

1.5

SOFTWARE ASSURANCE MATURITY MODEL

HOW TO GUIDE

A GUIDE TO BUILDING SECURITY INTO SOFTWARE DEVELOPMENT



Project leaders:
Sebastien Deleersnyder, Bart De Win
& Brian Glas

Creative Commons (CC) Attribution
Free Version at: <https://www.owasp.org>



This is an OWASP Project

OWASP is an international organization and the OWASP Foundation supports OWASP efforts around the world. OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security. We advocate approaching application security as a people, process, and technology problem because the most effective approaches to application security include improvements in all of these areas. We can be found at <https://www.owasp.org>.

License

This work is licensed under the Creative Commons Attribution-Share Alike 4.0 License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-sa/4.0/> or send an email to info@creativecommons.org. or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042.

/ CONTENTS

- 02 Executive Summary
- 03 Software Development
- 04 SAMP Evolution
- 06 **Applying The Model**
- 07 Using The Maturity Levels
- 08 Related Levels
- 08 Conducting Assessments
- 09 Creating Scorecards
- 11 Building Assurance Programs
- 14 Independent Software Vendor
- 16 Online Service Provider
- 18 Financial Services Organization
- 20 Government Organization
- 22 **Case Study**
- 23 Virtualware
- 26 Virtualware - Phase 1
- 29 Virtualware - Phase 2
- 32 Virtualware - Phase 3
- 35 Virtualware - Phase 4
- 38 Virtualware - Ongoing
- 40 Acknowledgements And Sponsors
- 40 Contributors & Reviewers
- 41 Sponsors

/ EXECUTIVE SUMMARY

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. The resources provided by SAMM will aid in:

- Evaluating an organization's existing software security practices.
- Building a balanced software security assurance program in well-defined iterations.
- Demonstrating concrete improvements to a security assurance program.
- Defining and measuring security-related activities throughout an organization.

SAMM was defined with flexibility in mind such that it can be utilized by small, medium, and large organizations using any style of development.

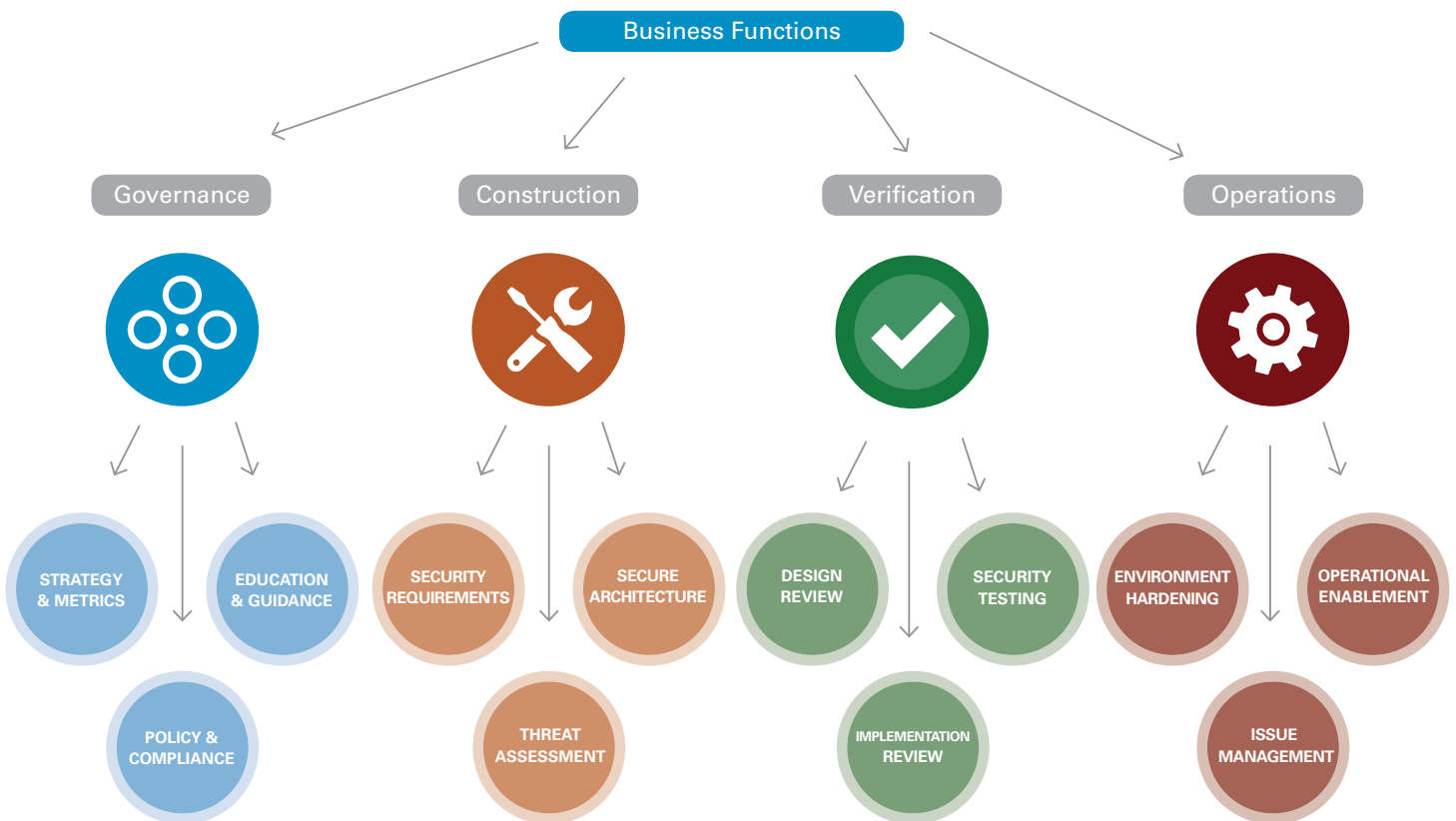
As an open project, SAMM content shall always remain vendor-neutral and freely available for all.

Besides the How-To Guide and the Core Model document, several other tools and documents have been made available during the last several years:

- The new Quick-Start Guide walks you through the core steps to execute your SAMM-based secure software practice.
- The updated SAMM Tool Box can be used to perform SAMM assessments and create SAMM roadmaps.
- Lots of OWASP resources are linked from the SAMM project page on the OWASP website. You can use these to implement SAMM roadmaps.
- With the SAMM Benchmark data, you can compare your maturity and progress with other organizations and teams.

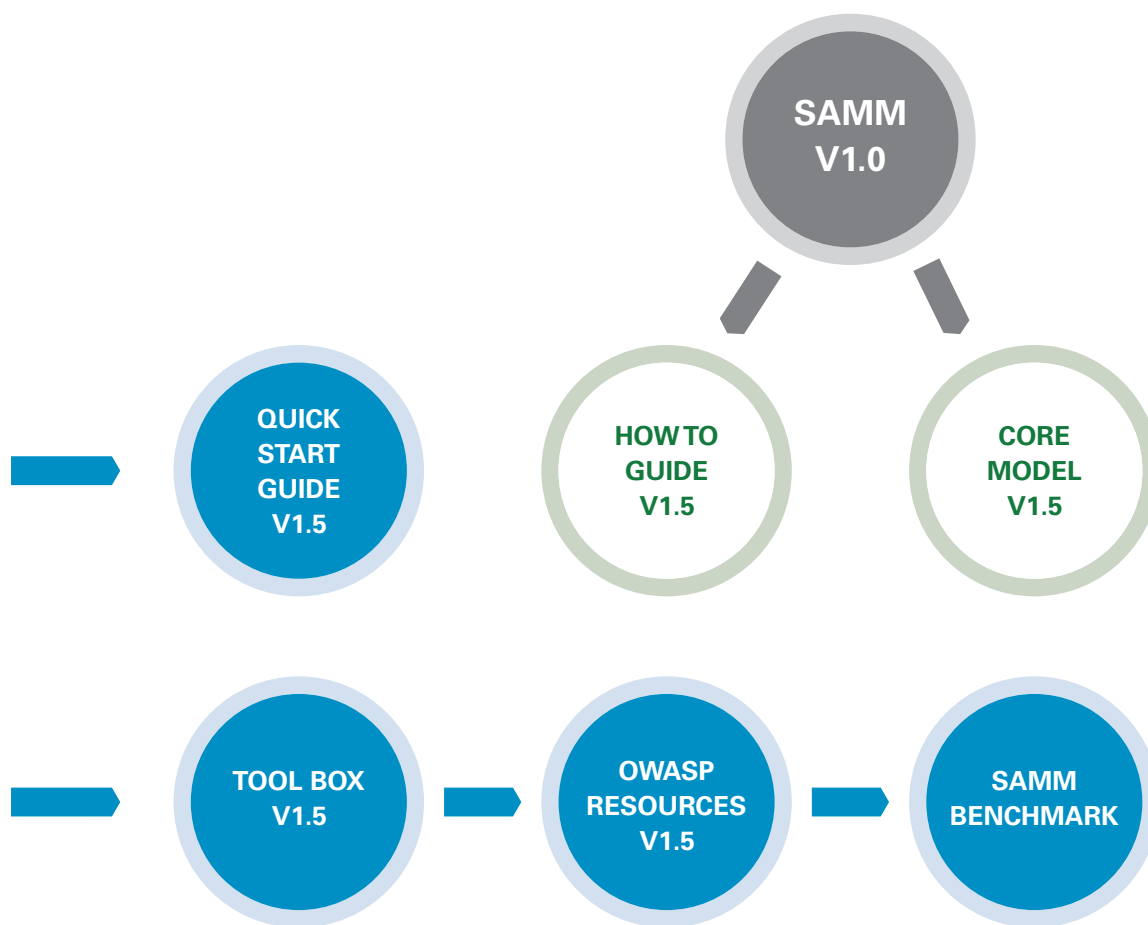
SOFTWARE DEVELOPMENT

- SAMP Overview -



/ SAMM EVOLUTION

SAMM v1.0 was originally developed, designed, and written by Pravir Chandra. As part of the v1.1 release, this How-To Guide has split off the SAMM implementation guidance from the SAMM Core Model document. SAMM v1.5 documentation continues with the same format as v1.1.



