Security Measurement

White Paper

V3.0 13 January 2006

Prepared on behalf of the PSM Safety & Security TWG

Practical Software and Systems Measurement

Comments to:
John Murdoch
Computer Science Department
University of York
YORK YO10 5DD UK

44 1904 43 2749
jm48@york.ac.uk

This paper is subject to the following copyright restrictions:

- The PSM Support Center is the sole manager of all PSM products and services and is the only one authorized to modify them.

- General Use: Permission to reproduce, use this document or parts thereof, and to prepare derivative works from this document is granted, with attribution to PSM and the original author(s), provided this copyright notice is included with all reproductions.

- Supplemental Materials: Additional materials may be added for tailoring or supplemental purposes, if the material developed separately is clearly indicated. A courtesy copy of additional materials shall be forwarded the PSM Support Center, psm@pica.army.mil. The supplemental materials will remain the property of the author(s) and will not be distributed.

- Author Use: Authors have full rights to use their contributions in an unfettered way with credit to the technical source.

Table of Contents

Executive Summary 5
PSM Safety & Security Measurement TWG 6
1 Introduction 7
1.1 Objectives 7
1.2 Context 7
1.3 Solution Strategy 9
1.4 Related Work 9
2 Fundamentals 10
2.1 Security and Dependability 10
2.2 Faults and Errors 12
2.3 Threats 13
2.4 Aggregating Security Properties 14
2.5 System and Component Properties 17
2.6 Trust and Assurance 18
3 Systems Theoretic Model 20
3.1 Measurement of Delivered Services 20
3.2 Measurement during Development 21
3.3 Establishing Benefits of Security Interventions 21
3.4 Systems Approach to Assurance 22
3.5 Optimistic and Pessimistic Views 22
4 Representative Security Practices 22
4.1 Security Principles and Policies 23
4.2 Types of Product and Service 23
4.3 Security Engineering 28
4.4 Operations 30
4.5 Risk Assessment 31
4.5.1 Modeling Threat Agents 31
4.5.2 Vulnerability Assessment 33
4.6 Evaluation, Testing 33
5 Information Needs Model 34
5.1 Common Information Needs 35
5.1.1 Viewpoint in Time 37
5.1.2 Costs and benefits 37
5.1.3 Uncertainty 39
5.1.4 Acquisition and Trust 40
5.2 Enterprise Level 40
5.3 Organization Level 41
5.4 Project Level 43
5.5 Technical/Professional Specialty Level 43
6 Security Measurement Map 46
7 Developing Security Measurements 47
8 Conclusion 51
9 References 52
Appendix 1 Glossary 54
Appendix 2 Security Risk 56
Appendix 3 Representative Practices 59

List of Figures

Figure 1 Security and safety processes viewed as specialist domains contributing to core systems engineering (SE), management and operations processes 8
Figure 2 Context in which security is defined: a provider system, a user system and a provided service [13]..... 10
Figure 3 The propagation of the effects of a fault, from its activation to create an error in the system state, to propagation to a failure in a provided service, to causing a fault in a user system [13]. Called a 'fault path' in the text..... 12
Figure 4 A fault path models the causal links between external threats (in this case), system faults, service failures and damages in the user environment. Such paths can be viewed (1) as hypothesis to guide design and operational policy, (2) as unfolding, in terms of security event management during operations and (3) retrospectively, as a basis for lessons learnt.13
Figure 5 Concept model relating terms defined in the text..... 14
Figure 6 Fault Trees and Attack Trees for a system as interpreted by developers and attackers . 15
Figure 7 Example simplification of a Fault Tree to form a Fault Path 16
Figure 8 Simple model of aggregation of security properties; fault paths are central to the integration of component properties and association with service failures 17
Figure 9 Fault Tree with agent-based components 18
Figure 10 Closed loop feedback from delivered service failures 20
Figure 11 Closed loop feedback based on concurrent monitoring and adjustment 21
Figure 12 Organization model comprising a developer, operator and user of a system..... 24
Figure 13 Typology of systems..... 26
Figure 14 Template system architectures to support security management (sketches only)..... 27
Figure 15 Linear view of development and operations processes with example sources of security risks 28
Figure 16 Model showing the application of security principles, expressed via security policies, to a system development project..... 29
Figure 17 Aspects of a threat agent, from [28] 32
Figure 18 Attack Tree 33
Figure 19 Information Needs Model: developer organizations 35
Figure 20 Generic information needs arising when a plan is enacted under uncertainty 36
Figure 21 Dependency between failure frequency and severity, and costs invested in reduction of these [31]..... 38
Figure 22 Security Measurement Map - system development..... 46
Figure 23 Illustration of the strategy used to develop security measures: top-down (based on information needs) and bottom-up (based on measurable artifacts)..... 48
Figure 24 Measurement applications in the development of a software-intensive system and its use in providing a security-critical service. 50

List of Tables

Table 1 Example information needs of managers of security-related work 45
Table 2 Traditional ROI calculation based on discounted cash flows, from [23] 60
Table 3 Example tracking of security threats and events 66

Executive Summary

Work performed by the PSM Technical Working Group on Safety & Security Measurement from February 2004 to December 2005 is reported. The objective is to tailor the PSM and ISO/IEC 15939 measurement framework to serve information needs relating to the security of software-intensive systems. This report addresses preliminary work on the fundamentals of security measurement, in preparation for the development of measurement guidance materials, to be documented in a subsequent report.

Related security metrics work includes the NIST SP series of documents, particularly NIST SP 800 55, current work in ISO/IEC JTC1/SC27 on the information security metrics standard ISO/IEC 27004, and the work of the Measurement Working Group of ISSEA in the frame of the Security Engineering Capability Maturity Model SSE-CMM and the related standard ISO/IEC 21827. Other related work includes the security metrics recommendations of a CISWG study that focuses on information policy compliance and the DHS Software Assurance Program.

This report seeks to complement these works by developing an integrated measurement framework, covering both development and operations, that emphasizes the engineering management and assurance of security properties. Security is taken to mean the ability of a system to prevent particular types of service failures from occurring with unacceptable frequency or severity. Assessing the security of systems is not straightforward. A risk-based framework is selected reflecting the uncertainty in predictions about future system performance. The framework embraces the design approach in which engineering knowledge is applied to achieve component and system properties judged to contribute to security risk reduction.

A 'systems-theoretic' model of measurement is proposed, in which measurement is integrated into decision-making loops that include the decision-maker who is informed by the measurement, the actions available, and the subject domain where actions are applied and measurements taken. This approach seeks to integrate the use of the classic PDCA (Plan Do Check Act) cycle with the multi-pass approach advocated in the project risk management field. Furthermore, the model provides a conceptual approach to increasing the role of objective measurement in decision situations dominated by expert, but subjective, judgment.

The report proposes a categorization of security information needs, building on existing PSM guidance materials under the headings of: Security Engineering; Schedule & Progress; Resources and Cost; Compliance; Performance Outcomes; Security Risk Management; and Assurance. The measurement framework is proposed as a platform for developing guidance materials compatible with the PSM methodology (i.e. measurement constructs, specifications, process guidance).

PSM Safety & Security Measurement TWG

Workshop Contributors:

Dennis Ahern, *Northrop Grumman*
Jeff Allen, *LM*
Frances Anderson, *Aerospace Corporation*
Matt Ashford, *SEI*
Nadya Bartol, *BAH*
Molly Campbell, *US Army ARDEC*
Paul Caseley, *DSTL UK MoD*
Vivian Cocca, *DoD*
Erin Fitzsimmons, *Rockwell Collins*
Phil Flora, *Texas Guaranteed Student Loan Corp*
John Gaffney, *Lockheed Martin*
Gary Hafen, *LM*
Jan Janigan, *DoD*

Joe Jarzombek, *National Cyber Security Division, DHS*
Cheryl Jones, PSM, *US Army ARDEC*
Mike Kass, *NIST*
Greg Larsen, *Inst for Defence Analyses*
John Van Orden
Jim McCurley, *SEI*
Jim Moore, *Mitre Corporation*
John Murdoch, *University of York UK*
Dana Van Orman, *DCMA*
Sam Redwine, *JMU*
Don Reifer, *Reifer Consultants*
John Riedener, *US Army*
Rob Robason, *Wind River Systems*
Garry Roedler, *LM*
Amos Rohrer, *BAE Systems*
Ioana Rus, *Fraunhofer USA, University of Maryland*
David Seaver, *PRICE Systems*

Dave Zubrow, *SEI*

Thanks also to the additional reviewers of this report, including:
Fred Hall
Ken Astley, *Loughborough University, UK*
Antony Powell, *University of York, UK*

1 Introduction

1.1 Objectives

This White Paper reports on research on the application of measurement principles to the security properties of software-intensive systems. The work has the objective of integrating security measurement with the general measurement principles as developed by the PSM project [1] and in accordance with the related standard ISO/IEC 15939:2002 [2]. It provides a rationale for security measurement recommendations issued by the PSM project in associated documentation [3]. The application of measurement principles to security is a relatively new field and presents several challenges, explored in the following sections.

In the tradition of the PSM project, this work aims to provide an integrative measurement framework, providing a bridge between technical, specialist domains, and the information needs of managers at project, capability and organization levels. Incorporating security measurement into a wider measurement framework enables re-use of generic measurement concepts (e.g. *cost* and *progress against plan*) as well as supporting trade-offs with other types of performance, as part of a systems engineering management approach.

The question addressed by this report is how best to apply measurement principles to support the achievement and assurance of security in software-intensive systems and provided services.

Several issues motivate the question. Expenditure on security has to be justified against competing demands. While appropriate security actions are understood at technical and organizational policy levels, it is often difficult to establish a quantifiable connection with service quality and assurance, as perceived by end users. Questions about *where* and *how* security should be improved in a particular system, *how much* security investment should be made and *how much* assurance should be undertaken, are not answered systematically or quantifiably today.

The underlying need is to link investments in security with predictions of improved performance, as perceived by users of the provided services, and to improve predictive ability through validation against actual performance. Without some quantification of such causal links, we are limited in being able to judge the effectiveness and efficiency of different security actions and to perform trade-offs.

1.2 Context

The security of a system is taken to mean the ability of the system to deliver a service in which the frequency and severity of defined types of service failure are acceptably low. Definitions follow the terminology of [13] and are collected together in Appendix 1.

The term *system* is used here in a general sense: it might be a software component, computer system, network or organization. In line with the main concerns of the PSM project, attention is directed at the technical management of the development and operation of software-intensive systems.

