Current Level1 ActA:

Q: Should we change data and application to assets everywhere? Assets include hardware, software application, data, permissions, people etc.

Objective Level 1: Measure relative value of data and software assets and choose risk tolerance

Proposed Level1 ActA: Estimate overall business risk profile

To develop risk-based assurance program, interview business owners and stakeholders and create a list of worst-case scenarios across the organization’s various ~~application and data~~ assets including application and data. Based on the way in which your organization builds, uses, or sells software, the list of worst-case scenarios can vary widely, but common issues include data theft or corruption, service outages, monetary loss, reverse engineering, account compromise, etc. After broadly capturing worst-case scenario ideas, collate and select the most important based on collected information and knowledge about the core business. Any number can be selected, but aim for at least 3 and no more than 7 to make efficient use of time and keep the exercise focused. Elaborate a description of each of the selected items and document details of contributing worst-case scenarios, potential contributing factors, and potential mitigating factors for the organization. The final business risk profile should be reviewed with business owners and other stakeholders for understanding.

While creating final business risk profile, explicitly reference evolving threat landscape and recent incidents at other companies to bring awareness. Make it clear how threats could become real incidents when threats are not addressed proactively.

Proposed Level1 ActB: Classify assets ~~data and applications~~ based on business risk

Establish a simple classification system to represent risk-tiers for applications. In its simplest form, this can be a High/Medium/Low categorization. More sophisticated classifications can be used, but there should be no more than seven categories and they should roughly represent a gradient from high to low impact against business risks. Working from the organization’s business risk profile, create project evaluation criteria that maps each project to one of the risk categories. A similar but separate classification scheme should be created for data assets and each item should be weighted and categorized based on potential impact to business risks. Evaluate collected information about each application and assign each a risk category based upon overall evaluation criteria and the risk categories of data assets in use. This can be done centrally by a security group or by individual project teams through a customized questionnaire to gather the requisite information. An ongoing process for application and data asset risk categorization should be established to assign categories to new assets and keep the existing information updated at least biannually.

Objective Level 2: Articulate business case; and Establish unified strategic roadmap for software security within the organization

Proposed Level 2 ActA: Build and maintain assurance program roadmap

Understanding the main business risks to the organization, build a business case to justify expenditure. Evaluate the current performance of the organization against each of the twelve Practices. Assign a score for each Practice from 1, 2, or 3 based on the corresponding Objective if the organization passes all the cumulative success metrics. If no success metrics are being met, assign a score of 0 to the Practice. Once a good understanding of current status is obtained, the next goal is to identify the Practices that will be improved in the next iteration. Select them based on business risk profile, other business drivers, compliance requirements, budget tolerance, etc. Once Practices are selected, the goals of the iteration are to achieve the next Objective under each. Iterations of improvement on the assurance program should be approximately 3-6 months, but an assurance strategy session should take place at least every 3 months to review progress on activities, performance against success metrics and other business drivers that may require program changes.

Proposed Level 2 ActB: Establish and measure per-classification security goals and align security expenditure with relevant business indicators and asset value

With a classification scheme for the organization’s application portfolio in place, direct security goals and assurance program roadmap choices can be made more granular. The assurance program’s roadmap should be modified to account for each application risk category by specifying emphasis on particular Practices for each category. For each iteration of the assurance program, this would typically take the form of prioritizing more higher-level Objectives on the highest risk application tier and progressively less stringent Objectives for lower/other categories. This process establishes the organization’s risk tolerance since active decisions must be made as to what specific Objectives are expected of applications in each risk category. By choosing to keep lower risk applications at lower levels of performance with respect to the Security Practices, resources are saved in exchange for acceptance of a weighted risk. However, it is not necessary to arbitrarily build a separate roadmap for each risk category since that can leads to inefficiency in management of the assurance program itself.

Using the application risk categories and the respective prescribed assurance program roadmaps for each, a baseline security cost for each application can be initially estimated from the costs associated with the corresponding risk category. Combine the application-specific cost information with the general cost model based on risk category, and then evaluate projects for outliers, i.e. sums disproportionate to the risk rating. These indicate either an error in risk evaluation/classification or the necessity to tune the organization’s assurance program to address root causes for security cost more effectively. The tracking of security spend per project should be done quarterly at the assurance program strategy session, and the information should be reviewed and evaluated by stakeholders at least annually. Outliers and other unforeseen costs should be discussed for potential affect on assurance program roadmap.

Objective Level 3: Establish objective measurements to help improve security posture of the organization and manage risk

Proposed Level 3 ActA: Identify metrics to measure effectiveness and progress

The intent of metrics is to define and measure progress against security roadmap in quantitative terms. Make sure metrics are well understood by all stakeholders and drive necessary behavior change. It is advisable to stay away from qualitative metrics. Example of metric could be number of detected & fixed vulnerabilities in a given time frame. Metrics should be collected in automated fashion to avoid additional collection overhead. Good metrics are agnostic of development methodology – waterfall, agile, iterative since various parts of the organization might be using different methodology.