● SPLIT & IMPROVE

● NEW in v1.1



/ APPLYING THE MODEL

Putting it all to work

This section covers several important and useful applications of SAMM. Given the core design of the model itself, an organization can use SAMM as a benchmark to measure its security assurance program and create a scorecard. Using scorecards, an organization can demonstrate iterative improvement through an assurance program. And most importantly, an organization can use SAMM road-map templates to guide the build-out or improvement of a security assurance initiative.

/ USING THE MATURITY LEVELS

Each of the 12 Security Practices have three maturity levels. Each level has several components that specify the critical factors for understanding and achieving the stated level. Beyond that, these prescriptive details make it possible to use the definitions of the security practices, even outside the context of using SAMM to build a software assurance program.

Objective

The objective is a general statement that captures the assurance goal of attaining the associated level. As the levels increase for a given practice, the objectives characterize more sophisticated goals in terms of building assurance for software development, deployment, and operations.

Activities

The activities are core requisites for attaining the level. Some are meant to be performed organization-wide, and some correspond to actions for individual project teams. In either case, the activities capture the core security function, and organizations are free to determine how they fulfill the activities.

Results

The results characterize capabilities and deliverables obtained by achieving the given level. In some cases, these are specified concretely, and in others, a more qualitative statement is made about increased capability.

Success Metrics

The success metrics specify example measurements that can be used to check if an organization is performing at the given level. Recommended data sources and thresholds are provided, and data collection and management is left to the choice of each organization.

/ PERSONNEL

Developers:

Individuals performing detailed design and implementation of the software.

Architects:

Individuals performing high-level design work and large scale system engineering.

Managers:

Individuals performing day-to-day management of development staff.

QA Testers:

Individuals performing quality assurance testing and pre-release verification of software.

Security:

Individuals with technical security knowledge related to software being produced.

Business Owners:

Individuals performing key decision making on software and its business requirements.

Support Operations:

Individuals performing customer support or direct technical operations support.

RELATED LEVELS

The related levels are references to levels within other practices that have some potential overlaps depending upon the organization's structure and progress in building an assurance program. Functionally, these indicate synergies or optimizations in activity implementation if the related level is also a goal or already in place.

CONDUCTING ASSESSMENTS

By measuring an organization against the defined security practices, an overall picture of built-in security assurance activities is created. This type of assessment is useful for understanding the breadth of security activities currently in place at an organization. Further, it enables that organization to then utilize SAMM to create a future roadmap for iterative improvement.

An important first step of the assessment is to define the scope of the assessment: An assessment can be done for a complete organization, for selected business units, or even at development team level. This scope must be agreed upon with the involved key stakeholders.

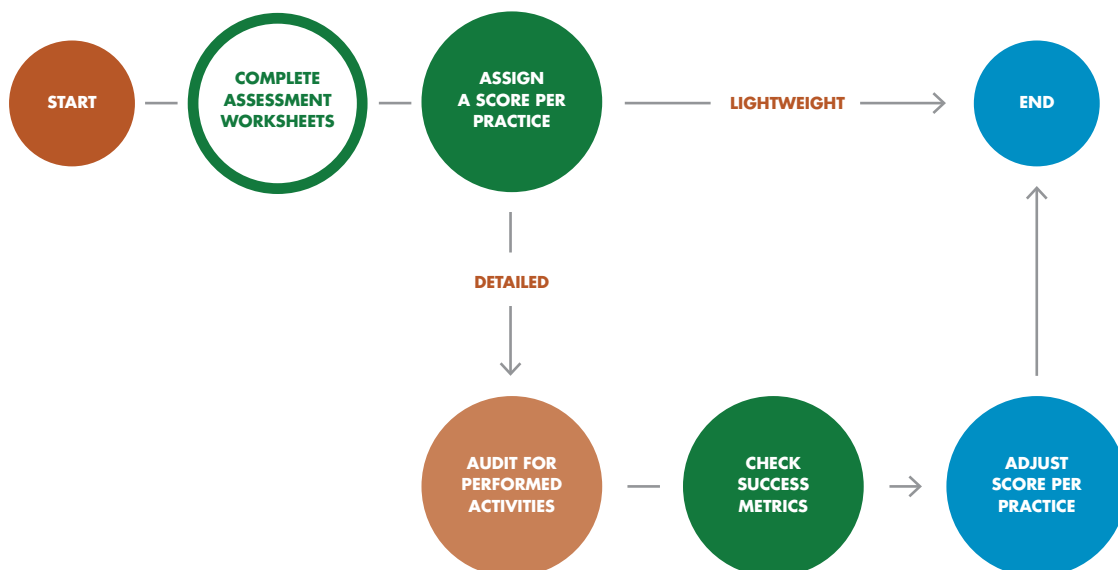
The process of conducting an assessment is simply evaluating an organization to determine the maturity level at which it is performing. The extent to which an organization's performance is checked will usually vary according to the drivers behind the assessment, but in general, there are two recommended styles:

Lightweight:

The assessment worksheets for each practice are evaluated, and scores are assigned based on answers. This type of assessment is usually sufficient for an organization that is trying to map their existing assurance program into SAMM to get a quick picture of where they stand.

Detailed:

After completion of the assessment worksheets, additional audit work is performed to check the organization to ensure the activities prescribed by each practice are in place. Additionally, since each practice also specifies success metrics, that data should be collected to ensure that the organization is performing as expected.



Scoring an organization using the assessment worksheets is straightforward. After answering the questions, evaluate the answer column to determine the score. It is recommended to use the toolbox spreadsheet or other application to assist in the calculation of the score.

Assurance programs might not always consist of activities that neatly fall on a boundary between maturity levels, e.g. an organization that assesses to a Level 1 for a given practice might also have additional activities in place, but not such that Level 2 is completed.

The scoring model in v1.5 provides more granularity to the scoring in an assessment. An organization will get credit for the different levels of work they have done in a practice. The scoring is fractional to two decimal places for each practice and a single decimal for an answer. Questions have also been changed from Yes/No to four options that represent different levels of coverage or maturity. Anyone who has filled out a SAMM assessment has had a discussion on whether to mark an answer yes or no, when it is honestly something in between.

The toolbox spreadsheet has context aware answers for each of the questions in the assessment. The formulas in the toolbox will average the answers to calculate the score for each practice, a roll up average for each business function, and an overall score. The toolbox also has scorecard graphics that help represent the current score, and can help show improvements to the program as the answers to the questions change. There are worksheets in the core document that align with the toolbox spreadsheet.

You can find the assessment worksheets in the SAMM Core Model document as of page 18. In v1.5 of SAMM, a separate SAMM Toolbox is made available to assist with scoring assessments. You can download the SAMM Toolbox from the SAMM page on the OWASP website.



CREATING SCORECARDS

Based on the scores assigned to each security practice, an organization can create a scorecard to capture those values. Functionally, a scorecard can be the simple set of 12 scores for a particular time. However, selecting a time interval over which to generate a scorecard facilitates understanding of overall changes in the assurance program during the time frame.

Using interval scorecards is encouraged for several situations:

Gap analysis

Capturing scores from detailed assessments versus expected performance levels.