The types of failure of concern are security-related: for example, unauthorized disclosure of information, improper/unauthorized access to system state, inability to track accountability for actions and interference in control functions. The causes of service failures are called *faults*, a general term covering defects, vulnerabilities, mistakes, mishaps, recreational actions and successful attacks from malicious agents. It follows that security is a property of a system *in relation to* the dangers facing it, which can be internal or external, non-malicious or malicious. Security in a software-intensive system is achieved and preserved by a wide range of activities associated with the original system development process and with system operations in the use environment, including maintenance. System development typically involves a core integrative

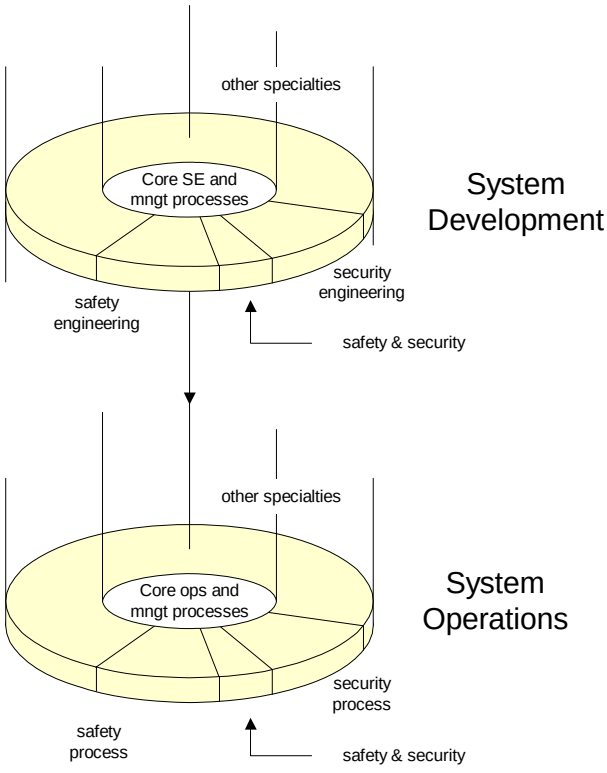


Figure 1 Security and safety processes viewed as specialist domains contributing to core systems engineering (SE), management and operations processes

systems engineering and management process, supported by many engineering specialties, including software, control, security and safety engineering (Figure 1). Broadly speaking, security engineering specifies constraints on system development and monitors the achievement of security properties at all levels of system decomposition. Components, tools and services specific to security (e.g. intrusion detection systems, software fault scanning tools) are important drivers of the integrated security performance of the systems in which they are deployed. Similar multi-specialist situations arise in operations. Activities undertaken to improve security consume resources and vary with respect to effectiveness and the efficiency with which they are conducted.

A need for assurance arises in situations where an agent feels exposed to risks arising from the actions of another party. Assurance provides a user (or a regulator acting on their behalf) with a basis upon which to place trust in the provided services i.e. with evidence that potential service failures are acceptable in terms of likelihood and severity. Providing assurance evidence also consumes resources.

1.3 Solution Strategy

It is recognized that security measurement is a challenging area and in its infancy, especially in terms of practice. The solution strategy adopted in this report combines three strands of development:

1. the measurement framework of PSM / ISO/IEC 15939, comprising a measurement development process based on *information needs*, *measurement constructs* and reference *measurement specifications*;
2. *risk management*, as implied by the adopted definition of security, following work in the dependability field;
3. a *systems-theoretic* approach to measurement design and management of system properties, in which measurement is viewed as embedded in decision-making loops. This approach is motivated by the uncertainty that prevails assessment of likelihood and severity of security failures.

The solution strategy seeks to bridge between management and technical responsibilities. The following six concepts are involved:

1. definitions of the subject system and of the delivered services in a use environment;
2. definitions of the failures associated with the subject services; association, in principle at least, of a frequency or likelihood and severity to each service failure;
3. a taxonomy of faults, internal and external to the system, judged to be involved in the service failures;
4. causal models that associate component properties and faults with service failures; the development of simplified fault trees that are sufficient to support particular decisions;
5. definitions of fault management techniques i.e. means to reduce, eliminate or contain faults during system development, and of policies to reduce failures during operations;
6. assurance evidence that enables a user to accept the security of a provider system and its services.

1.4 Related Work

Initial PSM efforts in the security field arose from work on safety measurement [4], important for defense and critical infrastructure systems, and were conducted independently of related security measurement work. More recently, links have been established with workers in the security measurement domain. Related security metrics work includes the NIST SP series of documents, particularly NIST SP 800 55 [5], current work in ISO/IEC JTC1/SC27 on an information security metrics standard ISO/IEC 27004 [6], and the work of the Measurement Working Group of ISSEA [7] in the frame of the Security Engineering Capability Maturity Model SSE-CMM [8] and the related standard ISO/IEC 21827 [9]. Other related work includes the security metrics recommendations of a CISWG study [10] that focuses on information security policy compliance and the DHS Software Assurance Program [11].

One of the initiatives of the DHS Software Assurance program is to coordinate and systematize the treatment of safety and security within the family of applicable ISO/IEC standards and related national standards. The standard ISO/IEC 15026, formerly on integrity levels, is under review as a standard for assurance, based on the concept of an *assurance argument*. The

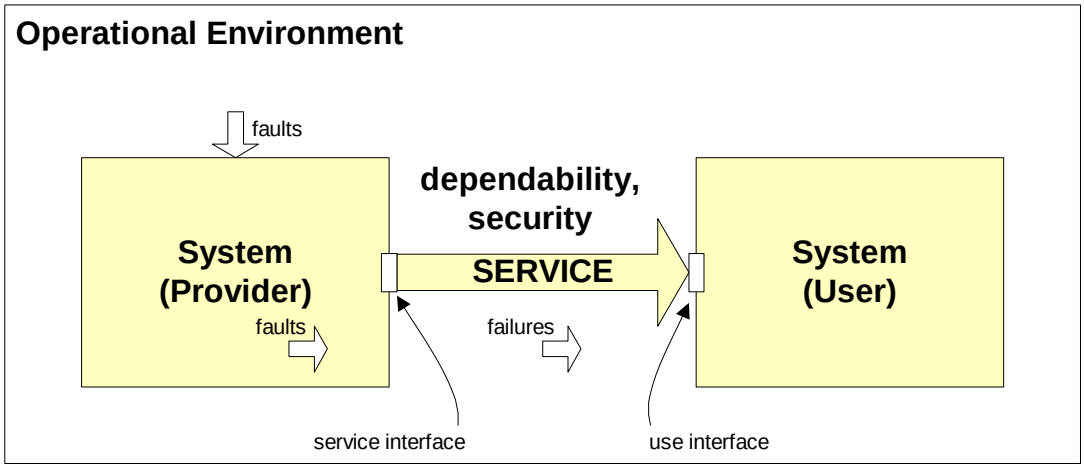


Figure 2 Context in which security is defined: a provider system, a user system and a provided service [13]

relationships between this standard, the framework standards for core management and technical processes (ISO/IEC 15288, 12207), and the generic standards for measurement [2] and risk [12] are evolving. Measurement principles play a key role in supporting and integrating engineering, management, risk management and assurance.

2 Fundamentals

2.1 Security and Dependability

Security is a property of a *system* or *service*. A *system* is an entity that has internal structure and interacts with other systems. We are interested in systems that are engineered; i.e. are developed and then operated to achieve some useful purpose. Software-intensive systems tend to be complex, meaning that they are composed of many components of different types which interact with each other to create properties not exhibited by the individual components. The purpose of the system is implemented as the *service* the system, acting as a provider, delivers to another system, the user system (Figure 2).

The user system is dependent on the provider system for the service. The delivered service usually will have many properties, depending on its type. Among these, the user system will be concerned about the *dependability* of the provider system, or, equivalently, of the provided service:

Definition:	**Dependability** (of a system delivering a service)
1	The ability to deliver a service that can justifiably be trusted. (calls for a justification of trust)
2	The ability to avoid service failures that are more frequent and more severe than is acceptable (implies criteria for deciding whether a service is dependable)
Source:	[13]

The second definition indicates a measurement approach to dependability, based on the likelihood and severity of service failures.

A particular service can fail in a variety of ways, resulting in dependability being a composite property, covering the following more specific properties (*more* of the property is indicative of *fewer* or *absence* of the corresponding failures):

Dependability Property of a System	Associated Types of Service Failure
Availability (readiness for correct service)	failures implied by the service being *incorrect*
Reliability	interruption or outage in correct service over a time interval
Safety	failures that cause catastrophic harm to users or the environment
Integrity	improper/unauthorized system alterations
Maintainability	service failures resulting from a system being difficult to successfully maintain during use

Like dependability, *security* is a composite property of a system or service, with different sub-properties being associated with different types of service failure:

Security Property of a System	Associated Types of Service Failure
Confidentiality	unauthorized disclosure of information
Integrity	improper/unauthorized system alterations
Availability (readiness for correct service)	types of failure implied by the term *correct*
Authenticity	A user not identified correctly – not who they claim to be
Non-repudiability	A neutral third party is unable to decide if a particular transaction or event did or did not occur

Definitions of security in the literature vary according to the types of failure that are of concern. The following are representative:

Definition:	**information security**
	preservation of confidentiality, integrity and availability of information; in addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved
Source:	ISO/IEC 17799:2005
Definition:	**security**
	Work that involves ensuring the confidentiality, integrity, and availability of systems, networks, and data through the planning, analysis, development, implementation, maintenance, and enhancement of information systems security programs, policies, procedures, and tools.
Source:	www.opm.gov/fedclass/text/GS-2200.htm

Dependability and security overlap in the sense that some types of failure fall under both properties. For convenience, security will be discussed as a single property in the following. It

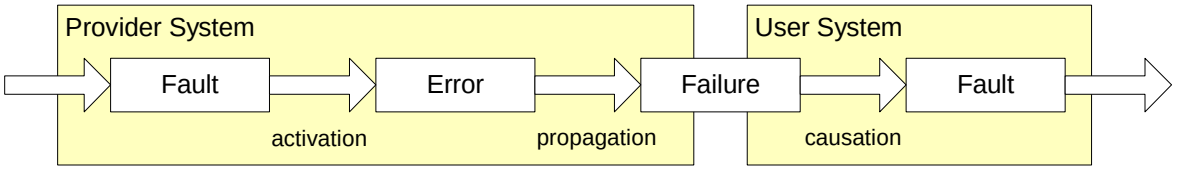


Figure 3 The propagation of the effects of a fault, from its activation to create an error in the system state, to propagation to a failure in a provided service, to causing a fault in a user system [13]. Called a 'fault path' in the text.

is understood that, for a particular system or service, dependability and security will be defined as some selection from the sub-properties, depending on the concerns of the user system.

The definition of dependability and security as the ability to avoid failures, raises the question of how a system or service can be measured with regard to such ability. Before addressing this question, we need to define a model of how a service failure is caused.

2.2 Faults and Errors

A service failure implies that the provider system's external states (i.e. those states observable by the user at the provider's service interface) deviate from the external states associated with the provision of a correct service. This deviation is called an *error*. The adjudged or hypothesized cause of an error is called a *fault*. Faults may be located within the provider system and/or in its environment.

A security vulnerability is a type of internal fault that enables an external fault to cause harm. An external fault may be the result of malicious actions of a threat agent. A system may have a property that is believed to remove or mitigate a fault or set of faults.

