.
- It is integrated and aligned with other change initiatives
- It requires a results focus

Value management is not a quick fix or a silver bullet.
- It involves sustained change over the long-term
- It entails changes in perceptions, relationships, management principles, and actions

Value management is not “one size fits all.”

Why Did (Ohio) OBM Create a Value Management Office?
Enterprise programs should be able to demonstrate achievement of expected benefits
- Process improvements and cost savings have not been captured to demonstrate benefits
- Some benefits have not been realized
- Future enterprise programs could improve cost recovery and savings for Ohio tax payers and agency constituents

While Ohio is clearly talking about the whole range of services that the state government delivers to its citizens, I would be happy to see all of the previously mentioned in a charter for any SVMO. Note that Ohio’s OBM sees the VMO as a project management function.

In February 2015, Andy Gill floated the idea on his blog* of a VMO to replace the project management office (PMO) in organizations under the title, “The PMO is dead! Long live the VMO!” Gill’s VMO would be

A business function, designed to oversee corporate governance, organisational change, quality, compliance and process management. The VMO is permanently staffed with a small core of Value champions to co-ordinate, act as gatekeepers and maintain continuity of the function but the ideas, requirements and solutions come from Value Action Teams from within the body of the organisation. These are virtual teams created from the people closest to the work, brought together to address real issues within their scope of expertise.

A supporting function, providing genuine assistance to the wider business. VMO staff should look at every activity they are involved with and be able to clearly articulate why they are doing it, who they are doing it for, what value it brings, and whether it is genuinely a business necessity.

A single conduit for all organisational change where it can evaluate the impact of change and how changes align with each other (or not) and prioritise accordingly. This helps to minimise the risk of change overwhelming the business, and the problem of multiple changes competing for the same resources.

Gill’s VMO concept is organizational rather than focused on software development and, even though this book is narrowly focused, Gill’s concept is consistent with my view of the SVMO because, as I have often repeated in this book, the value of software development can only be evaluated in an organizational context. Gill sees a functional VMO as a good replacement for dysfunctional PMOs and some of Gill’s ideas—business function, small core team enhanced with short-lived virtual teams—are included in my vision of the SVMO. The title of his blog post implies that Gill believes that VMOs can and should arise from the ashes of PMOs.

It is not clear if he intends the same people to transition from PMO to VMO. While I know many project managers whose interpersonal skills and analytical capabilities would make them excellent candidates for the VMO, in many organizations today the role of project manager has been downgraded to basic project administration. People in these roles may be important to support the VMO function, but they will not be capable of driving the work, especially in the early days, because of the need to challenge assumptions and produce creative ways to measure value.

Gill’s article includes some stark criticism of PMOs, which I have not repeated here. PMOs have been in existence in most large organizations for a long time. They were established to help standardize processes to ensure projects are executed efficiently and effectively and to measure that execution and effectiveness. PMOs and project managers have played an important role in streamlining organizations, but they are often disliked. Interestingly, they are often disliked equally among executives and staff. Executives see the PMO as always delivering bad news about missed deadlines and hold project managers accountable, or at least responsible, for
those missed deadlines. The staff perceives the PMO as parasitic—someone always looking over their shoulder to ensure the proper boxes are checked without taking into consideration the challenges these workers face in accomplishing the task at hand.

The big question is whether the PMO is actually driving business value or if it’s more focused on monitoring whether projects are on time and on budget. From a lean software engineering perspective, for maximum flow of value on a software development project, waste needs to be removed. In the lean world, waste means any activity that doesn’t add value.

In a Waterfall organization, the PMO is justified because it has become the oil that keeps the software development engine moving. But even in Waterfall organizations, the PMO is becoming increasingly commoditized. The PMO has an opportunity to transform from a group that standardizes processes to improve efficiency and measure compliance, to an SVMO that optimizes processes to maximize value flow and measures value.

Since Agile organizations tend to inherit a PMO of questionable value from Waterfall or not use one at all, the SVMO would be a new function. In Agile organizations, the function of the SVMO would be to get the business units to collaborate with the technical team on assigning value to features and epics and making that value visible so everyone is focused on driving value for the project.

What Is the Mission of the SVMO?
To make software value visible and to maximize end-to-end software value flow to customers.

What Are the Functions of the SVMO?
With thanks to Pisello, Kaplan et al., the State of Ohio, and Gill, I propose the key functions set out in Figure 9.1.