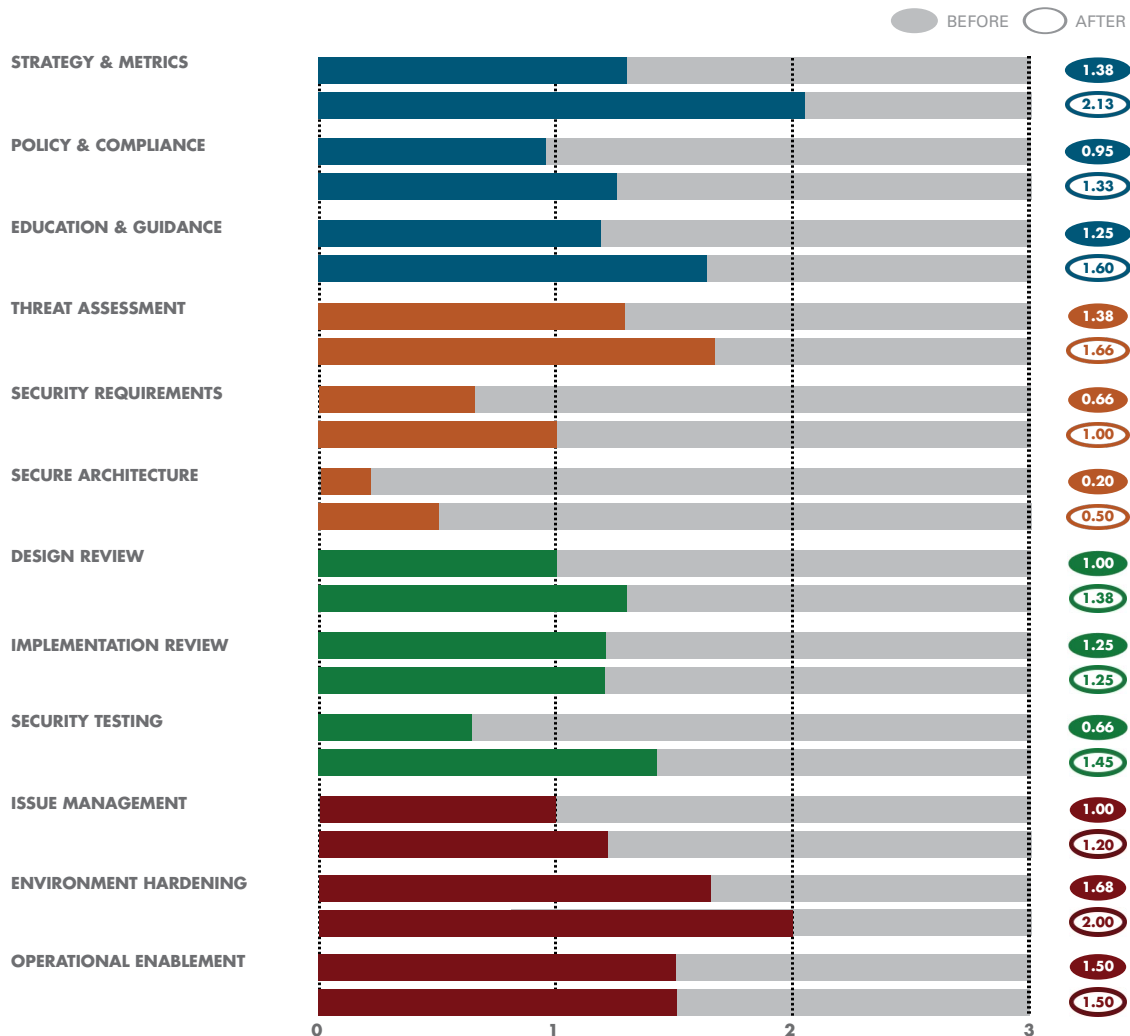
Demonstrating improvement

Capturing scores from before and after an iteration of assurance program build-out.

Ongoing measurement

Capturing scores over consistent time frames for an assurance program that is already in place.

The figure below shows an example scorecard for how an organization’s assurance program changed over the course of one year. If that organization had also saved the data about where they were planning on being at the end of the year, that would be another interesting data set to plot, since it would help show the extent to which the plans had to change over the year.



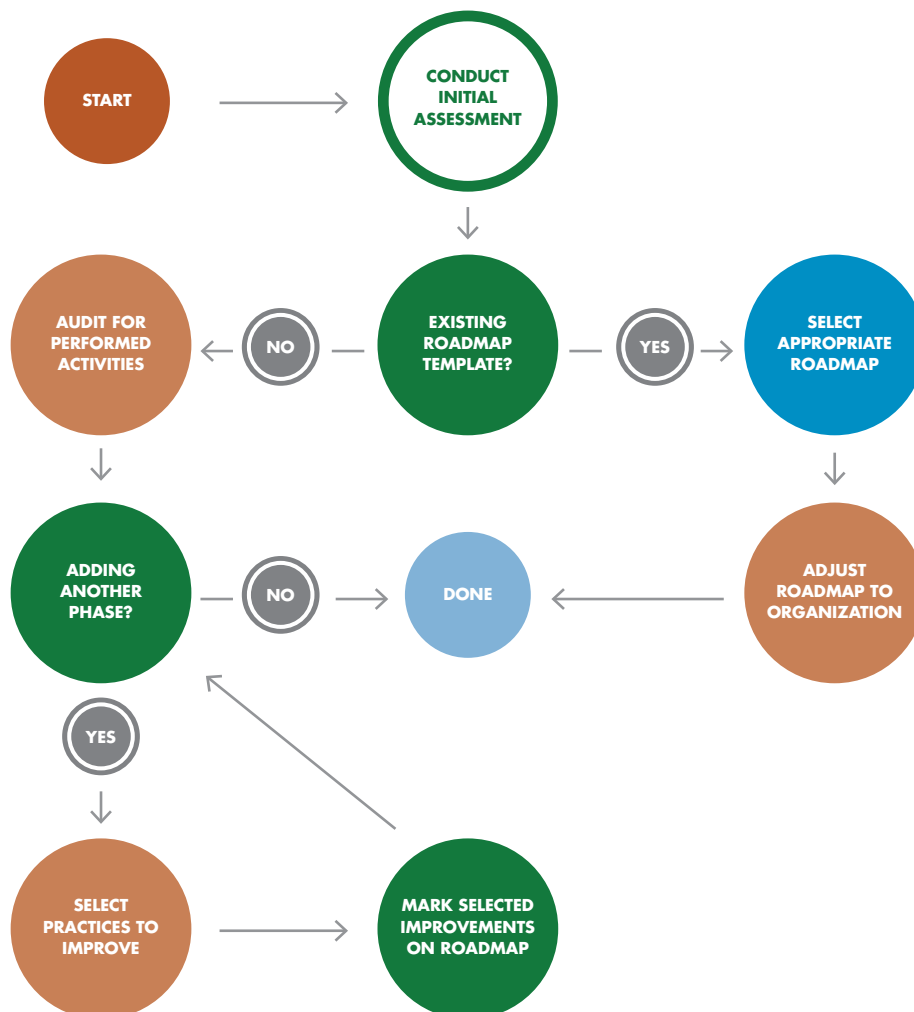
/ BUILDING ASSURANCE PROGRAMS

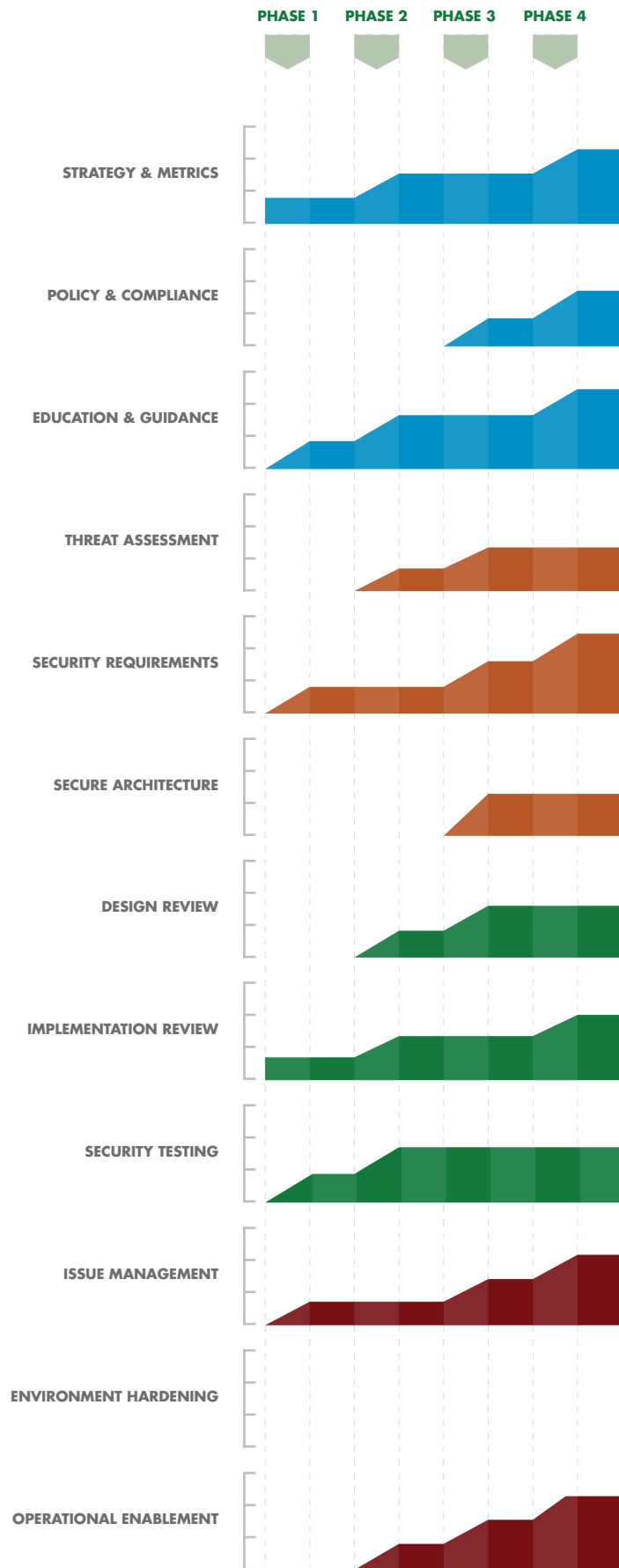
One of the main uses of SAMM is to help organizations build software security assurance programs. That process is straightforward, and generally begins with an assessment if the organization is already performing some security assurance activities.

Several roadmap templates for common types of organizations are provided. Thus, many organizations can choose an appropriate match and then tailor the roadmap template to their needs. For other types of organizations, it may be necessary to build a custom roadmap.

Roadmaps consist of phases (the vertical bars) in which several practices are each improved by one level. Therefore, building a roadmap entails selection of which practices to improve in each planned phase. Organizations are free to plan into the future as far as they wish, but are encouraged to iterate based on business drivers and organization-specific information to ensure the assurance goals are commensurate with their business goals and risk tolerance.

After a roadmap is established, the build-out of an assurance program is simple. An organization begins an improvement phases and works to achieve the stated levels by performing the prescribed activities. At the end of the phase, the roadmap should be adjusted based on what was actually accomplished, and then the next phase can begin.