Figure 3 shows the relationship between the definitions of faults, errors and failures, adapted from [13]. For brevity, the chain of security threats represented by Figure 3 is called here a *fault path*. The recursive nature of this concept implies that the model can handle 'systems' failures, as developed below in terms of a systems theoretic model. The fault path is a simplified concept: a more detailed model might be in the form of a Fault Tree or Markov model. This is discussed further in Section 2.4.

The model is applicable to direct physical and logical cause-consequence chains, for example at component level. *Services* are usually viewed as *functions* in such cases. The model of Figure 3 can be applied to development processes; a fault in a component is the result of a failure in the service provided by its development process, which in turn, might be tracked back to a fault in the development system.

Figure 4 illustrates a fault path linking three domains: a system/service environment, a system of interest that provides a service; and a user system in which service failures may cause damage.

Faults can arise in any aspect of a system. Avizienis et al [13] provide an extensive typology of faults, along several dimensions. It is often a question of judgment as to the root cause of a failure, i.e. where a chain of dependability and security threats begins. For example, the

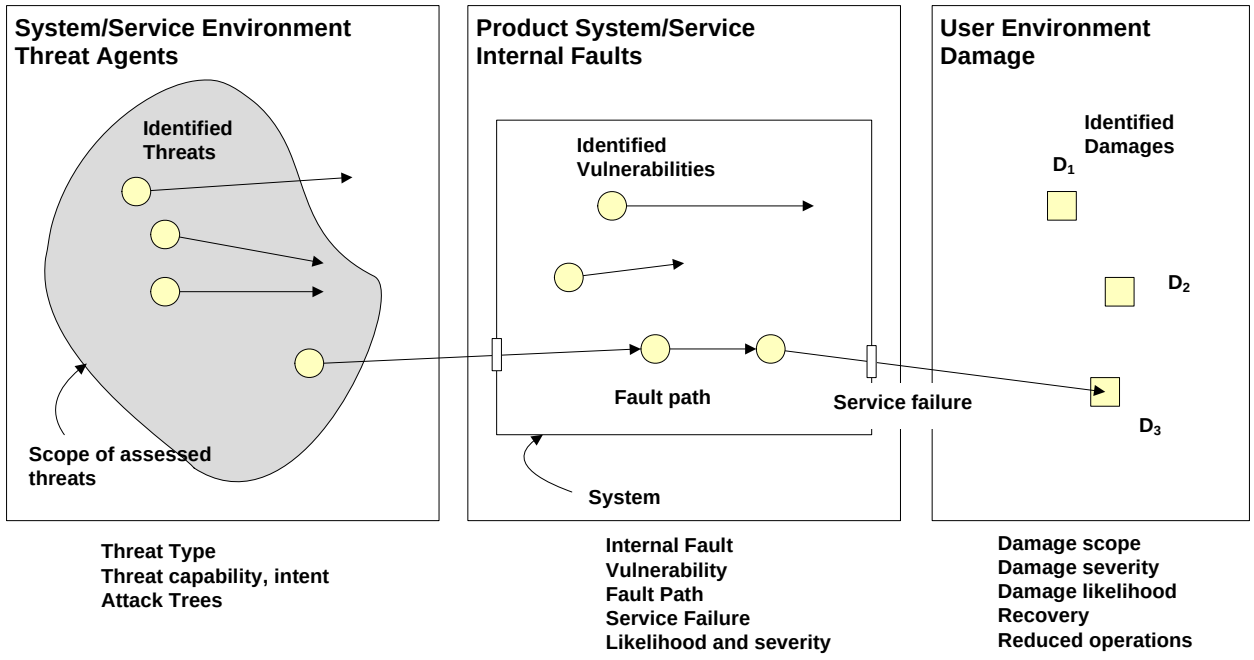


Figure 4 A fault path models the causal links between external threats (in this case), system faults, service failures and damages in the user environment. Such paths can be viewed (1) as hypothesis to guide design and operational policy, (2) as unfolding, in terms of security event management during operations and (3) retrospectively, as a basis for lessons learnt.

presence of a fault in a software component may be due to a failure in the software development process (viewed as a service provided by a project socio-technical system).

2.3 Threats

It follows from the definitions that security is a property of a system (and provided service) *in relation to* a threat environment. A given system may be acceptably secure in one environment, but not in another; or it may be acceptably secure today but not tomorrow.

Many types of fault of concern to security engineering are similar to safety faults: i.e. events in the natural environment, accidental, non-malicious actions during development etc. However, security has an additional type of fault arising from the presence of malicious threat agents in the operational and development environment. Such agents can learn and adapt, resulting in evolving external faults.

Attack Trees are used to map the objectives of a threat agent onto vulnerabilities of the system. Alternative attack sequences represent the possible ways the agent might achieve his/her goal. Development and operational policies can be adjusted to prioritize defensive actions.

Measurement can support the decision making involved, for example in the estimation of the cost to a threat agent of different attack sequences. Under certain assumptions, an increase in attack cost would imply a lowering of the likelihood of the attack sequence occurring and an increase in security with regard to the associated service failure.

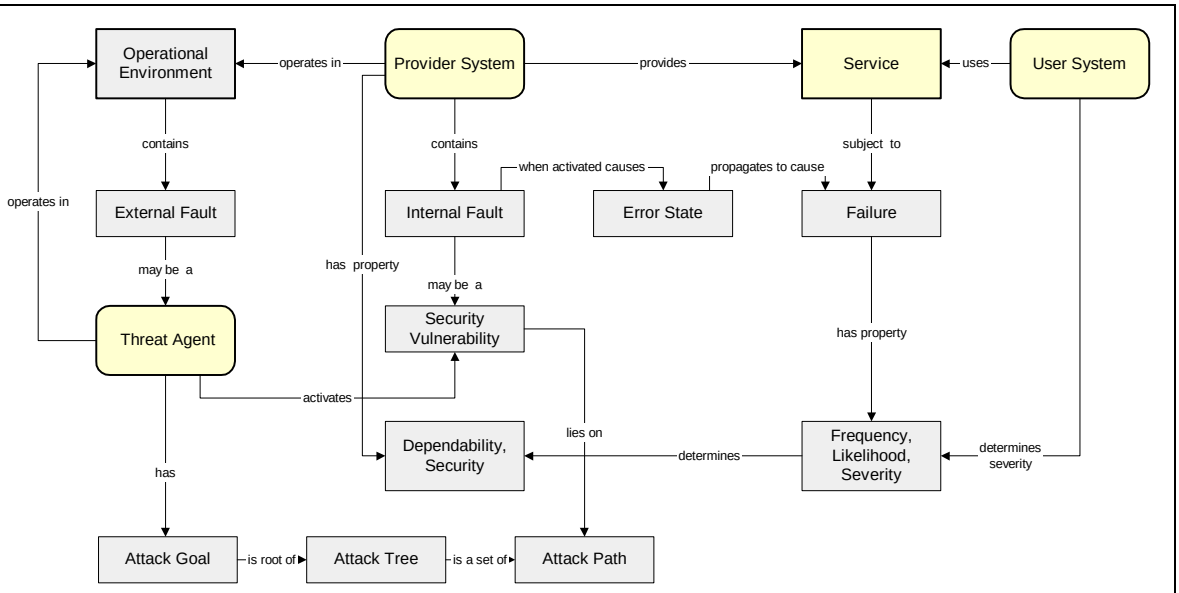


Figure 5 Concept model relating terms defined in the text

During system development, it seems necessary to establish a baseline threat environment to serve as a stable design objective. The assumptions made and the derived development policies have to be reviewed.

Figure 5 summarizes the relationships between the concepts defined so far.

2.4 Aggregating Security Properties

The principal challenge for security measurement is to develop a means for aggregating the assessed security properties of components and intermediate services, to provide an assessment of the overall security of a system. If this can be provided, we would have a basis for judging the effectiveness of security improvement actions and for performing trade-offs.

The classic model for aggregating the effects of faults in a system is the Fault Tree, developed originally in the nuclear safety field [14]. A service failure may be judged or hypothesized to be due to a combination of faults, described as an AND/OR tree. The links in a Fault Tree are causal links that are judged to be present in the provider system. For example, when the input conditions to an AND gate are true, the output event is held to follow. State-based models, such as Markov models, are used when the output event is dependent on the ordering or timing of input conditions. Here it is assumed that the results of such detailed analyses can be represented adequately in terms of links in Fault Trees. A given fault tree will have sets (called *cut sets*) of minimum numbers of base faults that cause the top failure to occur. The identification and likelihood of base fault events are developed by other analyses (e.g. Failure Modes and Effects Analysis).

An Attack Tree is effectively a Fault Tree developed by an attacker, based on their top-level objective and understanding of the system (Figure 6). A developer will hypothesize Attack Trees in order to develop defensive strategies.

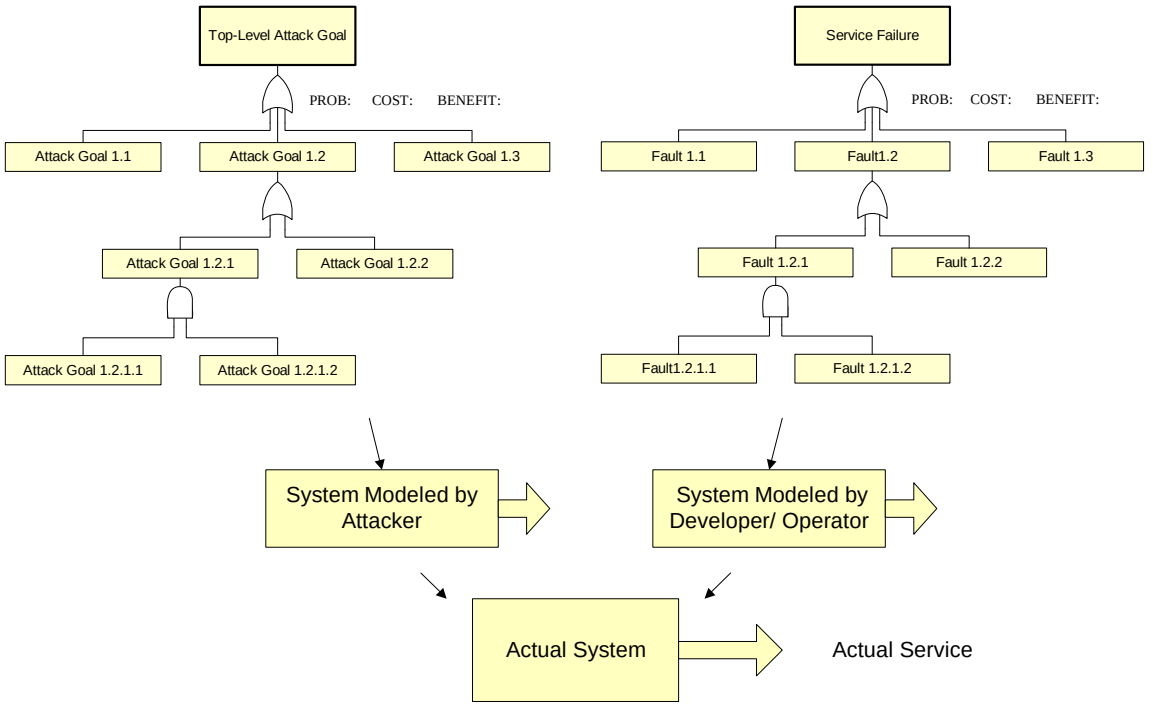


Figure 6 Fault Trees and Attack Trees for a system as interpreted by developers and attackers

The events and links in a Fault Tree refer to faults and failures in the components and intermediate services that make up the system, and to external faults where these are involved. The aggregation of security is sensitive to the architecture of the system. We have to develop an aggregation 'formula' for each system (or family, product line, program). Security is also sensitive to the security properties of the components involved. Achieving a property in a component may limit a type of failure.