Like Gill, I believe that the SVMO is a business function, which needs to be permanently staffed with a small core of value champions to coordinate, act as gatekeepers, and maintain continuity of the function. I believe the only way that this can be done is if some of the permanent staff in the SVMO have respectable business credentials and some have respectable software development credentials. I further agree with Gill that the SVMO should be supported by virtual value action teams from within the body of the organization so that ideas, requirements, and solutions come from the “coal face” of the business and software development. By way of example, BT in the United Kingdom has something they call CIO offices. These CIO offices are part of the business units and sit at the interface between the business and software development. They tend to be staffed by respected former
<table>
<thead>
<tr>
<th>SVMO 1</th>
<th>To be center of excellence aligning value measurement between business and software development</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMO 2</td>
<td>To enhance an organization's ability to improve software outcomes and costs across the enterprise</td>
</tr>
<tr>
<td>SVMO 3</td>
<td>To set priorities for continuous improvement projects</td>
</tr>
<tr>
<td>SVMO 4</td>
<td>To facilitate the creation of value-based payment models</td>
</tr>
<tr>
<td>SVMO 5</td>
<td>To ensure that new information technology platforms are aligned with the value agenda</td>
</tr>
<tr>
<td>SVMO 6</td>
<td>To ensure that the SVMO mission extends across the lifecycle of an initiative</td>
</tr>
<tr>
<td>SVMO 7</td>
<td>To help organizations achieve the intended outcome of a program or an initiative.</td>
</tr>
<tr>
<td>SVMO 8</td>
<td>To measure financial and nonfinancial benefits of initiatives</td>
</tr>
<tr>
<td>SVMO 9</td>
<td>To look across the organization to identify enterprise results and efficiency.</td>
</tr>
<tr>
<td>SVMO 10</td>
<td>To ensure that software value management is integrated and aligned with other change initiatives.</td>
</tr>
<tr>
<td>SVMO 11</td>
<td>To ensure that the software value management retains a results focus</td>
</tr>
<tr>
<td>SVMO 12</td>
<td>To ensure that software value management is sustainable over the long-term.</td>
</tr>
<tr>
<td>SVMO 13</td>
<td>To manage changes in perceptions, relationships, management principles, and actions</td>
</tr>
<tr>
<td>SVMO 14</td>
<td>To demonstrate achievement of expected benefits</td>
</tr>
<tr>
<td>SVMO 15</td>
<td>To take a lean software engineering view and remove process steps that do not add value.</td>
</tr>
</tbody>
</table>

Figure 9.1  Key functions of the SVMO.
members of the software development team seeking a temporary or permanent insight into the workings of the business as part of their career development. Their recent experience of the software development organization makes them invaluable for translating across the interface.

Software Value of People

In Figure 9.1, SVMO function #2 is, “To enhance an organization’s ability to improve software outcomes and costs across the enterprise.” I want to make explicit here an important point about where some, or most, or all of the software value resides in organizations. It is in people’s heads. This software value in people’s heads needs to be managed explicitly, because it represents a risk to the organization’s ability to improve software outcomes. The SVMO is a good place to manage this risk because they can operate with some detachment from the personal relationships in teams and between teams and their managers.

Identifying the risk of loss of software knowledge is relatively easy. At the team level, the SVMO needs to address the question of whether there are any parts of the software that only one (two, three) people understand sufficiently well to be trusted to make changes. At the product/program level and portfolio level, this question might be applied to outsourced vendors. One of our clients reached a point in their long-term outsourcing strategy where they realized that they no longer had the internal knowledge to validate the estimates that their vendors provided for enhancing some of their applications. They quickly started to recruit people with the right skills to redress the balance.

Managing the risk of software knowledge contained in people’s heads is not easy. Even the most dedicated and loyal employee understands that being the only person to understand a piece of software (or any other organizational asset) represents leverage. Why should they share when doing so potentially reduces their usefulness and value to the organization? A “brute force attack” to this challenge never works. Instead, the right incentives need to be set up. These will be different for each individual, but many in software development will respond to the offer of passing on routine chores in return for more interesting work. Incidentally, the Agilists are right—documenting the scarce knowledge is not sufficient and will probably not work in most cases. You need to have at least two practitioners who are knowledgeable about each piece of code—this is a little-mentioned benefit of the practice of pair programming (now claimed by Agile).

How Should the SVMO Itself Be Measured?

The mission of the SVMO is, “To make software value visible and to maximize end-to-end software value flow to customers.” It should be measured against this mission, which I will break up into its two constituent parts.
How do we measure if the SVMO is making software value visible? The success criteria here is whether or not software value is visible throughout the organization. If we invoke the Scaled Agile Framework® (SAFe) model for software development again, there are four levels for assessing this: portfolio, value stream, product/program, and team. There are also two perspectives: Is the appropriate information available? Are individuals making decisions based on that data?