/ INDEPENDENT SOFTWARE VENDOR

ROADMAP TEMPLATE

Rationale

An independent software vendor involves the core business function of building and selling software components and applications.

Initial drivers to limit common vulnerabilities affecting customers and users leads to early concentration on implementation review and security testing activities.

Shifting toward more proactive prevention of security errors in product specification, an organization adds activities for security requirements over time.

Also, to minimize the impact from any discovered security issues, the organization ramps up issue management activities over time.

As the organization matures, knowledge transfer activities from operational enablement are added to better inform customers and users about secure operation of the software.

Additional Considerations

Outsourced Development

For organizations using external development resources, restrictions on code access typically lead to prioritization of security requirements activities instead of implementation review activities. Additionally, advancing threat assessment in earlier phases would allow the organization to better clarify security needs to the outsourced developers. Since expertise on software configuration will generally be strongest within the outsourced group, contracts should be constructed to account for the activities related to operational enablement.

Internet-Connected Applications

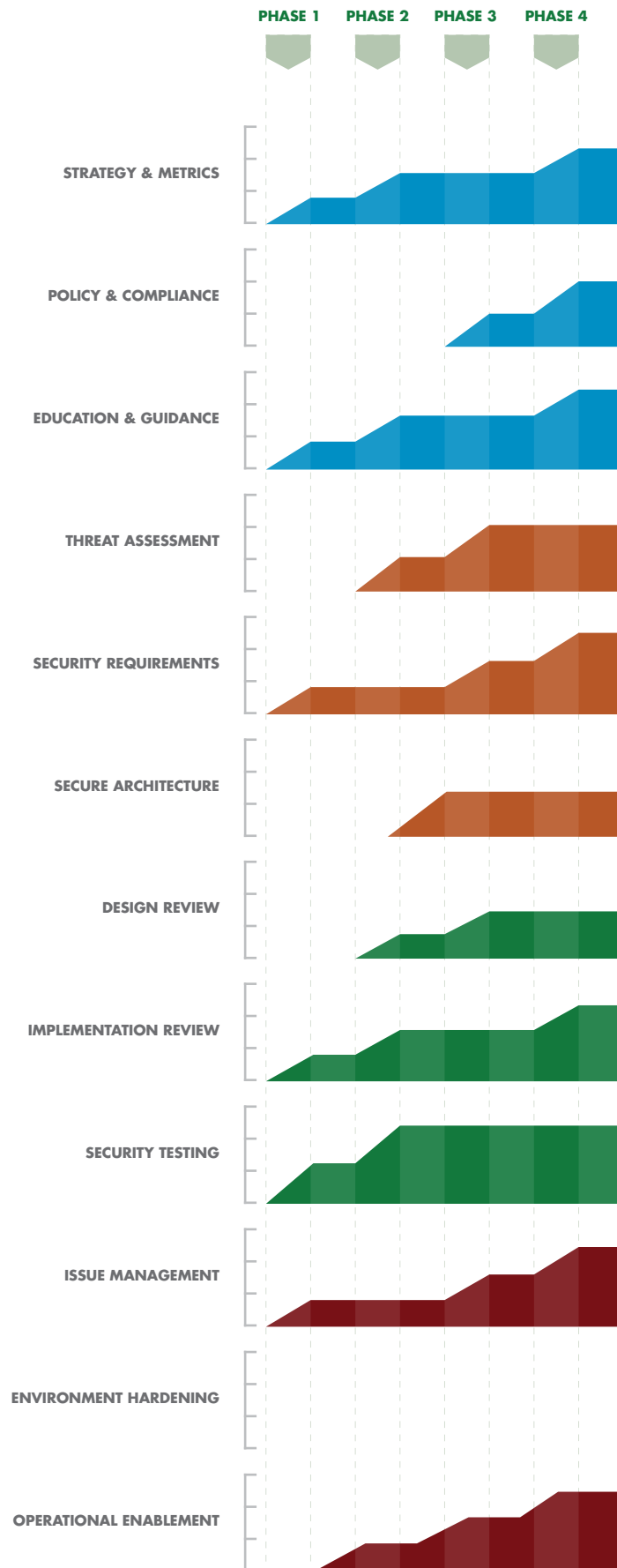
Organizations building applications that use online resources have additional risks from the core Internet-facing infrastructure that hosts Internet-facing systems. To account for this risk, organizations should add activities from environment hardening to their roadmaps.

Drivers and Embedded Development

For organizations building low-level drivers or software for embedded systems, security vulnerabilities in software design can be more damaging and costly to repair. Therefore, roadmaps should be modified to emphasize secure architecture and design review activities in earlier phases.

Organizations Grown by Acquisition

In an organization grown by acquisition, there can often be several project teams following different development models with varying degrees of security-related activities incorporated. An organization such as this may require a separate roadmap for each division or project team to account for varying starting points, as well as project-specific concerns if a variety of software types are being developed.



/ ONLINE SERVICE PROVIDER

ROADMAP TEMPLATE

Rationale

An online services provider involves the core business function of building web applications and other network-accessible interfaces.

Initial drivers to validate the overall soundness of design without stifling innovation may lead to an early concentration on design review and security testing activities.

Since critical systems will be network-facing, environment hardening activities are also added early and ramped over time to account for risks from the hosted environment.

Though it can vary based on the core business of the organizations, policy and compliance activities should be started early and then advanced according to the criticality of external compliance drivers.

As the organization matures, activities from threat assessment, security requirements, and secure architecture are slowly added to help bolster proactive security after some baseline expectations for security have been established.

Additional Considerations

Outsourced Development

For organizations using external development resources, restrictions on code access typically lead to prioritization of security requirements activities instead of implementation review activities. Additionally, advancing threat assessment in earlier phases would allow the organization to better clarify security needs to the outsourced developers. Since expertise on software configuration will generally be strongest within the outsourced group, contracts should be constructed to account for the activities related to operational enablement.

Online Payment Processing

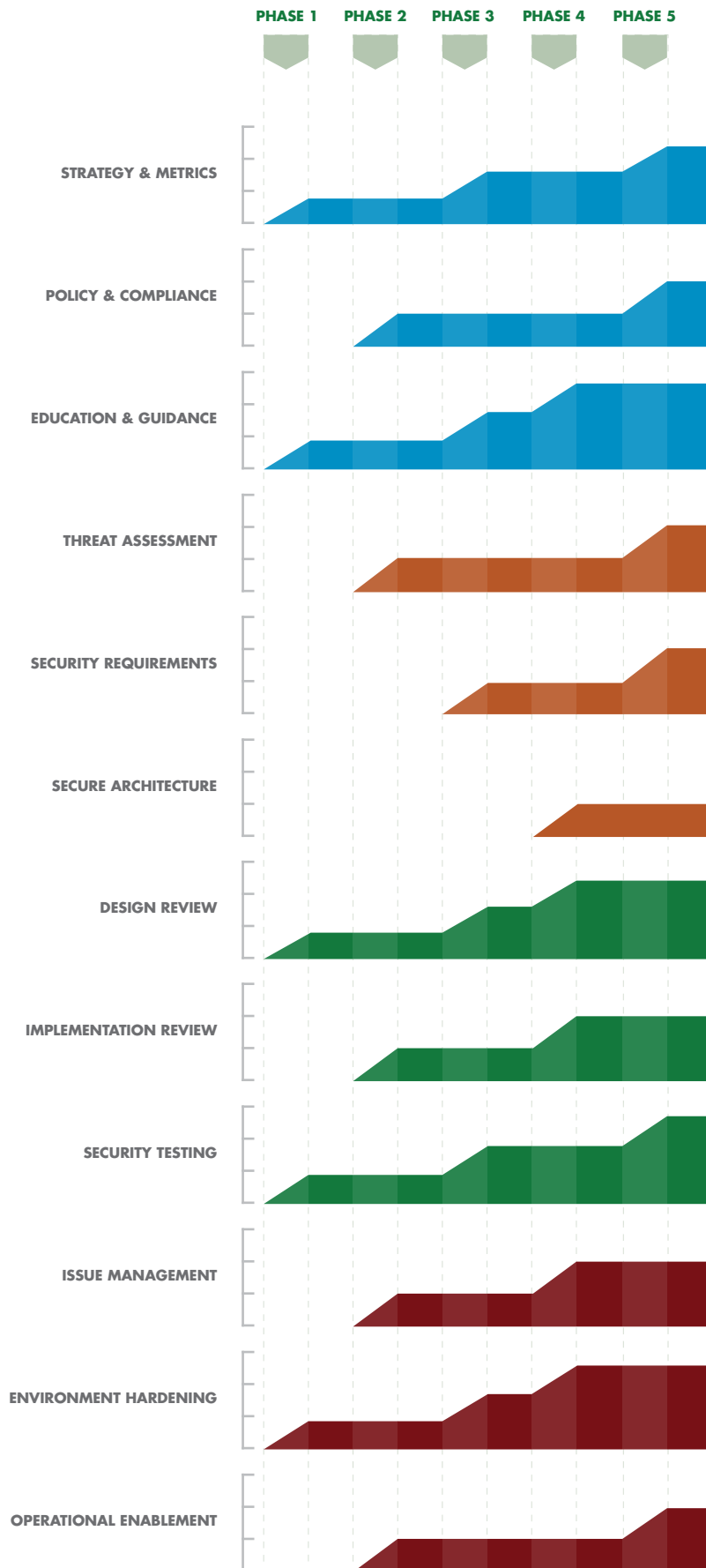
Organizations required to be in compliance with the Payment Card Industry Data Security Standard (PCI-DSS), or other online payment standards, should place activities from policy and compliance in earlier phases of the roadmap. This allows the organization to opportunistically establish activities that ensure compliance, and enables the future roadmap to be tailored accordingly.