Our interest here is to consider how to use the Fault Tree model to bridge between detailed technical assessments (which may generate complicated fault models) and technical management which typically needs summary information on which to base trade-offs etc.

Suppose, in a given decision situation, approximate Fault Tree models are developed to provide sufficiently accurate information to meet the decision need. Approximations are made, for example by setting the probability of conditions to 0 or 1 (Figure 7), effectively simplifying the tree structures. Such approximations can be reversible – if queried, or if underlying fault models change, the trees can be developed as needed. This is effectively a form of multi-pass analysis, in which as-simple-as-possible analyses and models are used.

The result of such simplification will be that a particular service failure is judged or hypothesized to be caused by a small set of prioritized fault paths (or simplified trees) through the provider system. Although approximations, such fault paths will be established to provide a sufficient means to aggregate security for the purposes of a particular decision situation.

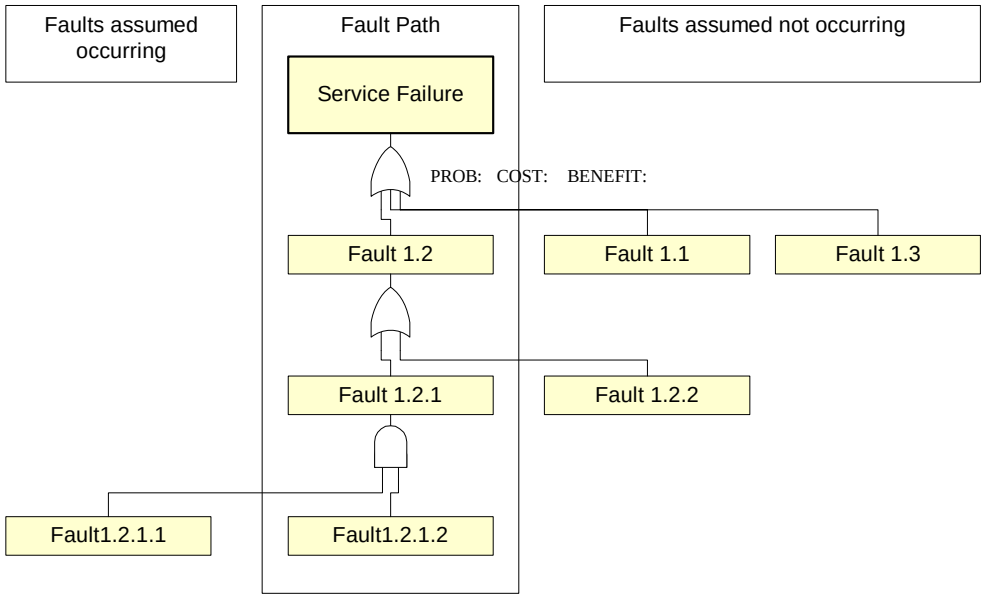


Figure 7 Example simplification of a Fault Tree to form a Fault Path

Security engineering and operations involve actions to eliminate or reduce the likelihood and/or severity associated with service failures. A single design or operational policy change may affect a set of fault paths and a set of potential service failures.

Figure 8 shows a simplified model of integrating security over a system. A particular service will be judged to be susceptible to some set of failures $\{F_i\}$, where each failure has a likelihood p_i of occurring in a specified time interval, with damage costs of sev_i. The system is operated in a use environment containing a set of threat agents $\{T_j\}$. Each agent is modeled as having a likelihood p_j of launching an attack, with a capability cap_j. The threat agents will influence the judged fault probabilities. A security intervention, whether in the design or operational phase, will have the objective of lowering one or more of the service failures $\{F_i\}$.

The aggregated security of the system, with respect to the provided service and the threat environment, will be judged on the basis of the set of identified failures $\{F_i\}$. Expectations of losses arising from failures that occur can be estimated, based on assumptions and estimates of likelihood and severity over a time interval. Establishing a basis for estimating probabilities and damages of failures, and the uncertainties in these, is a central task of security measurement. In many cases, there will be insufficient opportunity to establish probabilities based on observed frequencies. The difficulty of estimating damage costs varies between sectors, depending on the type of service being provided. An approach based on managing the uncertainty in estimates of risk and severity is required, similar to approaches advocated in the management of general project risk [15]. Where data is unavailable, it is rational to base initial assessments on subjective assessments of ranges of likelihood and severity. This amounts to a systematic approach to handling uncertainty that accommodates a transition to more objective analyses when data becomes available.

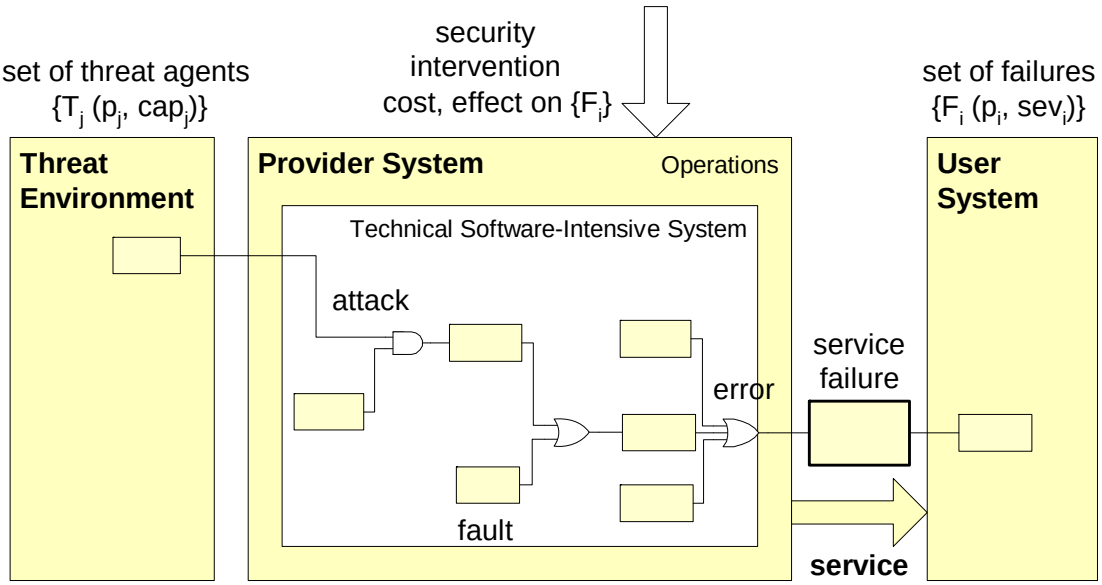


Figure 8 Simple model of aggregation of security properties; fault paths are central to the integration of component properties and association with service failures

Fault Trees are used traditionally in engineering domains in which the causal links represent physical or built-in logical processes. It seems possible to apply Fault Trees to systems in which there are intermediate agents i.e. where the causal links are mediated by learning agents (Figure 9). This approach is more appropriate for socio-technical systems in which faults are introduced by organizational processes, for example. The probability of a fault condition may then be managed, with the agent taking corrective actions based on local observation and measurement. This model supports consideration of the time constants involved in responsive loops and related issues.

2.5 System and Component Properties

A consideration of potential failures is intrinsic to the specification, design and build of any system. A requirement for a function or property can be viewed as a statement about the unacceptability of not achieving it. In this sense, the basic design of a system provides the platform for achieving security properties. A particular property of the system, if successfully achieved, may effectively remove the possibility of a set of potential failures. A property of a component of a system may interrupt or mitigate fault paths on which it lies. In such situations, demonstrating that the property holds becomes a means for demonstrating acceptably low risk for the associated failures.

Specifying and achieving properties of components and intermediate services are the basic means of building security in to a system. Constraints on system development are expressed through requirements specifications and development policies. For example, in Figure 8 a design intervention will be implemented in the belief that a beneficial effect on the failure set $\{F_i\}$ will result.

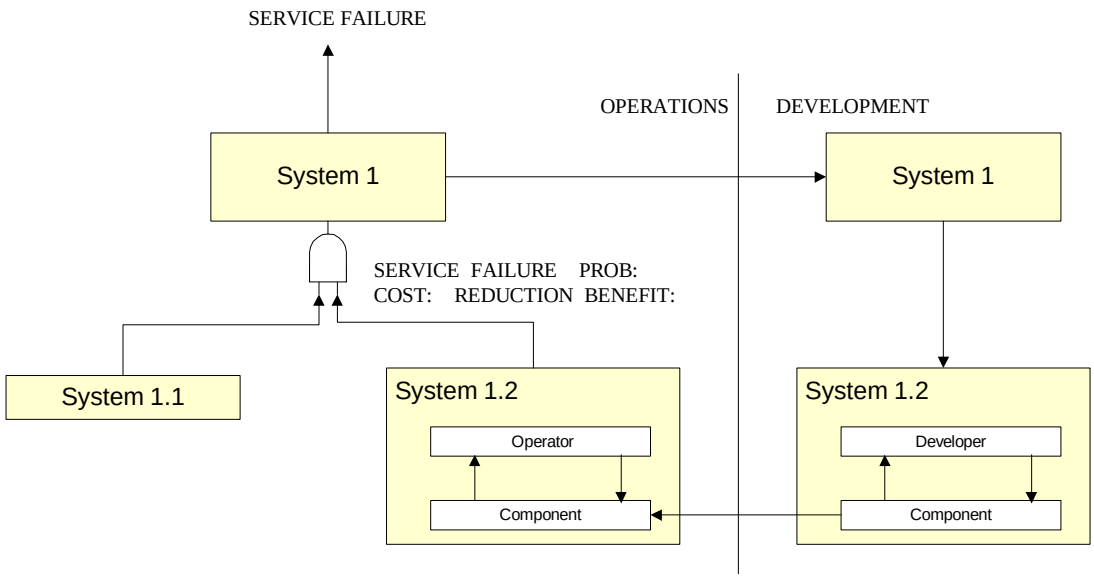


Figure 9 Fault Tree with agent-based components

Achievement of component properties may not eliminate faults arising from the integration of the components into a complete system. Policies and constraints are needed at different hierarchical levels in a product or process. Placing constraints on component design and development can be viewed as a means of engineering emergent properties at the aggregate level.

Relationships between properties, faults and failures begin in the hypothesized design domain. During development, the realization of specified properties in the product is a central concern. Measurement can play a part in showing compliance with specifications, i.e. to support verification. There remains the need for validation: a product may exhibit a specified property, but the property has to be established as sufficient in the use environment.

