

# Agile Program Management: Lessons Learned from the VeriSign Managed Security Services Team

Peter Hodgkins  
VeriSign, Inc.  
phodgkins@verisign.com

Luke Hohmann  
Enthiosys, Inc.  
lhohmann@enthiosys.com

## Abstract

*This report examines how the VeriSign Managed Security Services team (MSS) has successfully extended agile software development practices to deal with a complex, multi-product development organization. It showcases how VeriSign overcame several key challenges in the adoption of agile, such as the need to restructure the teams, and shows how the program management office provided organizational “glue” to hold things together to ensure a successful transition. It introduces several novel approaches to product-centric agile development, including the replacement of traditional Market Requirements Documents (MRDs) with a “Fast Track” business concept, and shows how cross-functional prioritization processes enable agility. It demonstrates how VeriSign collaborated with external consultants to ensure that best-practices were adopted and suggests ways that other companies can effectively manage the client-consultant dynamic. We conclude with an exploration of areas for further development, focusing on project level metrics for corporate planning.*

## 1. Project Background

The VeriSign Managed Security Services (MSS) team provides a set of integrated solutions that offer threat detection, analysis, and prioritized response to reduce security risks to reputation, operations, and compliance. Examples of such solutions include Firewall Management and Intrusion Detection and Prevention Services.

Comprising more than 500 people organized across 3 geographic locations, the MSS team interfaces with many other teams within VeriSign to leverage corporate assets and provide coordinated solutions for customers with complex security requirements. These solutions are organized into discrete offerings and combined into portfolios of solutions.

In Q2CY2006, the MSS team leadership decided to adopt a Scrum-influenced agile software development method. Project goals included improving quality, reducing time-to-market, and aligning organizational resources with customer demand.

The MSS team engaged the services of Enthiosys, Inc. for overview training and strategic planning and Digital Focus, Inc. for hands-on, project team coaching. Both Enthiosys and Digital Focus were already engaged at Verisign in other divisions, and the success of these prior engagements made the continued relationship with both vendors a natural choice. Both consulting organizations were further considered appropriate choices because of their commitment to enabling the VeriSign team to become self-sufficient in agile methods.

## 2. Why a PMO?

Most product management practices celebrate the role of the product manager as a person who is solely focused on the success of their product. Common phrases used to describe product managers include “The CEO of the product”, and someone who is “responsible for everything but owns nothing”. The former refers to the financial and market oriented criteria by which product managers are routinely evaluated. The latter refers to the fact that the traditional functional roles associated with a product, including engineering, QA, sales, and service don’t report to the product manager. Thus, effective product managers learn to manage through a combination of market-driven data, visionary leadership, and superior financial performance.

This structure works well when a company is offering a single product, or multiple but closely related products. It starts to break down when dealing with related product offerings from a single business unit. One example of a breakdown occurs when individual product managers are competing for limited, and often shared, resources, such as senior technical

leaders. Another is when a market deliverable must be coordinated among many project teams.

What is needed in such organizations is a higher level authority – one that is capable of dealing with the potentially competing interests of individual product teams. This is the role of the Program Management Office (PMO) – to assist organizations in managing the allocation of business unit resources across a set of projects in order to maximize returns while balancing risks. VeriSign had previously created a successful PMO, and it was considered a critical part of the MSS team.

### 3. The Agile Approach to Adopting Agile

In planning and strategy sessions with Enthiosys, VeriSign came to realize that most expressions of Scrum-centric agile development were based on the single product team/single product manager model. As such, these approaches were likely to break down when dealing with the complex MSS team structure. For example, the MSS team quickly realized that Scrum simply requires product managers to create a prioritized backlog. Little guidance is given about *how* to structure the backlog, or *when* to create a product release. This is not a criticism of Scrum, as Scrum is not intended to address the challenges of product management, which include numerous practices beyond software development (consider pricing, channel strategy, licensing, positioning, messaging, determination of value propositions, and so forth).