Web Services Platforms

For organizations building web services platforms, design errors can carry additional risks and be more costly to mitigate. Therefore, activities from threat assessment, security requirements, and secure architecture should be placed in earlier phases of the roadmap.

Organizations Grown by Acquisition

In an organization grown by acquisition, there can often be several project teams following different development models with varying degrees of incorporated security-related activities. An organization such as this may require a separate roadmap for each division or project team to account for varying starting points, as well as project-specific concerns if a variety of software types are being developed.



/ FINANCIAL SERVICES ORGANIZATION

ROADMAP TEMPLATE

Rationale

A financial services organization involves the core business function of building systems to support financial transactions and processing. In general, this implies a greater concentration of internal and back-end systems that interface with disparate external data providers.

Initially, effort is focused on improving the practices related to governance, since these are critical services that set the baseline for the assurance program and help meet compliance requirements for the organization.

Since building secure and reliable software pro-actively is an overall goal, practices within construction are started early on and ramped up sharply as the program matures.

Verification activities are also ramped up smoothly over the course of the roadmap to handle legacy systems without creating unrealistic expectations. Additionally, this helps ensure enough cycles are spent building out more proactive practices.

Since a financial services organization often operates the software they build, focus is given to the practices within operations during the middle of the roadmap, after some initial governance is in place, but before heavy focus is given to the proactive construction practices.

Additional Considerations

Outsourced Development:

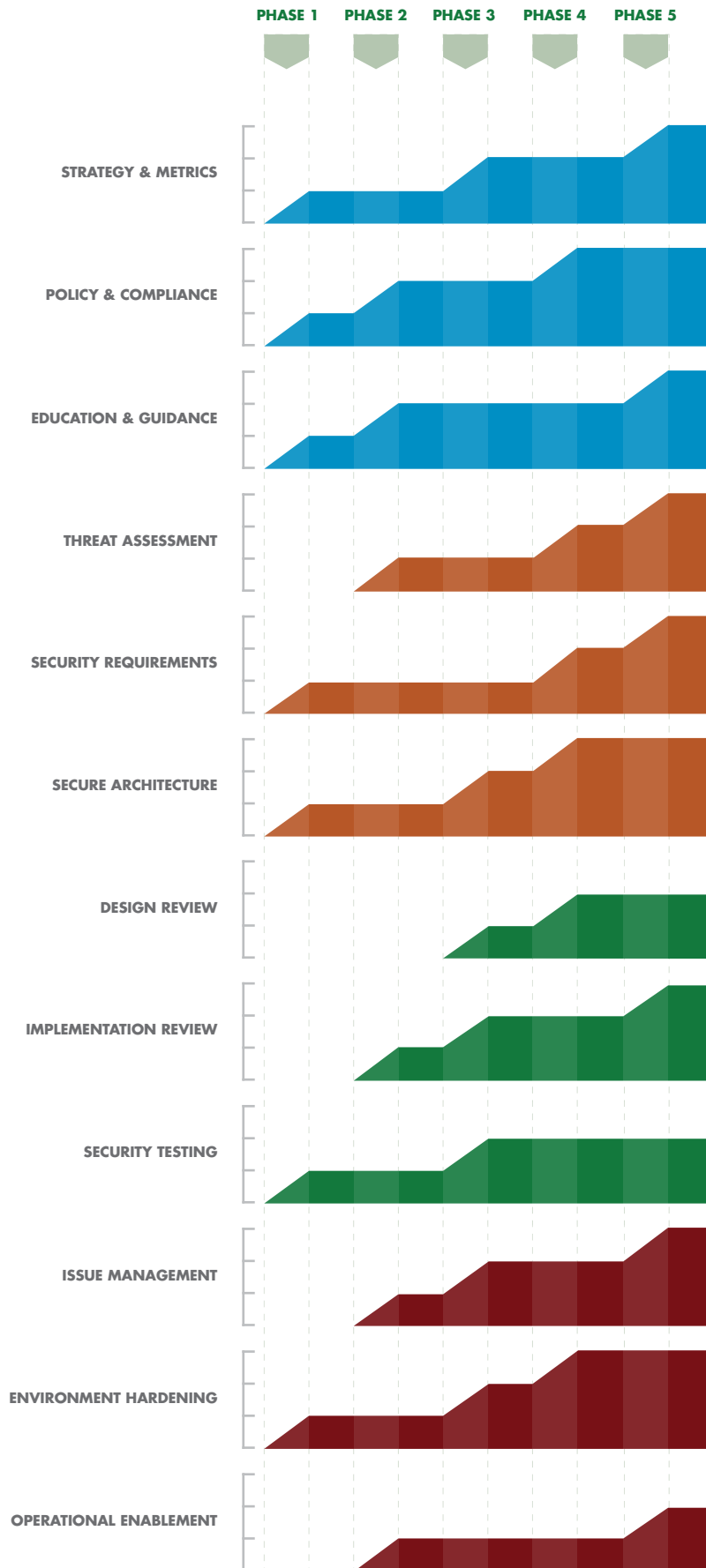
For organizations using external development resources, restrictions on code access typically leads to prioritization of security requirements activities instead of implementation review activities. Additionally, advancing threat assessment in earlier phases would allow the organization to better clarify security needs to the outsourced developers. Since expertise on software configuration will generally be strongest within the outsourced group, contracts should be constructed to account for the activities related to operational enablement.

Web Services Platforms:

For organizations building web services platforms, design errors can carry additional risks and be more costly to mitigate. Therefore, activities from threat assessment, security requirements, and secure architecture should be placed in earlier phases of the roadmap.

Organizations Grown by Acquisition:

In an organization grown by acquisition, there can often be several project teams following different development models with varying degrees of incorporated security-related activities. An organization such as this may require a separate roadmap for each division or project team to account for varying starting points, as well as project-specific concerns if a variety of software types are being developed.



/ GOVERNMENT ORGANIZATION

ROADMAP TEMPLATE

Rationale

A government organization involves the core business function of being a state-affiliated organization that builds software to support public sector projects.

Initially, governance practices are established, generally to get an idea of the overall compliance burden for the organization in context of the concrete roadmap for improvement.

Because of risks of public exposure and the quantity of legacy code generally in place, early emphasis is given to security testing within the verification practices, and later the more involved implementation review or design review practices are developed.

Similar emphasis is placed on the construction and operations practices. This helps establish the organization's management of vulnerabilities, and moves toward bolstering the security posture of the operating environment. At the same time, proactive security activities under construction are built up to help prevent new issues in software under development.

Additional Considerations

Outsourced Development:

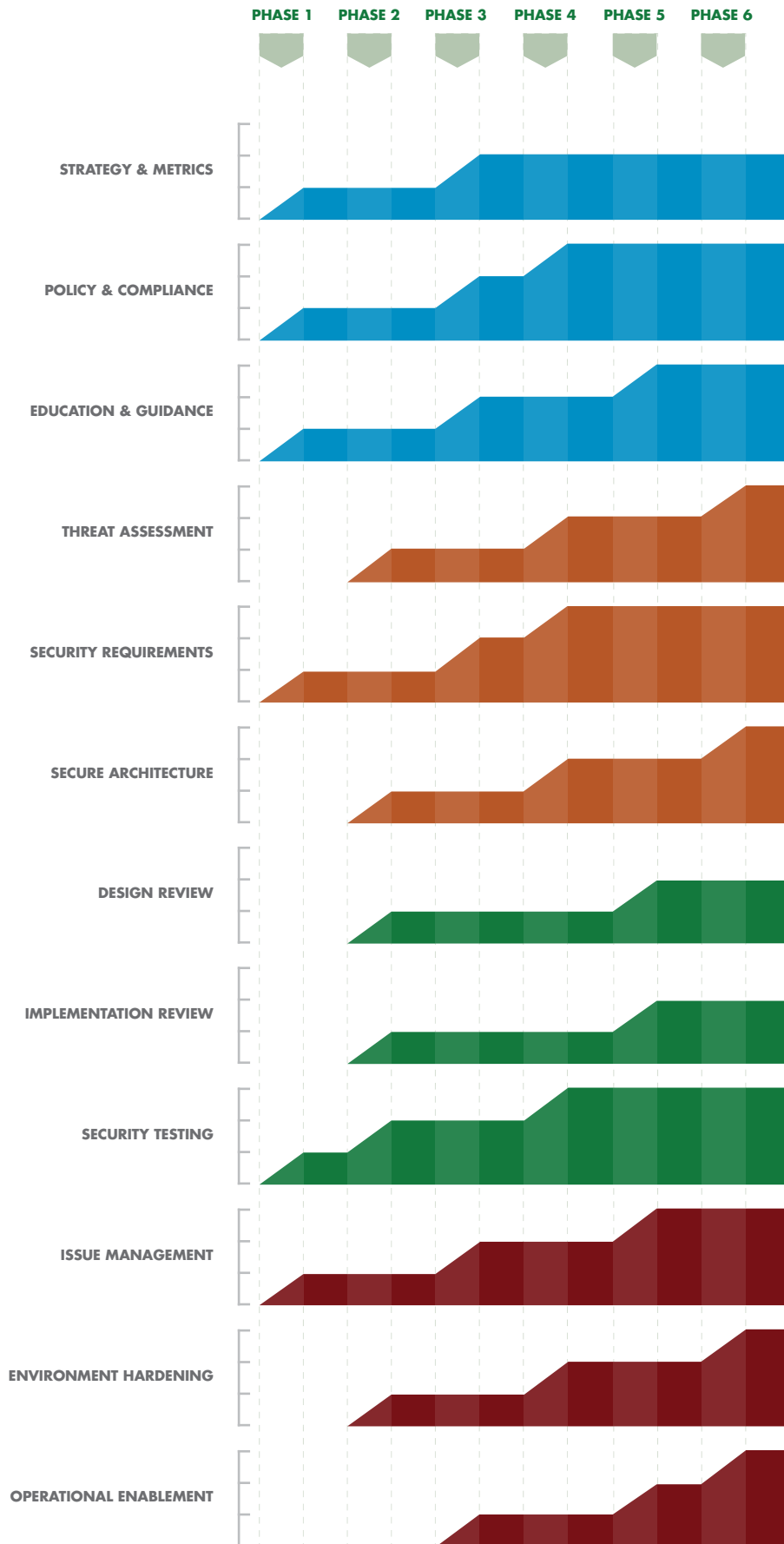
For organizations using external development resources, restrictions on code access typically leads to prioritization of security requirements activities instead of implementation review activities. Additionally, advancing threat assessment in earlier phases would allow the organization to better clarify security needs to the outsourced developers. Since expertise on software configuration will generally be strongest within the outsourced group, contracts should be constructed to account for the activities related to operational enablement.