Proposed Level 3 ActB: Use metrics to advance security goals

With well-defined metrics in place and metrics collection mechanism established, regularly share metrics on security posture with all stakeholders including employees. The intent is to bring awareness of the progress being made and drive improvement and behavior change. Keep metrics to manageable number to retain their effectiveness and usefulness. It’s better to use few meaningful metrics that organization will actually use. The tracking of security spend per project should be done quarterly at the assurance program strategy session, and the information should be reviewed and evaluated by stakeholders at least annually. Outliers and other unforeseen costs should be discussed for potential affect on assurance program roadmap.

The information might come as a dashboard with metrics for executives and software development management. Sometimes, publication is not shared with everyone in a firm, but rather with the relevant executives only. In this case, publishing information up to executives who then drive change in the organization is necessary.

Proposed Level 3 ActB: Document and implement Risk Management strategy

A pre-requisite for proactive risk management is to have an inventory of the mission critical applications with associated risk profiles that allow CISOs to identify the critical digital assets such as data and functions that need to be prioritized and planned for proactive risk mitigation activities

asset management - these assets include the applications, assets management requires an inventory of the applications that are managed by the organization in order to implement a risk management approach. This inventory includes information on the type of applications, the risk profile for each application, the type of data that is stored and processed, patching requirements and the security assessments such as vulnerability testing that are required.

Technical risks can be measured as the impact of confidentiality, integrity and availability of the asset caused by a technical event/cause such as vulnerability that is identified by an application security assessment.

HIGH, for example, to prioritize such vulnerability for mitigation ahead of vulnerabilities that are scored as MEDIUM or LOW risk

Business risk management occurs when the value of the asset is taken into account to determine the impact to the organization. This requires the association of technical risk of the vulnerability with the asset value to quantify the risk. The risk can be factored as the likelihood of the asset being compromised and the business impact caused by the exploit of the vulnerability.

**Risk management strategies**

Level of risk impact and probability

### Identify gate locations, gather necessary artifacts.[SM1.4: 81]

The software security process includes release gates/checkpoints/milestones at one or more points in the SDLC or, more likely, the SDLCs. The first two steps toward establishing security-specific release gates are: 1) to identify gate locations that are compatible with existing development practices and 2) to begin gathering the input necessary for making a go/no-go decision. Importantly at this stage, the gates are not enforced. For example, the SSG can collect security testing results for each project prior to release, but stop short of passing judgment on what constitutes sufficient testing or acceptable test results. Shorter release cycles, as are seen in organizations practicing agile development, often require creative approaches to collecting the right evidence and rely heavily on lightweight, super-fast automation. The idea of identifying gates first and only enforcing them later is extremely helpful in moving development toward software security without major pain. Socialize the gates, and only turn them on once most projects already know how to succeed. This gradual approach serves to motivate good behavior without  
requiring it.

### Enforce gates with measurements and track exceptions.

SDLC security gates are now enforced: in order to pass a gate, a project must either meet an established measure or obtain a waiver. Even recalcitrant project teams must now play along. The SSG tracks exceptions. A gate could require a project to undergo code review and remediate any critical findings before release. In some cases, gates are directly associated with controls required by regulations, contractual agreements, and other business obligations, and exceptions are tracked as required by statutory or regulatory drivers. In other cases, gate measures yield key performance indicators that are used to govern the process. A revolving door or a rubber stamp exception process does not count. If some projects are automatically passed, that defeats the purpose of enforcing gates. Even seemingly innocuous development projects, such as a new mobile client for an existing back-end or porting an application to a cloud environment from an internal data center, must successfully pass the prescribed security gates in order to progress.

### [SM2.5: 19] Identify metrics and use them to ~~drive budgets~~. measure software security initiative progress

The SSG and its management choose the metrics that define and measure software security initiative progress. ~~These metrics will drive the initiative’s budget and allocation of resources, so simple counts and statistics won’t suffice~~. Metrics also allow the SSG to explain its goals and its progress in quantitative terms. One such metric could be security defect density. A reduction in security defect density could be used to show a decreasing cost of remediation over time. ~~Recall that in agile methodologies, metrics are best collected early and often in a lightweight manner.~~ The key here is to tie technical results to business objectives in a clear and obvious fashion in order to justify funding. ~~Because the concept of security is already tenuous to business people, making this explicit tie can be very helpful.~~

### [SM2.6: 29] Require security sign-off.

The organization has an initiative-wide process for accepting security risk and documenting accountability. A risk acceptor signs off on the state of all software prior to release. For example, the sign-off policy might require the head of the business unit to sign off on critical vulnerabilities that have not been mitigated or SSDL steps that have been skipped. The policy must apply to outsourced projects, such as a boutique mobile application, and to projects that will be deployed in external environments, such as the cloud. Informal or uninformed risk acceptance alone does not count as security sign off, as the act of accepting risk is more effective when it is formalized (e.g., with a signature, form submission, or something similar) and captured for future reference. ~~Similarly, simply stating that certain projects never need a sign-off does not achieve the desired results.~~