2.6 Trust and Assurance

The user system of Figure 2 is *dependent upon* the provided service to some level, determined by the criticality of the service to his/her operations. To benefit from the service, the user must be prepared to place some level of trust in the provider system. Following [13], trust is defined as *accepted dependence*. The user's criteria for trusting the provider system are expressed in the terms of dependability and security; i.e. failures in the provided service have to occur with acceptable frequency and severity. How does a user establish trust in (or assess the dependability of) a provider system? The user requires some basis on which to assess the dependability of a product or service; this basis is called *assurance*.

For most complex systems, a user requires an *assessment* of the dependability of a system, before deciding to accept the risks involved in using it. Suppose a product is delivered to the user (similar arguments apply to services) with a *true* dependability i.e. a set of failures with probabilities and severities that are accurate, objective descriptions of the future statistical behavior of the product. A user will wish to have a description as close as possible to this true description.

The acceptability of a potential service failure to a user (the user's risk tolerance) will depend on:

1. the likelihood;
2. the potential damage suffered by the user; this can take many forms; equivalent dollar costs can usually be derived e.g. for disrupted operations and recovery. Financial losses can be difficult to assess for some failures, particularly catastrophic ones;
3. the cost of further reducing the failure likelihood and severity;
4. the benefits to the user from having the associated primary service.

The concept of acceptability of service failures is important from a measurement perspective, because it provides a point of reference. It also highlights the subjective (or role-sensitive) nature of security; a security risk that is acceptable to one party may not be to another. Often, different agents are subjected to the risk of a service, benefit from the service as a user or provider and carry the costs of risk reduction.

There will be uncertainty in any practical assessment, no matter who performs it. Suppose we distinguish between technical uncertainty and behavioral uncertainty. Technical uncertainty arises from limitations in technical and management methods where these are applied professionally and diligently. Behavioral uncertainty arises from agents, for financial or other reasons, not striving to assess the true dependability. Behavioral uncertainty arises typically when one agent can benefit financially from transferring risk covertly to another agent.

There are three basic ways of providing assurance in a product [16]:

1. Quality of the people involved in development;
2. Quality of the development processes employed;
3. Direct assessment of the product through analysis and testing.

The last of these provides the strongest evidence, but is costly for complex systems. The provider is usually best placed to perform such analyses, from a technical point of view. A user will often not have the expertise or resources to perform an independent assessment, but will have to place trust in the provided assessment. There are two related levels of trust involved - in *the system* and in *claims about the system*; both have to be addressed. They are not the same, since there is sometimes a trade-off between investing in the engineering of a system versus investing in an assessment of it. Gelen and [17] have modeled the situation in terms of expectation and variance in assessed performance.

At component level, there may be markets of many potential users for single products. There are efficiency gains from sharing assessment costs; products can be assessed by a governmental or neutral third party, relieving users from having to perform independent assessments. There is a long history in the security field that addresses such strategies [16] [18] for security-specific components. However, there are practical problems, for example in carrying out assessments sufficiently quickly in highly time-sensitive commercial markets and in the maturity of products submitted for evaluation.

Currently, the concept of an *assurance argument* is being developed as a means to integrate the evidence of dependability of complex systems [19].

A quantitative approach to uncertainty in risk assessments and assurance would improve decision-making [15]. We wish to be in a position where trade-offs can be performed between investment in different forms of assurance; between phases and activities in the SDLC when evidence is gathered; and between actions to improve expected product behavior and actions to reduce uncertainties in expected behavior. The costs of assurance processes, including regulatory, audit and product evaluation processes are considerations in decision situations.

3 Systems Theoretic Model

A wide range of technical and management activities are directed at managing faults and failures. The application of measurement principles has to address this diversity in an integrated way. An approach to measurement is proposed that associates measurement with 'closing the loop' between decision-makers and the effects of actions they undertake. The systems-theoretic approach has been developed in the safety domain [22]. Two cases are discussed here: measurement of delivered services and measurement during development.

3.1 Measurement of Delivered Services

The fault propagation model of Figure 3 illustrates the transition from a fault in a provider system to an error state in the service interface, to a failure in the provided service. The service failure then causes a fault in the user system, and so on. This model represents causality by a linear cause-consequence chain. This is often an appropriate model for the propagation of fault effects between components in traditional (mechanical, electro-mechanical etc) systems and for modeling faults and effects in sequential processes.

One application of measurement may be viewed as providing feedback from the service, as experienced by the user system, to the provider system (Figure 10). The advantage of this approach is that the feedback represents the difference between service as perceived by the 'customer' system and the user expectations. The effectiveness of actions taken by the provider system has to be validated in some way by assessment of the effects on the provided service. The disadvantage of user system measurements is that they are reactive; measurements are made after delivery of the service. This is acceptable in many situations (for example, when failures are tolerable, time constants are acceptable and correction costs are tolerable).

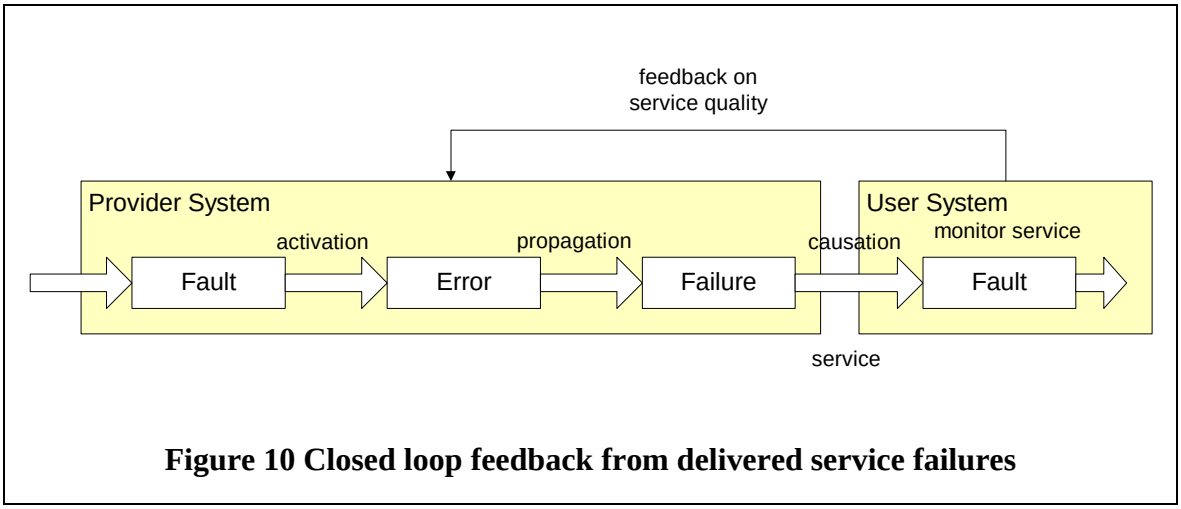


Figure 10 Closed loop feedback from delivered service failures

3.2 Measurement during Development

Usually a provider system will wish also to act 'proactively' to improve the service before delivery. Such actions can be informed by measurements within the provider system, as well as feedback from the user system. Measurement can be viewed as providing feedback from the service-providing processes to a 'higher level' of control (Figure 11). For example, during design or policy implementation, a requirement or constraint can be placed on a process with the objective of reducing the likelihood of introducing faults. In system and software development, a component may be required to have a specified property, for the purpose of eliminating or mitigating particular types of fault or error. This generates a 'vertical', layered model representing traditional hierarchical control; actions are placed on the development process and are monitored for compliance. Measurement can be viewed as a component of a control loop bridging between management 'layers'.

Policy constraints on product design and development are treated similarly to policy constraints on operations; both are directed at mitigating faults and propagation mechanisms. A developer seeks to demonstrate that a specification is met (a property is achieved in the product as-designed and as-implemented). An operator seeks to demonstrate that operational policy requirements are complied with.

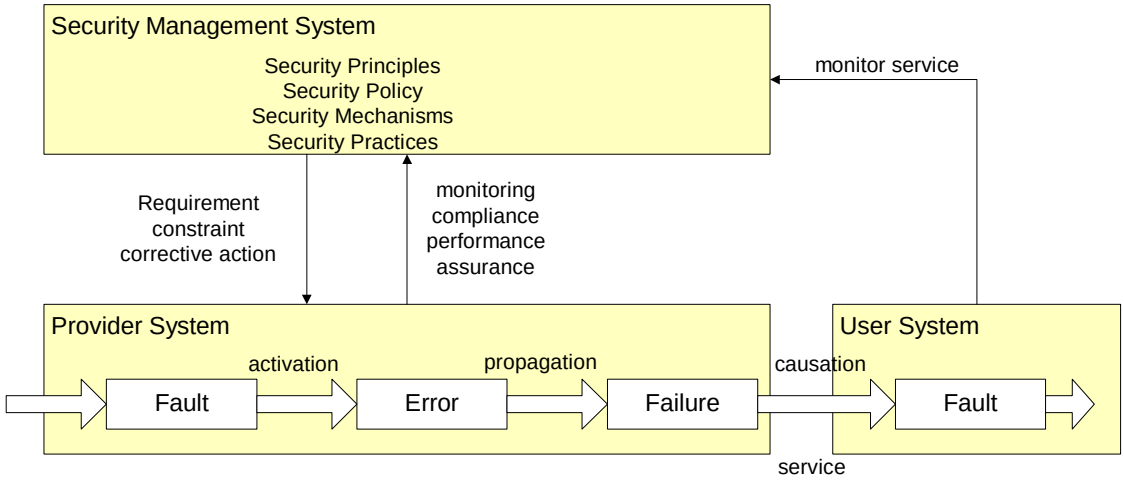


Figure 11 Closed loop feedback based on concurrent monitoring and adjustment

This approach is consistent with the layered organizational model as used in PSM (Section 5).

3.3 Establishing Benefits of Security Interventions

It is recognized that causal connections cannot always be established between 'local' engineering and operations activities and delivered service security. The use of Fault Trees to integrate security risks over a system is a response to the observation that there is no *general* basis for aggregating security: each system has a different architecture and sets of fault paths. The basis for security aggregation is developed as the architecture is developed and as the system become better understood. The systems theoretic model for measurement seeks to establish a basis on which the required learning can occur; a learning loop (c.f. the PDCA cycle as applied in [20])

seems to be required. A local intervention can be monitored in terms of local objectives and outcomes. The implication for the end service, although initially a matter of judgment, can over time, be increasingly quantified.

The concept of a *decision space* has been found useful in the domain of engineering and project management [21], particularly in addressing issues of risk and uncertainty. The system theoretic model is compatible with this view; it is not assumed that sufficient measurement information is always available for decision-making. Instead, the model aims to provide a framework for improving the information basis for decisions.