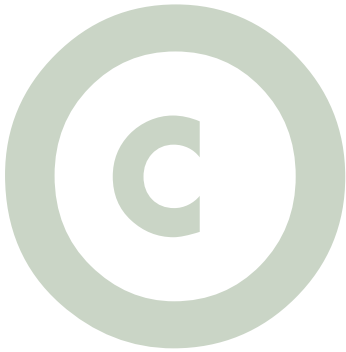
Web Services Platforms:

For organizations building web services platforms, design errors can carry additional risks and be more costly to mitigate. Therefore, activities from threat assessment, security requirements, and secure architecture should be placed in earlier phases of the roadmap.

Regulatory Compliance:

For organizations under heavy regulations that affect business processes, the build-out of the policy and compliance practice should be adjusted to accommodate external drivers. Likewise, organizations under a lighter compliance load should take the opportunity to push back build-out of that practice in favor of others.





/ CASE STUDY

A walkthrough of an example scenario

This section features a scenario in which the application of SAMM is explained in the context of a specific business case. Using the roadmap templates as a guide, the case study tells the story of how an organization might adapt best practices and take into account organization-specific risks when building a security assurance program.

Note: This case study was originally written for v1.0 so the scoring intervals are in whole numbers and may not precisely align with the v1.5 scoring.

/ VIRTUALWARE

CASE STUDY: MEDIUM-SIZED, INDEPENDENT SOFTWARE VENDOR

Business Profile

VirtualWare is a leader in the market of providing integrated virtualized application platforms, to help organizations consolidate their application interfaces into a single environment. Their technology is provided as a server application and desktop client built for multiple environments, including Microsoft, Apple, and Linux platforms.

The organization is of medium size (200-1000 employees), and has a global presence around the world with branch offices in most major countries.

Organization

VirtualWare has been developing their core software platform for over eight years. During this time, they have had limited risk from common web vulnerabilities due to minimal usage of web interfaces. Most of the VirtualWare platforms are run through either a server-based systems or thick clients running on the desktop.

Recently, VirtualWare started a number of new project streams, which deliver their client and server interfaces via web technology. Knowing the extent of common attacks seen over the web, this has driven the organization to review their software security strategy and to ensure that it adequately addresses possible threats towards their organization going forward.

Previously, the organization has undertaken basic reviews of the application code, but has been more focused on performance and functionality rather than security. VirtualWare developers have been using a number of code quality analysis tools to identify bugs and address them within the code.

With this in mind, the upper management team has set a strategic objective to review the current status of the security of their applications and to determine the best method of identifying, removing, and preventing vulnerabilities in them.

Environment

VirtualWare develops their virtualization technology on a mixture of Java, C++, and .NET technology. Their core application virtualization technology has been written in C++ and has had a number of reviews for bugs and security, but currently no formal processes exists for identifying and fixing known or unknown security bugs.

VirtualWare has chosen to support their web technology on Java, although the back-end systems are built using Microsoft and C++ technologies. The development team that is focused on the new web interfaces is primarily composed of Java developers.

VirtualWare employs over 300 developers, with staff broken up into teams based on the projects that they work on. There are 12 teams, with around 20–40 developers per team. Within each team there is minimal experience with software security, and although senior developers perform basic assessments of their code, security is not considered a critical goal within the organization.

Each team within VirtualWare adopts a different development model. Currently, the two primary methodologies used are Agile SCRUM and iterative Waterfall style. There is minimal to no guidance from the IT department or project architects on software security.

Key Challenges

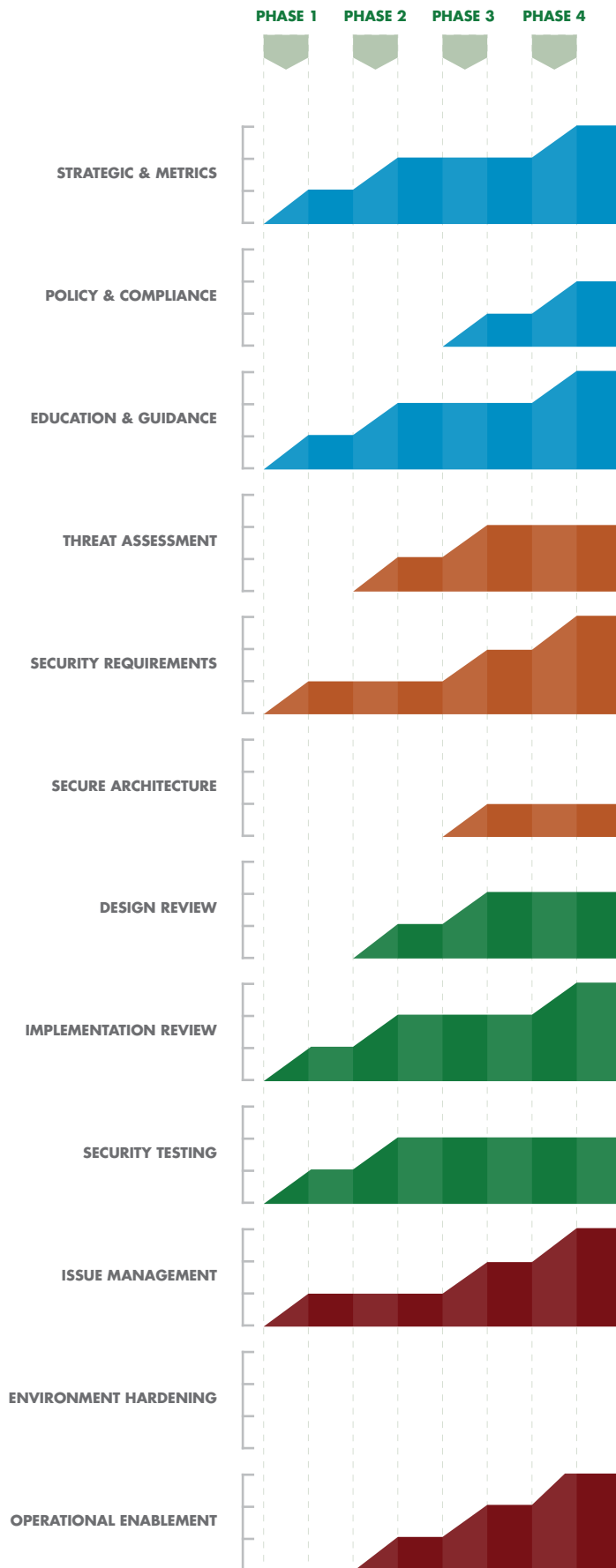
- Rapid release of application features to ensure they maintain their competitive edge over rivals.
- Limited experience with software security concepts — currently minimal effort is associated with security related tasks.
- Developers leave the organization and are replaced with less experienced developers.
- Multiple technologies used within applications, with legacy applications that have not been updated since originally built.
- No understanding of existing security posture or risks facing the organization.

VirtualWare wanted to focus on ensuring that their new web applications would be delivered securely to their customers. Therefore, the initial focus was on education and awareness for their development teams, as well as providing some base technical guidance on secure coding and testing standards.

The organization previously had received bug requests and security vulnerabilities through their support@virtualware.net address. However, as this was a general support address, existing requests were not always filtered down to the appropriate teams within the organization or handled correctly. The need to implement a formal security vulnerability response program was also identified by VirtualWare.

Implementation Strategy

The adoption of a security assurance program within an organization is a long term strategy. There are significant impacts on the developer culture, as well as the business' development and delivery of its applications. The adoption of this strategy is set over a 12 month period, and due to the size of the organization, will be relatively easy to implement in that period.



/ PHASE 1 (MONTHS 0 - 3) - AWARENESS & PLANNING

VirtualWare previously identified that they had limited knowledge and awareness of application security threats to their organization, and limited secure coding experience. The first phase of the VirtualWare deployment focused on training developers, and implementing guidance and programs to identify current security vulnerabilities.

Development teams within VirtualWare had limited experience in secure coding techniques, therefore, an initial training program was developed to provide the organization's developers with defensive programming techniques.

With over 300 developers and multiple languages supported within the organization, one of the key challenges for VirtualWare was to provide an education program that was technical enough to teach developers some of the basics in secure coding concepts. The objective of this initial education course was primarily on coding techniques and testing tools. The course developed and delivered within the organization lasted for one day, and covered the basics of secure coding.