In addition to this challenge, no matter how the adoption of agile was managed, the reality of the large and complex structure of MSS was that different projects were in different phases of development. Some project teams were completing a release, making them good candidates for agile adoption. Others were in the middle of a release cycle, and while adopting agile was attractive, the risk to previously committed deliverables was too great. As is common with many large organizations, agile practices must be adopted in the context of ongoing work.

These needs motivated an agile approach to adopting agile. To get things started, VeriSign engaged Enthiosys to provide a high-level, two-day training class tailored to the needs of the MSS team and providing an overview of how the organization was going to adopt agile within various teams. Following this, specific teams were asked to adopt agile, with Digital Focus providing detailed project management coaching. While this approach takes longer than a “big bang – everyone adopts agile all at once”, it has the virtue of ensuring that each team’s learning’s and

experiences provide an increasingly broad foundation for subsequent teams. It also reduces a reliance on outside consultants, as internal teams can provide guidance to new teams.

The MSS team also realized that it wasn’t just the product managers and development team who needed to change – the PMO needed to change as well. We’ll now cover these changes in greater detail, starting with strategic business unit planning, business unit and product offering backlogs, and the restructuring of the organization to align agile deliverables to specific offerings.

### 4. The Agile PMO – Roadmaps

The agile literature celebrates several layers of planning, ranging from corporate strategy to the daily tasks undertaken within a given sprint. Product managers live in the middle of this world, and are charged with creating product plans that are congruent with larger corporate strategies while providing a sufficient time horizon for the product. In a complex business unit such as MSS, additional planning documents that deal with business unit goals are required.

To address this challenge, the MSS team created two kinds of roadmaps. The first were a series of product-level roadmaps created and managed by each product manager. The second was a single, business unit roadmap that consolidated key aspects of the product level roadmaps into an even high level, strategic offering. The regular format of the product-level roadmaps were based on the roadmapping pattern language described in [2] making the creation of the portfolio roadmap significantly easier.

#### Why Create Roadmaps When You Have Backlogs?

While the Scrum backlog is a powerful mechanism for managing the efforts of the development team, it is insufficient to address the strategic management needs of most software companies. Unlike the backlog, roadmaps are intended to identify and clarify the strategic intent of your product. Internally, well-crafted roadmaps serve many functions, including:

- Providing a useful filter to aide in backlog prioritization
- Communicating strategic intent
- Reducing the “person who shouts the loudest gets what they want”
- Enabling the technical team to communicate critical needs and/or opportunities to product managers

- Enabling product managers to facilitate the architectural evolution of the offerings

Roadmaps also serve several critical external functions, including:

- Providing customers with access to near-term commitments and long term points of view, enabling them to engage in appropriate planning processes
- Inviting customers to share their perspectives on the future, which further binds them to the company and the offering.

Given the benefit of roadmaps, we're continually surprised with the number of teams that don't create them. We suspect that roadmapping, like other forms of strategic thinking, is challenging in many agile-centric contexts because the longer timeframes of roadmaps are at odds with the short term horizons of agile methods. Moreover, roadmaps often ask leaders make commitments about uncertain futures. These challenges can be overcome, as once a base roadmap is in place it can be managed using agile-centric techniques of responding to change over following a plan.

The creation of these roadmaps was accomplished through two two-day meetings attended by all MSS product managers and senior technical architects and facilitated by Enthiosys over roughly 6 weeks of calendar time. The first meeting introduced the format and resulted in a draft roadmap. The intervening calendar time allowed product managers and technical architects to answer key questions and discuss critical issues before making binding commitments. In the second two-day meeting, the final roadmaps were created and published to the team. In general, the targeted time frame for these roadmaps was 6 quarters (18 months) into the future, with some product managers in more mature offerings creating longer roadmaps and others creating shorter roadmaps.