We can present the success criteria for this first set of SVMO metrics on the simple dashboard shown in Figure 9.2.

**Portfolio, Product/Program, and Team Measurement**

Each of the metrics for the performance of the SVMO in Figure 9.2 needs to be measured for each level: portfolio, product/program, and team because the nature of the software value information required will be different. Certainly, some of the data can be aggregated in Agile implementations. However, just because all stories have software value data associated with them for all teams in the program, it doesn’t necessarily follow that all epics at the portfolio level have software value data. In fact, this is probably close to the best case scenario today—all the product owners of all the teams in a mature Agile organization are assigning local value points. As I have said earlier in this book, I have no complaints about this scenario, because it is better than nothing. It represents a sort of local maximum, though, because those local value points should be derived in some way from business value data at the portfolio level. Put another way, the method for assigning value points at the team level will have to change before it can get better. There may be some resistance to this change among the teams.

**Software Value Information Available**

The first of the four success metrics for the SVMO, as described in Figure 9.2, is determining the software value information that is available. This is an entry-level

<table>
<thead>
<tr>
<th></th>
<th>Portfolio</th>
<th>Product/program</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software value info</td>
<td>Yes/No</td>
<td>Yes/No</td>
<td>Yes/No</td>
</tr>
<tr>
<td>% of individuals with access to software value data</td>
<td>0–100%</td>
<td>0–100%</td>
<td>0–100%</td>
</tr>
<tr>
<td>% of individuals using software value data to inform decisions</td>
<td>0–100%</td>
<td>0–100%</td>
<td>0–100%</td>
</tr>
<tr>
<td>Software value flow increase versus last quarter</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
</tbody>
</table>

Figure 9.2: Dashboard for measuring SVMO success at software value visibility.
metric for the SVMO—something to work on immediately. In a fairly stable organization, it should be quite easy to “switch on the lights” for these binary input metrics and keep them on. In a more volatile organization, perhaps with lots of new initiatives or in an outsourcing scenario, keeping these metrics positive will require ongoing diligence and training of new staff.

**Percentage of Individuals with Access to Software Value Data**

The second success metric is the assessment of the percentage of individuals with access to the software value data. This is an input metric and provides more granular data than the “information available” metric, because all individuals must have access if the software value data is to influence their decision making.

**Percentage of Individuals Using Software Value Data to Make Informed Decisions**

The next metric is the percentage of individuals who are using the software value data to make more informed decisions. I consider this to be an output metric, but I concede that it is a “soft” metric because the only way to gather the metric data is by surveying the individuals, perhaps quarterly. Like all surveys, it is reasonable to expect that some individuals will not be entirely truthful, especially when the desired behavior is so obvious. However, I believe this metric is worthwhile because, as a soft metric, it has the purpose of encouraging the right behavior rather than imposing it—always a good change management tactic.

**Software Value Flow Increase versus Last Quarter**

Measuring the software value flow increase compared with the last quarter is the final metric. This is the ultimate “hard” output metric for the SVMO—Is the SVMO making a difference to the flow of value? I would strongly recommend normalizing the underlying value flow metric here for reasonable comparison of historical data.

One simple normalizing approach could be to use a return on investment (ROI) type methodology by measuring software value delivered per $100 thousand of budget spent on software development in that quarter.

Another simple normalizing approach that might be useful at the portfolio and product/program levels (and team level for Waterfall) is to measure software value delivered per function point of software functionality delivered.

While the ROI approach represents the bottom line, it can hide nuances in the software value productivity of different parts of the software development team or vendors because of differences in hourly cost rates in different geographic regions. Executives reading this (including me!) might initially say that the ROI
is “the bottom line,” but exchange rates and local hourly rates can change, so I would recommend using both metrics. For example, it would be wrong to credit the SVMO with the 10% improvement in ROI-based value productivity for U.K. teams that was a result of the drop in that country’s exchange rate after the Brexit vote.

SVMO and Earned Value Management

As we talk about SVMO metrics, *earned value management* (EVM) comes to mind. As mentioned in Chapter 8, EVM is a specialized discipline that is most often used in the context of long-running U.S. federal government projects to enable the measurement of progress and, in some cases, enable stage payments based on that progress. The Project Management Institute (PMI) has published a simple, *Practice Standard for Earned Value Management—Second Edition*,” (as a complement to the PMBOK® Guide), which provides an accessible introduction to earned value management that allows me to summarize (and oversimplify) here. EVM consists of:

- Building a project plan by decomposing the high-level project activities into a set of dependent and independent work tasks, usually called the *work breakdown structure* (WBS).
- Each work task in the WBS has planned scope, duration, resources and, hence, cost (usually called *planned value* even though it isn’t really value—see the Rodrigues quote at the end of this chapter). From this data, a curve of planned value can be plotted on a cost versus time chart (See Figure 9.3) by calculating the number of WBS line items that are planned to have been completed by the chosen date and the amount of planned value associated with each item. In its simplest form, the planned value is simply the number of WBS line items due to be delivered by the chosen date (i.e., each WBS line item has an assumed value of one).
- Each WBS item has a planned or budgeted cost. Hence, there is a total cost associated with the planned value curve, which is known as the *budget at completion* (BAC).
- Once work has commenced, the cumulative *earned value* can be tracked by calculating the number of completed WBS line items and their cumulative value. As can be seen in my example (Figure 9.3), there is often variance between the planned value and the earned value. In my example, the earned value is less than the planned value. This is a scope delivery problem because less scope has been delivered than was planned by the report date. This suggests a possible schedule overrun. It is worth noting here that there are different opinions about what counts as a “completed” WBS line item. I prefer the straightforward binary interpretation, it is either done or it isn’t. The Agile concept of *definition of done* can be helpful here. However, I have seen
projects in which the project managers ask the teams for “percent complete” on their WBS line items and use those numbers to calculate a “rough” earned value. As a manager, I have heard “90 percent complete” from too many software development teams to trust anything based on percent complete.

In addition to, but separate from, the earned value calculation, the cumulative actual cost incurred to date can be calculated. It might be expected that with earned value running behind schedule, actual cost might be less than planned. It can be seen in my example (Figure 9.3) that this is not necessarily a valid assumption. The project in Figure 9.3 is behind in schedule and overrunning in cost. Indeed, the forward projection of cost for the project suggests a significant difference between the budget at completion (BAC) and the current estimate at completion. Figure 9.3 is a project in trouble! Of course, that information is extremely valuable for project management information, but not a useful contribution to measuring the software value of the organization or even the contribution to be expected from this project.

For organizations that use EVM, the SVMO would be a great place for EVM to be monitored and managed but, at face value, it is hard to see how value information could be gathered unless each WBS line item has a true or relative value metric associated with it. Based on value metrics described in previous chapters, the granularity of a typical WBS does not really lend itself to separate value metrics for each line item. A typical WBS line item is too detailed and small to sustain an independent value. For readers with an Agile rather than Waterfall background,
WBS line items are typically as small as or smaller than user stories. Clearly, trying to assign even relative independent value to individual line items in a WBS would be difficult, time-consuming, and ultimately counter-productive.

Is there any way that an SVMO can work with a requirement for EVM and use it to build good value data? Perhaps. The key is to remind ourselves that if we have built a process to maximize value flow, then we only need to establish relative value for decision making. We can assume (somewhat heroically) that the project (for which we are doing EVM based on a WBS) has already been prioritized against other projects for maximum value delivery. How, then, can we develop an algorithm or process for translating the relative value of the project into values for each WBS line item? I have set out one possible solution in Figure 9.4.

Clearly, the assignment of 100 “earned value points” to each project is arbitrary, but my thinking was to drive toward some sense of the percentage value of different parts of the WSJF. Additionally, I recommend striking a pragmatic compromise between the huge WBSs that exist in many government projects and time that could be wasted subdividing value when the subdivisions are increasingly unjustifiable and meaningless.

---

**Figure 9.4** Assigning relative value to line items in an EVM WBS.
The PMI quotes Alexandre Rodrigues,* and I’m happy to leave him with the last word in this section:

The EVM method was developed to measure scope accomplishment and cost and schedule performance. The term ‘earned value’ actually refers to ‘scope accomplished.’

However, the term ‘earned value’ is often interpreted by managers to mean ‘realized benefits’ or ‘produced economic value.’ In most cases for projects, the budget value of the scope accomplished does not equate to the value of business benefits achieved, nor economic value produced.

For that to be the case, EVM concepts would have to be applied at the program and portfolio level, to measure the performance of programs based on benefits realization and the performance of portfolios based on the creation of organizational values.

Summary

In this chapter, I have focused on why and how to set up an SVMO. We have considered the relative value to the organization going forward of a PMO and an SVMO. We have seen that EVM and software value management are different, but could be made to work together where EVM is a required process.

In the next chapter, I will switch gears and look at the value of software estimation—essentially planning value delivery. Arguably, software value delivery planning could be a function of the SVMO, but I have excluded it intentionally because I believe in the lean-Agile principle of “responding to change over following a plan.” I think there is a risk that the SVMO would revert to being a traditional PMO without adhering to that principle and I don’t want that to happen.