VirtualWare was aware that they had a number of applications with vulnerabilities, and no real strategy in which to identify existing vulnerabilities or address the risks in a reasonable timeframe. A basic risk assessment methodology was adopted, and the organization undertook a review of the existing application platforms.

This phase also included implementing a number of concepts for the development team to enhance their security tools. The development teams already had a number of tools available to perform quality type assessments. Additional investigation into code review and security testing tools was performed.

Target Objectives

During this phase of the project, VirtualWare implemented the following SAMM practices & activities:

	SM 1	A. Estimate overall business risk profile B. Build and maintain assurance program roadmap
	EG 1	A. Conduct technical security awareness training B. Build and maintain technical guidelines
	SR 1	A. Derive security requirements from business functionality B. Evaluate security and compliance guidance for requirements
	IR 1	A. Create review checklists from known security requirements B. Perform point review of high-risk code
	ST 1	A. Derive test cases from known security requirements B. Conduct penetration testing on software releases
	IM 1	A. Identify point of contact for security issues B. Create informal security response team(s)

To achieve these maturity levels, VirtualWare implemented a number of programs during this phase of the rollout. The following initiatives were adopted:

- One Day Secure Coding Course (high-level) for all developers.
- Build a technical guidance whitepaper for application security on technologies used within the organization.
- Create a risk process and perform high-level business risk assessments for the application platforms, and review business risk.
- Prepare initial technical guidelines and standards for developers.
- Perform short implementation reviews on application platforms that present significant risk to the organization.
- Develop test and use cases for projects and evaluate the cases against the applications.
- Appointed a role to application security initiatives.
- Generated a draft strategic roadmap for the next phase of the assurance program.

Due to the limited amount of expertise in house within VirtualWare, the company engaged with a third-party security consulting group to assist with the creation of the training program, and assist in writing the threat modeling and strategic roadmap for the organization.

One of the key challenges faced during this phase was to get all 300 developers through a one day training course. To achieve this, VirtualWare ran 20 course days, with only a small number of developers from each team attending the course at one time. This reduced the overall impact on staff resources during the training period.

During this phase of the project, VirtualWare invested significant resources effort into the adoption of a risk review process, and reviewing the business risk to the organization. Although considerable effort was focused on these tasks, they were critical to ensuring that the next steps implemented by VirtualWare were in line with the business risks faced by the organization.

VirtualWare management received positive feedback from most developers within the organization on the training program. Although not detailed, developers felt that the initial training provided some basic skills that could assist them immediately with writing secure code.

Implementation Costs

A significant amount of internal resources and costs were invested in this phase of the project. There were three different types of costs associated with this phase.

Internal Resource Requirements

Internal resource effort used in the creation of content, workshops, and review of application security initiatives within this phase. Effort is shown in total days per role.



Training Resource Requirements (Training per person for period)

Each developer within VirtualWare was required to attend a training course, and therefore, every developer had a single day allocated to the application security program.



Outsourced Resources

Due to the lack of knowledge within VirtualWare, external resources were used to assist with the creation of content, and creation/delivery of the training program to the developers.



/ PHASE 2 (MONTHS 3 - 6) - EDUCATION & TESTING

VirtualWare identified in Phase 1 that a number of their applications contained vulnerabilities that may be exploited by external threats. Therefore, one of the key objectives of this phase was to implement basic testing and review capabilities to identify the vulnerabilities and address them in the code.

Automated tools were introduced to assist with code coverage and findings weaknesses as it was identified as one of the biggest challenges in this phase of the implementation. Traditionally, in the past developers have used automated tools with great difficulty, and therefore implementing new tools was seen as a significant challenge.

To ensure a successful rollout of the automation tools within the organization, VirtualWare proceeded with a staged rollout. The tools would be given to senior team leaders first, with other developers coming online over a period of time. Teams were encouraged to adopt the tools, however, no formal process was put in place for their use.

This phase of the implementation also saw the introduction of a more formal education and awareness program. Developers from the previous training requested more specific training in the areas of web services and data validation. The new six hour-specific training course was developed with these two focus areas. VirtualWare also implemented additional training programs for architects and managers, and adopted an awareness campaign within the organization.

Target Objectives

During this phase of the project, VirtualWare implemented the following SAMM practices & activities:

		A. Classify data and applications based on business risk B. Establish and measure per-classification security goals
		A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams
		A. Build and maintain application-specific threat models B. Develop attacker profile from software architecture
		A. Identify software attack surface B. Analyze design against known security requirements
		A. Utilize automated code analysis tools B. Integrate code analysis into development process
		A. Utilize automated security testing tools B. Integrate security testing into development process
		A. Capture critical security information for operations B. Document procedures for typical application alerts

To achieve these maturity levels VirtualWare implemented a number of programs during this phase of the rollout. The following initiatives were adopted:

- Additional education and training courses for QA testers, managers, and architects.
- Conduct data asset classification and set security goals.
- Develop the risk assessment methodology into a threat modeling approach with attack trees and profiles.
- Review and identify security requirements per application platform.
- Introduction of automated tools to assist with code coverage, and security analysis of existing applications and new code bases.
- Review and enhance existing penetration testing programs.
- Enhance the existing software development lifecycle to support security testing as a part of the development process.

VirtualWare adapted the existing application security training program to provide a less technical version of a Business Application Security Awareness program. This was a shorter, four hour course, and was extended to managers and business owners of the organization.

A high-level review of the existing implementation review and penetration testing programs identified that the process was inadequate, and needed to be enhanced to provide better testing and results on application security vulnerabilities. The team set out to create new penetration testing and implementation review programs. As a part of these programs, each senior developer on a program team was allocated approximately four days to perform a high-level source implementation review of their application.

VirtualWare management understood that the infrastructure and applications were tightly integrated, and during this phase, the operational side of the application platforms (infrastructure) were reviewed. This phase looked at the infrastructure requirements and application integration features between the recommended deployed hardware and the application interfaces.

During this phase, the strategic roadmap and methodology for application security was reviewed by the project team. The objective of this review and update was to formally classify data assets, and set the appropriate level of business risk associated with the data assets and applications. From this, the project team was able to set security goals for these applications.

Implementation Costs

A significant amount of internal resources and costs were invested in this phase of the project. There were three different types of costs associated with this phase.

Internal Resource Requirements

Internal resource effort used in the creation of content, workshops, and review of application security initiatives within this phase. Effort is shown in total days per role.



Training Resource Requirements (Training per person for period)

Additional personnel within VirtualWare were required to attend a training course, and therefore several roles had time allocated to training on application security.



Outsourced Resources

Due to the lack of knowledge within VirtualWare, external resources were used to assist with the creation of content, and creation/delivery of the training program to the developers.



PHASE 3 (MONTHS 6 - 9) - ARCHITECTURE & INFRASTRUCTURE

The third phase of VirtualWare's software assurance program builds on the previous implementation phases, and focuses on risk modeling, architecture, infrastructure, and operational enablement capabilities.












The key challenge in this phase was establishing a tighter integration between the application platforms and operational side of the organization. In the previous phase, VirtualWare teams were introduced to issue management and the operational side of application security. During this phase, VirtualWare has adopted the next phase of these areas, and introduced clear incident response processes and detailed change control procedures.

VirtualWare has chosen to start two new areas for this implementation. Although VirtualWare is not impacted by regulatory compliance, a number of their customers have started to ask about whether the platforms can assist in passing regulatory compliance. A small team has been set up within VirtualWare to identify the relevant compliance drivers and create a checklist of drivers.

In the previous phase, VirtualWare introduced a number of new automated tools to assist with the review and identification of vulnerabilities. Although not focused on in this phase, the development teams have adopted the new tools, and reported that they are starting to gain a benefit from using these tools within their groups.

Target Objectives

During this phase of the project, VirtualWare implemented the following SAMM practices & activities:

		A. Identify and monitor external compliance drivers B. Build and maintain compliance guidelines
		A. Build and maintain abuse-case models per project B. Adopt a weighting system for measurement of threats
		A. Build an access control matrix for resources and capabilities B. Specify security requirements based on known risks
		A. Maintain list of recommended software frameworks B. Explicitly apply security principles to design
		A. Inspect for complete provision of security mechanisms B. Deploy design review service for project teams
		A. Establish consistent incident response process B. Adopt a security issue disclosure process
		A. Create per-release change management procedures B. Maintain formal operational security guides

To achieve these maturity levels, VirtualWare implemented a number of programs during this phase of the rollout. The following initiatives were adopted:

- Define and publish technical guidance on security requirements, and secure architecture for projects within the organization.
- Identify and document compliance and regulatory requirements.
- Identify and create guidelines for security of application infrastructure.
- Create a defined list of approved development frameworks.
- Enhance the existing threat modeling process used within VirtualWare.
- Adopt an incident response plan and prepare a security disclosure process.
- Introduce change management procedures and formal guidelines for all projects.

To coincide with the introduction of automated tools for developers (from the previous phase), formal technical guidance on secure coding techniques was introduced into the organization. These were specific technical documents relating to languages and technology, and provided guidance on secure coding techniques in each relevant language/application.

With a combined approach from the education and awareness programs, technical guidance, and the introduction of automation tools to help the developers, VirtualWare started to see a visible difference in the code being delivered into production versions of their applications. Developers provided positive feedback on the tools and education made available to them under the program.