3.4 Systems Approach to Assurance

It is speculated that the provision of assurance can also be viewed in system theoretic terms. Suppose a user wishes to increase confidence in the service delivered by a provider. An assurance 'control loop' might be established to bridge between the provider and user; the user is seeking to act proactively by assessing the performance of the provider's processes. The concept of an *assurance system* is implied: the provision of acceptable assurance is an 'emergent' outcome. A similar approach might be considered for establishing compliance with regulations – usually a matter of negotiation and interpretation. A user's concerns are limited to those areas that expose him to particular types of risk. The core system development and operations processes have to trade-off interests of all parties.

3.5 Optimistic and Pessimistic Views

Where the management loop contains a human decision-maker, both a positive/optimistic and a negative/pessimistic view can be taken. The positive view assumes that the actions being taken are essentially correct and effective; measurements are made to confirm the benefits, amounting to verification in terms of the design intent. The negative view is skeptical of all design solutions; measurements are made to uncover faults, assess risks and performance independently of assumptions made during development. The skeptical view amounts to an emphasis on validation and can be viewed as a secondary control loop, assessing and reflecting on the actions of the primary optimistic loop. This concept is captured by the idea of double-loop learning, used in the systems dynamics field, for example [].

4 Representative Security Practices

Security practices have the general objectives of establishing and preserving acceptable security properties of systems and services, managing security risks, providing assurance and demonstrating compliance with applicable regulations and standards. During development, security engineering practices are enacted to achieve security goals expressed as requirements placed on products. During operations, security operations practices are enacted to achieve security goals expressed as performance goals.

Security practices are briefly reviewed under the following headings:

- Security principles and policies
- Types of product and service
- Security engineering
- Operations
- Risk Assessment
- Evaluation and Test

4.1 Security Principles and Policies

The security field is a large one – information security is perhaps the most general term (to which might be added control system security). The fields of computer security, network security and software security are more specialized areas of professional engineering practice. Each has more specialized areas of expertise. System security engineering addresses the concerns from the viewpoint of software-intensive systems, compatible with systems engineering as defined by ISO/IEC 15288 and related standards. The overlap with system safety engineering has been addressed in recent years [23].

The long history of computer security has established several principles that are used to guide the architectural design and operation of secure computer-based systems. They can be viewed as being expressed through design policies and requirements and include [16]:

1. Accountability;
2. Least privilege;
3. Minimize the variety, size and complexity of trusted components;
4. Secure default configurations;
5. Defense in depth.

Such principles guide strategic design choices that reduce the likelihood of common types of service failure. Security principles are implemented using a selection of security mechanisms, for example [16]:

1. Defining and implementing domains, i.e. areas of stored data and applications with restricted access;
2. Linking users with domains;
3. Authorizing operations;
4. Auditing operations;
5. Cryptography.

Security mechanisms are implemented by a range of security components (i.e. components whose primary functions are security-related) forming the security architecture of the system, and operations policies. Systems and software security engineering specialties are responsible for specifying, designing and implementing these systems, and for supporting general systems and software engineering functions in realizing the security properties of the total system product.

Measurements can be developed to (1) assess the degree to which an implemented and operated system meets the design intent and (2) the degree to which the design intent, as implemented, meets the needs of users.

4.2 Types of Product and Service

Systems and services can be of many different types. An important difference is between purely technical (engineered) systems, and socio-technical (organizational) systems. Both types are of interest in this study: software-intensive, technical systems and the organizational systems which develop and operate them.

Three situations typically arise in security measurement (Figure 12):

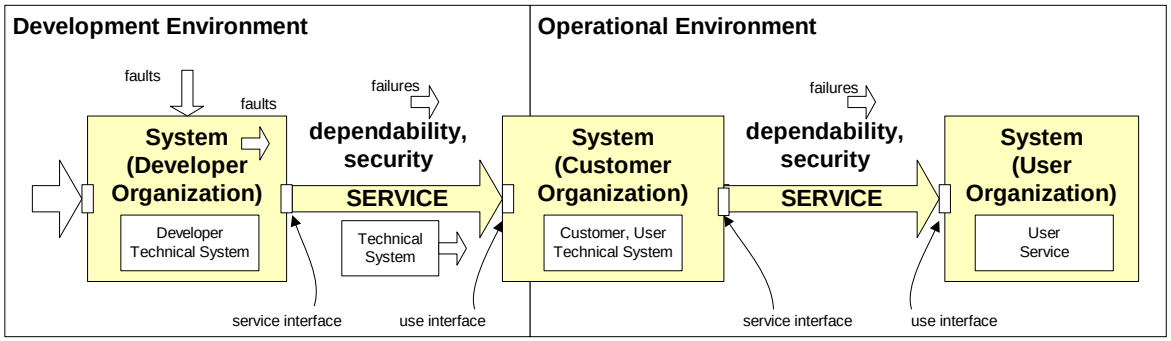


Figure 12 Organization model comprising a developer, operator and user of a system

1. a developer system develops an engineered (technical) system product for a customer, which is intended for use by a provider system;
2. a provider system makes use of an engineered (technical) system product to provide a service to a user system;
3. a user of the provided services.

The engineered product may provide:

1. a service (e.g. information storage, transaction processing) which is *not* primarily a security service, but for which security is a property;
2. a service (e.g. intruder detection, fault finding), which is primarily a security service. In this case, security would also be a property in the sense of (1).

A secure system will be developed in accordance with security principles as expressed by a development policy, an architectural design, security mechanisms and components. It follows that security measurement has to be considered over an enormous range of types of technology and component, for example:

- Software at code level, bit/register level;
- Software module, object
- Software application
- Software system, architecture;
- Hardware component, particular technology/ physical principles;
- System, an aggregation of software and hardware components (single, monolithic entity);
- Networked system, where communication links and nodes lie within protected Environments;
- Systems of systems, i.e. systems that are developed to independent goals, but are required to inter-operate;
- Systems with specific prime function; information processing, command & control, embedded real-time control etc;
- System or component with a prime function to mitigate security risk;

- Internet technology component or system, where communication links and nodes are provided by many other parties;
- Grid systems;
- Mobile/ ubiquitous systems;
- System in which safety and security properties are inter-dependent;
- Organization with information security policies;
- Development system (infrastructure used to design, develop, manufacture and operate security-critical components and systems).

Examples of Items of Concern

Figure 13 shows one categorization of systems.

The design of measurements depends on the identification of entities involved in system development. These will vary across the different types and scales of entity. It is useful to develop a set of representative component/ system types, such as those sketched in Figure 14. Such templates help specifying measurement constructs, the assumptions made in their development and the tailoring of them to particular situations. The security architecture of a system may lead to the identification of measurement, for example associated with domains, perimeters and ports. Specified components will also imply potential measurements, associated with functions and potential failures.

Notations vary between areas of security practice. The system architecture notation recommended in the DoDAF standard [24] might be appropriate for defense systems. Network security could make use of the notations of the CISCO SAFE methodology [25], for example. Software security could make use of standard software architecture and design notations. The threat modeling approach to software application development reported in [26] makes use of classic data flow diagrams.

The concept of a *local* operational (threat) environment seems to be important, because it represents a distribution of risk mitigation between the system of concern and other, external systems. This enables modeling of the distribution of risk acceptance over complete systems. Defense-in-depth involves *as-designed* distributions of risk reduction, complemented with maintenance and adaptation during operations.

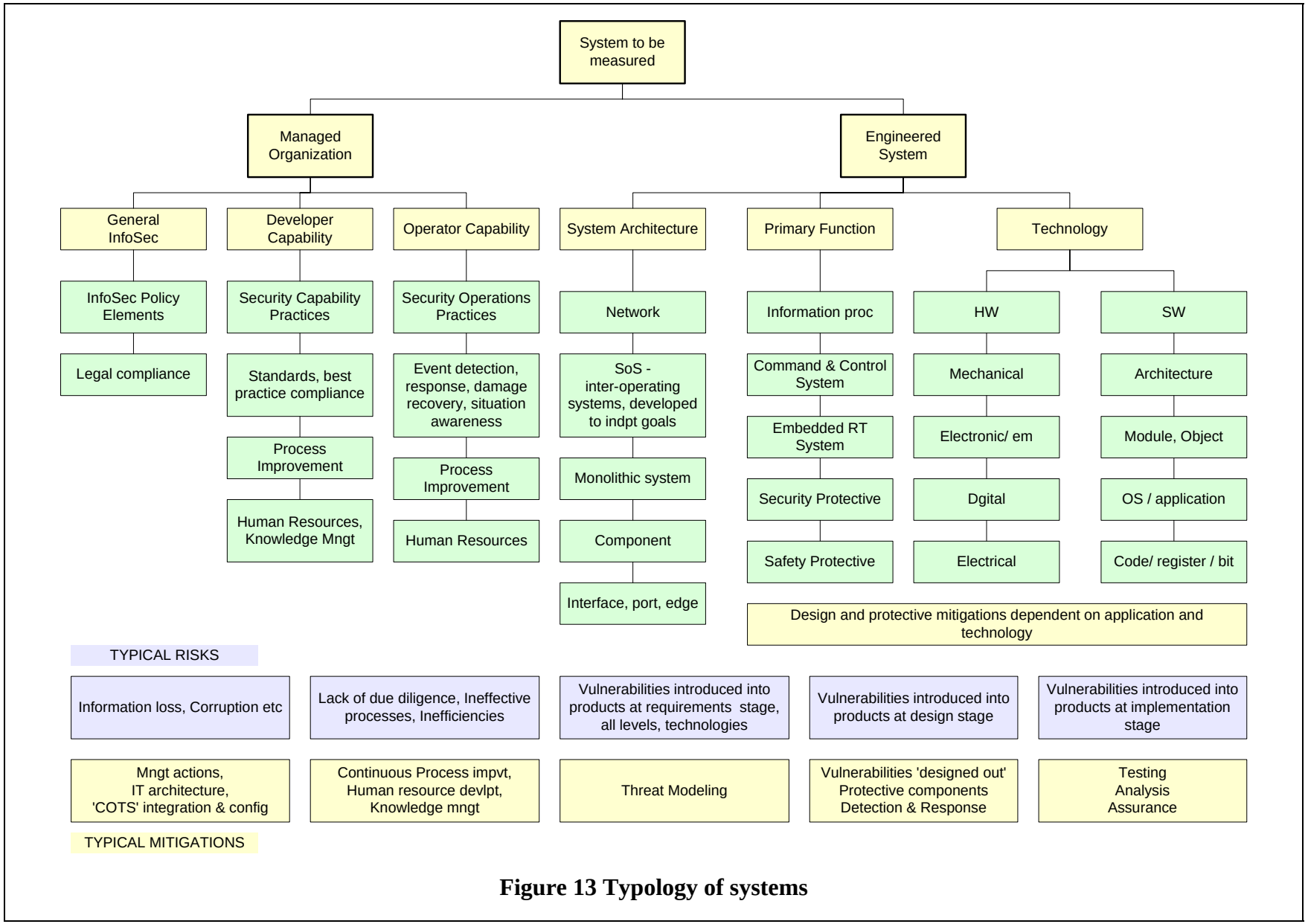
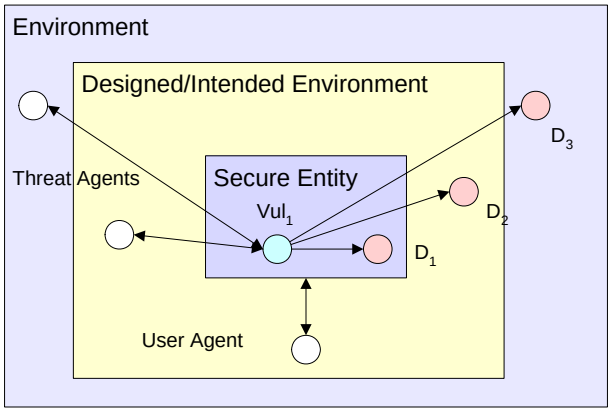
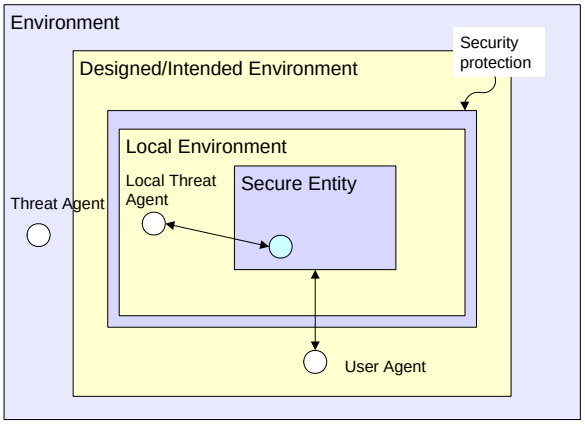