Several benefits were realized from this approach, some of which we'll discuss now and some in subsequent sections. Perhaps the most important benefit is that roadmaps provide a point of view about the future that is critical for successful agile development. Backlogs are great, and we need them, but you can't get a sense of where you're headed or why this destination is important just from looking at the backlog. A roadmap, on the other hand, provides exactly this, in a format that promotes communication, collaboration, and understanding on the markets being served, the key features and benefits being offered to

these markets, and the necessary technical infrastructure required to realize these benefits. VeriSign chose to share roadmaps through large poster prints hung in project rooms, an expression of the osmotic communication favored by agile practitioners.

The regular structure of specific product roadmaps enabled the PMO office to create a roadmap-of-roadmaps. This document serves the leaders of the business unit by providing a single point of view on everything from planned releases of individual products, to the mix of offerings being targeted to shared market segments. Critical market events and market rhythms, such as moves from competitors, or regularly attended conferences allow business unit leaders to allocate critical resources. As described later, the roadmaps also enabled further agility relative to new product planning.

## 5. The Agile PMO – Backlogs

A core tenant of agile is the creation of a prioritized backlog of business requirements. While this sounds easy, any product manager can tell you that creating and managing a good backlog is an ongoing challenge. This challenge is even more complex when there are multiple products, each with an individual, possibly related backlog.

In order to bring some order to the creation of backlogs, the PMO created a multi-product backlog process. First, business unit leaders and product managers were consulted to create a business unit backlog. The items on this backlog represent the key benefits and over-arching solutions that are required to serve a market segment, regardless of which specific product, or which set of products, will implement a given functionality. This backlog is prioritized at the business unit level.

Backlog items are then distributed to product managers, who integrate these business unit priorities with known product-level priorities. To illustrate, a new kind of security threat is probably going to be captured at the business unit level before being distributed to one or more product offerings for detailed execution. A bug fix for a particular product is managed by the product manager. As a result, product managers are responsible for managing their own backlog in the context of the PMO business unit backlog.

In the remainder of this section, we focus on the role of the PMO in the management of the business unit backlog. As you read the rest of this section, keep in mind that product managers work often involves translating market needs into product requirements,

while the PMO tends to work exclusively at the level of market needs.

#### **Backlog Items and Prioritization Attributes**

In our model, backlog items simply represent work that must be done. This work could be anything from implementing a user story, fixing a bug, or making a significant architectural enhancement. Most agile teams create a set of attributes to make prioritizing backlog items easier. The following represents attributes that we have used to help prioritize requirements. Note that not all attributes are used in any given project, and that the acquisition of data needed to support a given set of attributes (such as market research or financial modeling) can itself be a complex, not necessarily agile project whose time & cost that must be included.

- The results of various market research techniques, such as conjoint analysis, MaxDiff, Forced Rankings, or Kano analysis
- Various financial ranging from perceived cost, NPV, IRR, and so forth.
- The wishes of key internal stakeholders (sales, services, development, etc.)
- Corporate priorities such as time to market, compliance with regulatory agencies, or customer commitments to customers who account for more than 10% of gross revenue.
- Requirements source (e.g., a customer who provides a disproportionate amount of revenue is likely to have their requests ranked more highly than other customers)
- Affected systems.
- Development risk.
- Strategic relevance or similar weightings.

Every product manager should take the time to define the attributes that they will be using to guide them in their prioritization process. This does not mean, however, that a product manager is blindly led by this process, and they should be held accountable for the ultimate success of their product.

The first step the PMO office did was to tailor an Enthiosys provided backlog prioritization spreadsheet with attributes that made sense to the MSS business unit, focusing on aligning individual product initiatives with business unit strategy. Peter Hodgkins drove this effort, and worked with business unit leaders through several iterations of the spreadsheet to create a model that worked for the team. A well constructed prioritization process enables prioritization decisions by creating “separation” among distinct backlog items.