For the first time, VirtualWare project teams took responsibility for the security and design of their application platforms. A formal review process and validation against best practices were performed by each team during this phase. Some teams identified gaps relating to both security and business design that needed to be reviewed. A formal plan was put in place to ensure these gaps were addressed.

A formal incident response plan and change management procedures were introduced during this phase of the project. This was a difficult process to implement, and VirtualWare teams initially struggled with the process, as the impact on culture and the operational side of the business was significant. However, over time each team member identified the value in the new process, and the changes were accepted by the team over the implementation period.

Implementation Costs

A significant amount of internal resources and costs were invested in this phase of the project. There were two different types of costs associated with this phase.

Internal Resource Requirements

Internal resource effort used in the creation of content, workshops, and review of application security initiatives within this phase. Effort is shown in total days per role.



Outsourced Resources

Due to the lack of knowledge within VirtualWare, external resources were used to assist with the creation of content, and creation/delivery of the processes, guidelines, and to assist teams.



/ PHASE 4 (MONTHS 9 - 12) - GOVERNANCE & OPERATIONAL SECURITY

The fourth phase of the VirtualWare software assurance program continues on from the previous phases by enhancing existing security functions within the organization. At this point, VirtualWare has implemented a number of critical application security processes and mechanisms to ensure that applications are developed and maintained securely.

A core focus in this phase is bolstering the alignment and governance disciplines. These functions play a critical role in the foundation of an effective, long term application security strategy. A completed education program is implemented, whilst at the same time, a long term strategic roadmap is put in place for VirtualWare.









The other key focus within this phase is on the operational side of the implementation. VirtualWare management previously identified a critical need for incident response plans and dedicated change management process in their long term strategy.

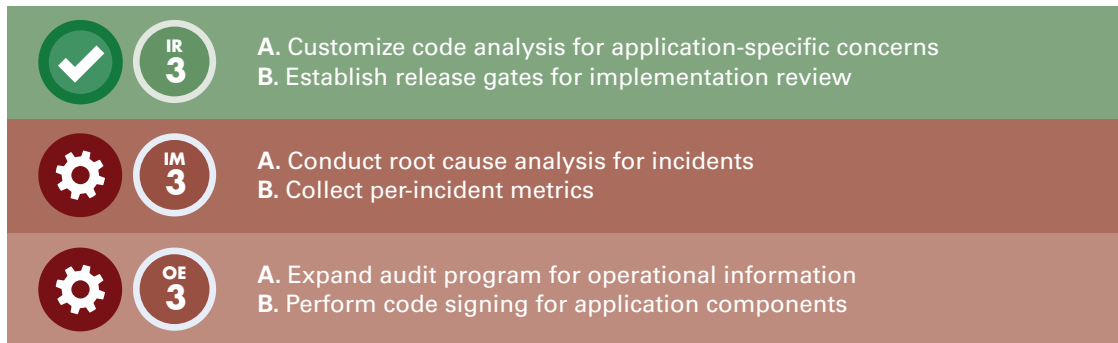
VirtualWare saw this phase as the stepping stones to their longevity. This phase saw the organization implement a number of final measures to cement the existing building blocks that have been laid down in the previous phases. In the long term, this will ensure that the processes, concepts, and controls put in place will continue to work within the organization to ensure the most secure outcome for their application platforms.

VirtualWare chose this phase to introduce their customers to their new application security initiatives by providing their customers with the details of a series of application security initiatives. VirtualWare deployed applications securely and provided the ability to report vulnerabilities in their applications. The key goal from these programs is to instill confidence in their customer base by showing that VirtualWare applications are built with security in mind, and also, by assisting customers to ensure their application environments using VirtualWare technology are secure.

Target Objectives

During this phase of the project, VirtualWare implemented the following SAMM practices and activities:

		<ul style="list-style-type: none"> A. Conduct periodic industry-wide cost comparisons B. Collect metrics for historic security spend
		<ul style="list-style-type: none"> A. Build policies and standards for security and compliance B. Establish project audit practice
		<ul style="list-style-type: none"> A. Create formal application security support portal B. Establish role-based examination/certification
		<ul style="list-style-type: none"> A. Build security requirements into supplier agreements B. Expand audit program for security requirements



To achieve these maturity levels, VirtualWare implemented a number of programs during this phase of the rollout. The following initiatives were adopted:

- Create well defined security requirements and testing program for all projects.
- Create and implement a incident response plan.
- Reviewed existing alerts procedure for applications, and document a process for capturing events.
- Create a customer security whitepaper on deploying applications security.
- Review existing security spend within projects and determine if appropriate budget has been allocated to each project for security.
- Implement the final education and awareness programs for application roles.
- Complete a long term application security strategy roadmap for the organization.

In previous phases,, VirtualWare had released a formal incident response plan for customers to submit vulnerabilities found with their code. During this phase, VirtualWare took the results of the submitted vulnerabilities and conducted assessments of why the problem occurred, how, and attempted a series of reporting to determine any common theme identified amongst the reported vulnerabilities.

As a part of the ongoing effort to ensure applications are deployed internally securely, as well as on customer networks, VirtualWare created a series of whitepapers, provided to customers based on industry standards for recommended environment hardening. The purpose of these guidelines is to provide assistance to customers on the best approach to deploying their applications.

During this phase, VirtualWare implemented a short computer-based training module so that existing and new developers could maintain their skills in application security. It was also mandated that all “application” associated roles undertake a mandatory one day course per year. This was completed to ensure that the skills given to developers were not lost and new developers would be up skilled during their time with the company.

One of the final functions implemented within VirtualWare was to complete a “AS IS” gap assessment, and review, and determine how effective the past 12 months had been. During this short program questionnaires were sent to all team members involved, as well as a baseline review against SAMM. The weaknesses and strengths identified during this review were documented into the final strategic roadmap for the organization, and the strategy for the next 12 months was set for VirtualWare.

Implementation Costs

A significant amount of internal resources and costs were invested in this phase of the project. There were two different types of costs associated with this phase.

Internal Resource Requirements

Internal resource effort used in the creation of content, workshops and review of application security initiatives within this phase. Effort is shown in total days per role.



Outsourced Resources

Due to the lack of knowledge within VirtualWare, external resources were used to assist with the implementation of this phase, including documentation, processes, and workshops.



/ ONGOING (MONTHS 12+)

Over the past 12 months, VirtualWare implemented a number of training and education programs, along with developing internal guidelines and policies. In the final phase of the assurance program implementation, VirtualWare began to publish externally and work with their customers to enhance the security of their customer application platforms.

VirtualWare Management set an original mandate to ensure that software developed within the company was secure, and ensured that the market was aware of the security initiatives taken and assisted customers in securing their application platforms.

To achieve these management goals, the first 12 months set the path for an effective strategy within VirtualWare, starting from awareness to assisting customers in securing their application environments. Moving forward, VirtualWare has set a number of initiatives within the organization to ensure that the company doesn't fall into their old habits. Some of these programs include:

- Business owners and team leaders are aware of the risk associated with their applications and are required to sign off on applications before release.
- Team leaders now require all applications to formally go through the security process, and implementation reviews are performed weekly by developers.
- Ongoing yearly training and education programs (including CBT) are provided to all project staff, and developers are required to attend a course at least once a year.
- A dedicated team leader for application security has been created, and is now responsible for customer communications, and customer technical papers and guidelines.

Going forward, VirtualWare has created a culture of security as part of their SDLC, thus ensuring that applications developed and provided to customers are secure and robust. An effective process has been put in place where vulnerabilities can be reported on and handled by the organization when required.

During the final implementation phase, a project gap assessment was performed to identify any weaknesses that appeared during the implementation. In particular, due to the high-turnover of staff, VirtualWare needed to constantly train new developers as they started with the organization. A key objective set to address this problem was a program introduced specifically for developers so that they receive formal security training when they start with the organization. This also helps to create the mindset that security is important within the organization and its development team.

Maturity Scorecard

The maturity scorecard was completed as a self assessment during the implementation of the software assurance program by VirtualWare. The final scorecard (shown to the right) represents the status of VirtualWare at the time it began and the time it finished its four-phase improvement project.



/ ACKNOWLEDGEMENTS AND SPONSORS

This document was originally created through the OpenSAMM Project led by Pravir Chandra (chandra@owasp.org), an independent software security consultant. Creation of the first draft was made possible through funding from Fortify Software, Inc. Since the initial release of SAMM, this project has become part of the Open Web Application Security Project (OWASP). This document is currently maintained and updated through the OWASP SAMM Project led by Sebastien Deleersnyder, Bart De Win, and Brian Glas. Thanks to the many contributors and reviewers (listed below) and to the supporting organizations that are listed in the Sponsors section.

/ CONTRIBUTORS & REVIEWERS

This work would not be possible without the support of many individual reviewers and experts that offered contributions and critical feedback.

- Fabio Arciniegas
- Bart De Win
- James McGovern
- Matt Bartoldus
- John Dickson
- Matteo Meucci
- Jonathan Carter
- Alexios Fakos
- Jeff Payne
- Darren Challey
- David Fern
- Gunnar Peterson
- Brian Chess
- Brian Glas
- Jeff Piper
- Justin Clarke
- Kuai Hinojosa
- Andy Steingruebl
- Dan Cornell
- Jerry Hoff
- John Steven
- Michael Crague
- Carsten Huth
- Chad Thunberg
- Dinis Cruz
- Bruce Jenkins
- Colin Watson
- Sebastien Deleersnyder
- Daniel Kefer
- Jeff Williams
- Justin Derry
- Yan Kravchenko
- Steven Wierckx

/ SPONSORS

We would like to thank the following sponsors who donated funds to the SAMM project.