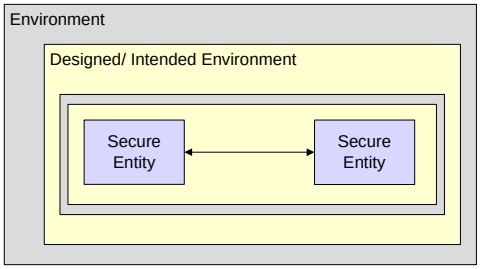
Figure 13 Typology of systems



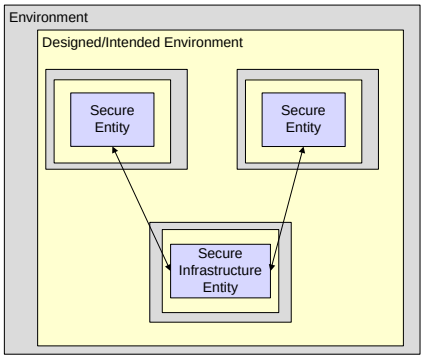
Single entity, sw, hw, mixed



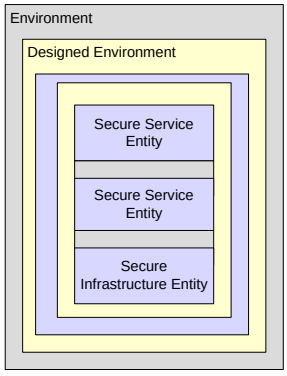
Secure entity with protection



Networked entities within a protected Environment; trusted paths, integrity



Networked entities with access from unprotected Environment



Layered entities e.g. ports

e.g. NIST models [27]
DoDAF models [24]
SW models [26]
Network models [25]

Architecture models

Figure 14 Template system architectures to support security management (sketches only)

4.3 Security Engineering

Core technical and management processes are responsible for the integrated development of a system through a System Development Life Cycle (SDLC) (Figure 15). Security engineering typically carries responsibility for assessing security requirements and properties of the system, for recommending security system design, monitoring progress and developing security assurance. The 22 practice areas of the SSE-CMM [9] provide a reference model for security engineering (Appendix 3). The safety and security applications areas extensions to the iCMM/CMMI [23], developed by DoD and FAA, are also useful starting points (Appendix 3). A report from NIST SP 800-55 [5] provides further measurement recommendations. The Measurement Working Group of ISSEA [7] is addressing measurement issues in relation to the SSE-CMM model and incorporating the earlier NIST work.

Security engineering can be viewed as a concurrent specialty process enacted concurrently with the core development processes (Figure 1). Figure 16 shows a model of the interaction between such processes, at system level. Similar process steps are used at application software level [26]:

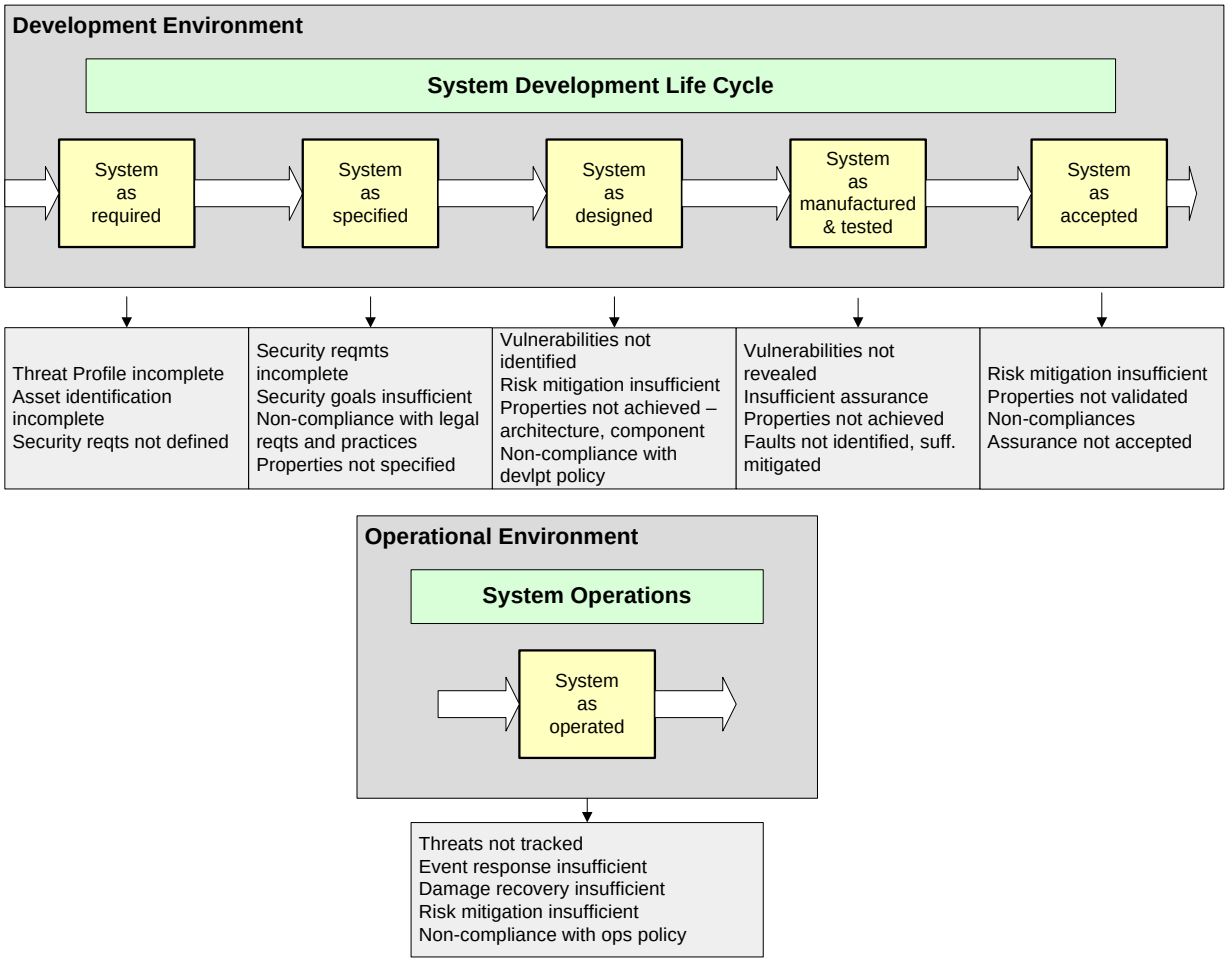


Figure 15 Linear view of development and operations processes with example sources of security risks

1. Gather security requirements, including threat analysis;
2. Secure design (architecture, components);
3. Model threats; analyze for vulnerabilities;
4. Perform implementation-level analyses (e.g. code reviews and static analysis);
5. Apply formal methods where appropriate; develop formal proofs – verification that code implements specifications;
6. Perform tests of security functionality and penetration tests;
7. Secure deployment;
8. Integrate feedback.

Unfortunately, the development of all components (hardware, software, security-specific, primary function) is subject to the introduction of faults. Faults are defined with respect to:

1. A specification; required product properties (e.g. not being able to bypass security functionality);
2. good engineering practice (e.g. avoidance of specified programming constructs);
3. faults in the product type, known to the industry (e.g. buffer overflow);
4. certification/ regulatory conditions (e.g. FAA requirements in the flight safety domain);
5. faults known to the developer in terms of the product family, product line or program (e.g. lessons learnt from previous product versions).

Improving the general quality of product development contributes to security. Reducing the number of faults introduced and improving subsequent detection and correction, are forms of general risk management.

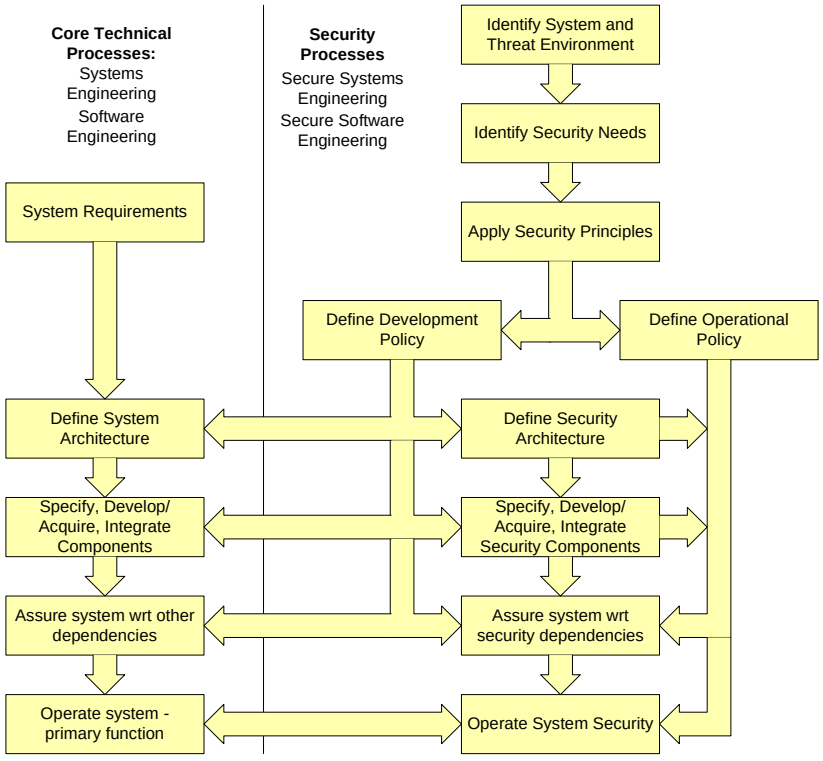


Figure 16 Model showing the application of security principles, expressed via security policies, to a system development project

In software development, for example, a combination of fault-related strategies are adopted:

1. improve general product quality; reduction of faults introduced at each stage of development (lessons learnt from general software engineering practice);
2. check for faults typically implicated in security vulnerabilities, based on accumulated experience in the type of product being developed (lessons learnt from security engineering practice within the industrial sector);
3. check for faults against particular attack goals and trees, using security risk assessment conducted for the particular product under development and a specified threat environment (lessons learnt by the developer within the program product type/family).