For example, one prioritization attribute is the degree to which a given feature has been contractually committed to a given customer, a common practice in enterprise software. For example, a given feature may be required to close a deal, may be required to realize channel partner revenue, or may have been purchased through non-recurring engineering (NRE). In the original spreadsheet, the cutoff for customer commitments was \$20K. Unfortunately, this number was too low, and more than 40 commitments were identified at this level. Additional levels of customer commitment were added to the spreadsheet at higher dollar amounts, which enabled the kind of separation that facilitates prioritization decisions.

The second step that Peter instituted was chairing a monthly, cross-functional team meeting of stakeholders to gain their feedback and realize consensus in the backlog. By directly incorporating stakeholders into a shared meeting, prioritization decisions can be made in the open, with different stakeholder interests clearly represented. One challenge in this approach is dealing with the feature treadmill, which happens when all stakeholders continually ask for new features without consideration for underlying architectural improvements. To address this problem, anthropomorphize the system so that it can represent itself as a stakeholder (*Anthropomorphism* is the attribution of uniquely human characteristics and qualities to nonhuman beings, inanimate objects, or natural or supernatural phenomena). This allows architectural and other technical improvements to be considered as first-class backlog items. A balanced release results when all stakeholders are represented.

#### **Treating Your System Like a Person**

The role of architecture in agile software development is pretty interesting. Some practitioners equate architectural planning to “Big Up Front Design” (BUFD) and minimize its importance. We believe that this is inappropriate, and instead believe that up-front architectural analysis and design is critically important to the long-term success of agile projects.

That said, every architecture, no matter how well-designed, must be maintained. This can be a challenge in agile teams who focus their attention on backlog items that are most associated with top-line revenue growth and/or customer demands. To combat this tendency, we advocate anthropomorphizing the system. This enables the system to literally “have a voice” in its own “care and feeding” and helps encourage agile teams to make architectural investments that are every bit as important to the long-term needs of the business as various features are to other customers.

Peter reported that this approach saved considerable time in the backlog effort and resulted in stickier decisions. Previously, reaching agreement on release deliverables took approximately six 2-3 hour meetings. Using this approach, the release prioritization process required one 2 hour meeting. This has since been further optimized to a single 2 hour monthly meeting that reviews and prioritizes new backlog items, ascertains the status of previously prioritized items, and confirms business unit plans.

## 6. Restructuring Teams to Gain Further Agility

As Peter continued to work with product managers in creating individual backlogs for each offering, Rob Scudiere, Dir of Engineering, took a fresh look at how his team was consuming backlog items. He found that his teams didn't match the backlogs. More specifically, one team was asked to address two different backlogs, causing significant confusion among the team and frustration among the various groups. To address this problem, Rob worked with other leaders to substantially restructure his organization, creating at least one dedicated team for each backlog.

The improvements were immediate and striking. Internal teams, who had previously felt that their needs were always secondary to external product offerings, found that these dedicated resources substantially improved their ability to plan future process changes. Moreover, these changes could be more directly correlated with external product releases, as previously monolithic teams were forced to communicate and inter-operate through more rigorously defined interfaces, reducing both architectural and organizational coupling.

Perhaps more importantly, the new structure enabled the level of resource sharing that an effective PMO office requires. Specifically, if one of the jobs of the PMO is to facilitate the allocation of resources, you have to have an organizational structure that is itself agile. By structuring a core team for every backlog, specific resources could be subtly shared among individual teams, better matching resource demands with capacity. To illustrate, suppose that one team needed to increase velocity to meet a critical deadline. Although Brooks Law [1] states that adding programmers to a late project makes it later, Abdel-Hamid and Madnick have demonstrated that you can cheat Brook's law provided certain conditions are met, such as programmers who are intimately familiar with the problem domain, tools, and fellow members of the

team. Thus, members of the larger MSS organization have been successfully shared among various discrete product teams without generating any confusion over individual project or shared priorities.

## 7. Reducing Unnecessary Waste – Integrating Lean

With prioritized backlogs and a new organizational structure in place, the leadership team had cause to take a fresh look at their development process as they considered the role of traditional Marketing Requirements Documents (MRDs). Specifically, one of the teams was in the process of adding some new functionality into the product, and they started to question the value of writing an MRD in the context of agile. The agile literature doesn't say much about MRDs, which can cause problems as MRDs are often associated with less-than-agile development processes.

There are many challenges with traditional MRDs on agile projects, the chief of which is the traditional MRD structure, which integrates business case, strategic intent, and market requirements into the same document, with no clear relationship between these concepts and roadmaps or backlogs. When coupled with the fact that many MRDs are completed with the intention of "filling out a template for a process checkpoint" instead of "providing the information needed to run the business", you find agile teams itching to through out MRDs for good reason.

Simply throwing out MRDs is not a viable option. Any project that is undertaken by an organization should have a business case suited towards it lifecycle stage and the culture of the organization. Some organizations, for example, require considerable rigor in business cases, while others are more comfortable with less detail. Regardless of whether it is a simple email manifesto or a multi-page analysis, business cases (to whom are you selling what, and why do you think you'll make money doing the same?) are critically important in every business, agile or otherwise.

The specific example that brought this issue to the fore was whether or not the MSS team should provide support to the Cisco Adaptive Security Appliance (ASA), Cisco's strategic replacement for its current PIX firewall and 4200 Series IDS offerings. The prior process employed by VeriSign required that an MRD be written, but this just didn't make sense to the whole product team, who knew that validating this decision through an MRD was simply unnecessary busy work.

To address this situation and see how their process could be further improved, the MSS product leadership

team convened a meeting and invited Enthiosys to further participate in the ongoing design of their product development process. The result of this meeting was the creation of a Dynamic Track process, in which new items for consideration are slotted into one of three tracks:

- **Business Case Track:** This track is used when a newly requested functionality needs a more heavy up front definition (such as defining the target market or investigating functionality requirements). Where there are multiple layers of requirements of varying complexity, in this track a full Business Requirement (MRD) is written and even a full FRS. This then moves into user stories / backlog items and release / iteration planning.
- **Fast Track:** This track is used when the newly requested functionality can be shown to be congruent with previously created roadmaps and/or existing product requirements. It moves directly into user stories and follows normal agile processes.
- **Inquiry Track:** This track is used when there is no straightforward decision as to which of the other tracks the requirement should fall into. It is used to investigate the newly requested functionality further and determine the scope of the work needed to allow development to be defined.

This process has enabled greater agility at the business unit planning level and produced more thorough MRDs, as MRDs are created to serve their intended purpose, and not created to simply “check off” a process.

Using this new process, the development team was able to easily place the Cisco ASA into the Fast Track, saving considerable time and money.

## 8. Next Steps – PMO Metrics

While we are proud of the steps that we have taken and the improvements we’ve realized, we have additional goals for our business unit. One of these goals includes the creation of business unit operational metrics that will enable us to better understand overall investment choices and quality improvements. This is challenging, because the points-estimating process favored by our teams does not result in metrics that can be transferred between teams. With hard work, we hope to share our successes in establishing these metrics with the agile community in a future paper.

## 9. Conclusions

The adoption of agile development practices within complex multi-product portfolios requires the full integration of the Program Management Office (PMO). At the same time, the portfolio management team must adjust its approach to project governance to allow for the benefits of agile. Our approach, while evolving to meet the ongoing needs of the VeriSign MSS team, provides a suitable foundation that can be tailored to meet the specific needs of other organizations.

## 10. References

- [1] Brooks, F. *The Mythical Man-Month: Essays on Software Engineering, 20<sup>th</sup> Anniversary Edition*, Addison-Wesley Professional, 1995.
- [2] Hohmann, L. *Beyond Software Architecture: Creating and Sustaining Winning Solutions*, Addison-Wesley, 2002.