The design and implementation of a security-specific function is a further responsibility of the security process. The actions involved in mitigating security risks vary enormously over the types of system, threat and vulnerability involved. Mitigation actions may be categorized as follows:

1. reduce the likelihood of an attack attempt, for example by seeking to modify motivation and access;
2. reduce likelihood of a successful attack by design means (e.g. security architecture, functions, components);
3. reduce likelihood of a successful attack by operational means (e.g. detection of attack attempts and defensive response);
4. reduce effects of a successful attack, through design and/or operational means;
5. improve damage repair and recovery following a successful attack.

The set of implemented prevention and mitigation actions (as part of a security policy or development process) can be tracked and monitored as for other tasks. Schedule, cost and progress measures can be developed for mitigation actions.

4.4 Operations

Policies are used to manage the implementation of security principles during operations. For example, the CSIWG study [10] provides a set of practices and measures applicable to information security operations, emphasizing compliance with policy. Measurements can be developed to monitor compliance and to assess the effectiveness of a policy, as implemented.

Operational policies are usually decomposed into sub-policies applicable to particular services, components and ‘security controls’, for example [25]:

1. physical security policies; 2. access control policies; a. password properties, change properties etc 3. dialup and analog policies; a. modem response b. one-time passwords c. traffic monitoring d. fax line use e. password storing f. strong authentication 4. remote access policies; a. T1 b. Frame relay c. VPN access 5. remote configuration policies; a. secure sockets layer b. secure shell 6. VPN and encryption policies; a. User management	b. Time length control c. Encryption standard 7. network policies; a. router policy b. firewall policy c. DMZ policy d. Extranet policy e. www policy f. wireless policy g. server policy 8. data sensitivity, retention and ethics policies; 9. software policies; a. operating system policy b. virus protection policy c. user software policy i. installation policy ii. database policy iii. e-mail policy.
Example elements of an operational security policy [25] – not complete	

4.5 Risk Assessment

Given that security is defined in risk terms (likelihood and severity of failures), all security-related activity can be regarded as forms of risk reduction. Strengthening engineering practices reduces the risk of introducing faults during development. Engineering a security architecture and security-specific components reduces the risk of failures due to common types of faults. Assurance activity reduces uncertainty about risks, especially for users and others not directly involved in development.

Two specific risk assessment activities are important in the security field: threat modeling and vulnerability assessment.

4.5.1 Modeling Threat Agents

Factors involved in assessing the security risk posed by a particular agent have been modeled by [28] (Figure 17). These factors can be assessed on the basis of qualitative scales, enabling risks to be prioritized. For example, the threat capability of a group of terrorist threat agents might be assessed on the basis of [28]:

1. Group size;
2. Level of education;
3. Cultural factors;
4. Access to communications and the Internet;
5. Technical expertise;
6. History of activity;
7. Sponsoring countries;
8. Funding.

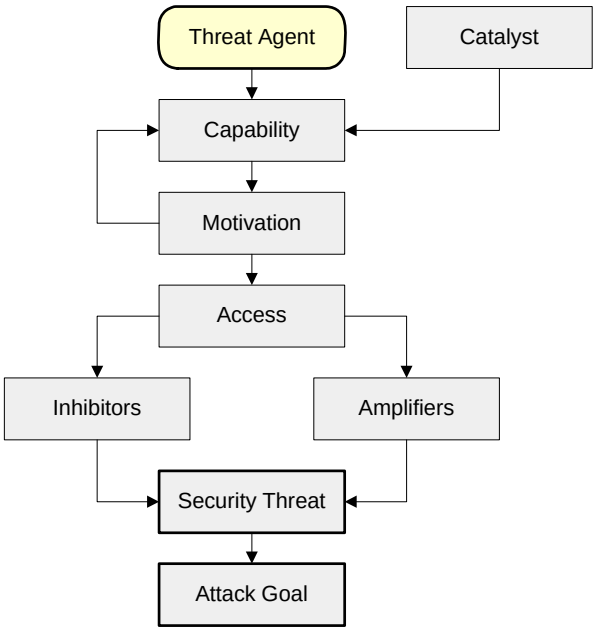


Figure 17 Aspects of a threat agent, from [28]

The level of threat (potential to cause damage) of a threat agent is also influenced by their motivation and opportunities to access the system, among other factors.

Attack trees model the particular attack goals and the options for achieving them in relation to the attacked system. A top-level goal (Figure 18) is decomposed into sub-goals in an AND/OR tree. The path from a leaf node to the top-level root is an *attack path*. The set of all identified threats to a system from a particular threat agent, is the agent's *threat profile*.

Attack trees may be used to integrate quantified assessments of the costs to the attacker in achieving the goal at each node. Alternatively, a probability of success may be associated with each node, based on judgments about the threat presented by the agent and the protection presented by the system. If probabilities could be assigned to nodes, the likelihood of a successful attack could be assessed from the probabilities along the complete network of potential attack paths. The security risk associated with the attack is assessed from the costs associated with the effects of the successful attack.

In addition to the probability and cost aspects, measurements can also be based on tracking identified threats and attack paths (as in a project risk register); the number of threats (top level goals) and attack paths, under selected categories, can be tracked over time. Time and costs associated with mitigation actions can be tracked.

The particular form of attack goals and sub-goal strategies will depend on the target system and assets. For example, threat effects have been classified as follows in the development of secure application software (not a complete list) [29]:

1. Spoofing;
2. Tampering;

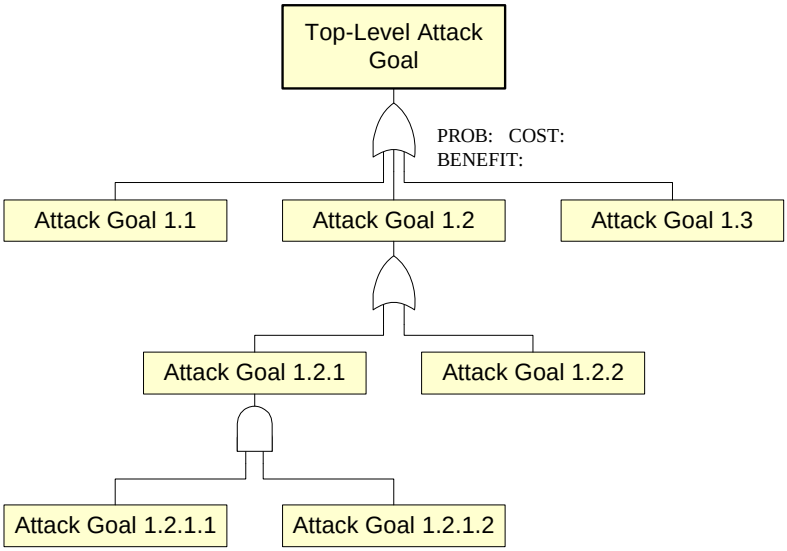


Figure 18 Attack Tree

3. Repudiation;
4. Information disclosure;
5. Denial of service;
6. Elevation of privilege.

4.5.2 Vulnerability Assessment

Assessing system and software designs and implementations for potential vulnerabilities complements the threat-driven approach. For software, vulnerability scanning tools (e.g. from Fortify Software and Ounce Labs) are available today to assist with the detection of defects commonly associated with security events.

The tracking of potentially exploitable defects and vulnerabilities enables the measurement of numbers of these over time, in different type and status categories. In practice false positives can be a severe problem, particularly for legacy code.

4.6 Evaluation, Testing

The Common Criteria (CC), established as an international standard ISO/IEC 15408, provide a framework for the independent evaluation of the security properties of IT component products. The evaluation process involves:

1. the identification of a baseline set of security objectives, constituting a *Security Target* (ST), against which a product is to be evaluated;
2. the optional use of standard *Protection Profiles* (PP), representing typical sets of security requirements; association of the ST with the PPs it satisfies;
3. the product to be evaluated - the *Target of Evaluation (*TOE);
4. the evaluation of the TOE against the ST and therefore PPs;
5. several evaluation levels (EAL 1 through EAL 7), providing different levels of evaluation rigor, and therefore confidence in the performance.

A developer of a secure system can assess requirements against the PPs, identify which products have been evaluated against them (via the applicable ST), and at what assurance level. The suitability of the component can then be assessed. Like the Orange Book framework that preceded it, the Common Criteria provide a sound basis for component evaluation, in principle, but there seem to be complications in practice.

5 Information Needs Model

Measurement systems are deployed to serve the information needs of decision-makers. There are many different roles and types of decision involved in the development and operation of secure systems. An investigation of typical information needs is based on a model of typical roles and decisions.

An organization can be modeled as several management layers, following the PSM model [3]. For example, a system development organization is assumed to be layered as follows (Figure 19):

1. Enterprise management: development of the enterprise in its legal, market and financial environments; governance;
2. Organization management: development of the organization's capabilities; management of resources; program, product line management;
3. Project management: development of a single product or service;
4. Technical/ professional specialty work: core work involved in system development.

A 'layer' is a generic role, representing a collection of typical information needs. The allocation of roles and responsibilities varies between organizations, resulting in different groupings of information needs. However, the basic responsibilities of Figure 19 will be recognizable to most product development, project-orientated organizations.

Project management is replaced with *operations management* for operational organizations (a similar three-layer model is used in [10]).

At the technical level, several roles are distinguished as follows:

- Systems engineering, secure systems engineering;
- Software engineering, secure software engineering;
- Specialty engineering, secure component engineering.

More detailed information needs can be developed from role/responsibility models, matched to local organizational practices. For example, the following roles might be involved in software application security:

1. security requirements developer, threat analyst;
2. software architect;
3. developer/programmer;
4. tester; verifier;
5. reviewer, auditor
6. manager of application development;
7. configuration manager;

8. tool developer, support.

5.1 Common Information Needs

A systematic approach to reviewing information needs is proposed. Although the coverage in this paper is far from complete, a systematic approach should yield benefits when applied in particular situations, supporting a complete exploration of potential needs.

The 'systems-theoretic' approach to measurement advocated in this report considers measurement as embedded in 'control loops' linking a decision-making agent with the actions available and the domains in which measurements are made and the actions have an effect. This approach amounts to an extended application of the classic PDCA cycle that has been applied to information security management in [20].

A decision-maker is assumed to be acting purposefully to achieve some goal (Figure 20). The basic tasks involve planning, enacting the plan and checking the outcome. The following measurements needs are implied:

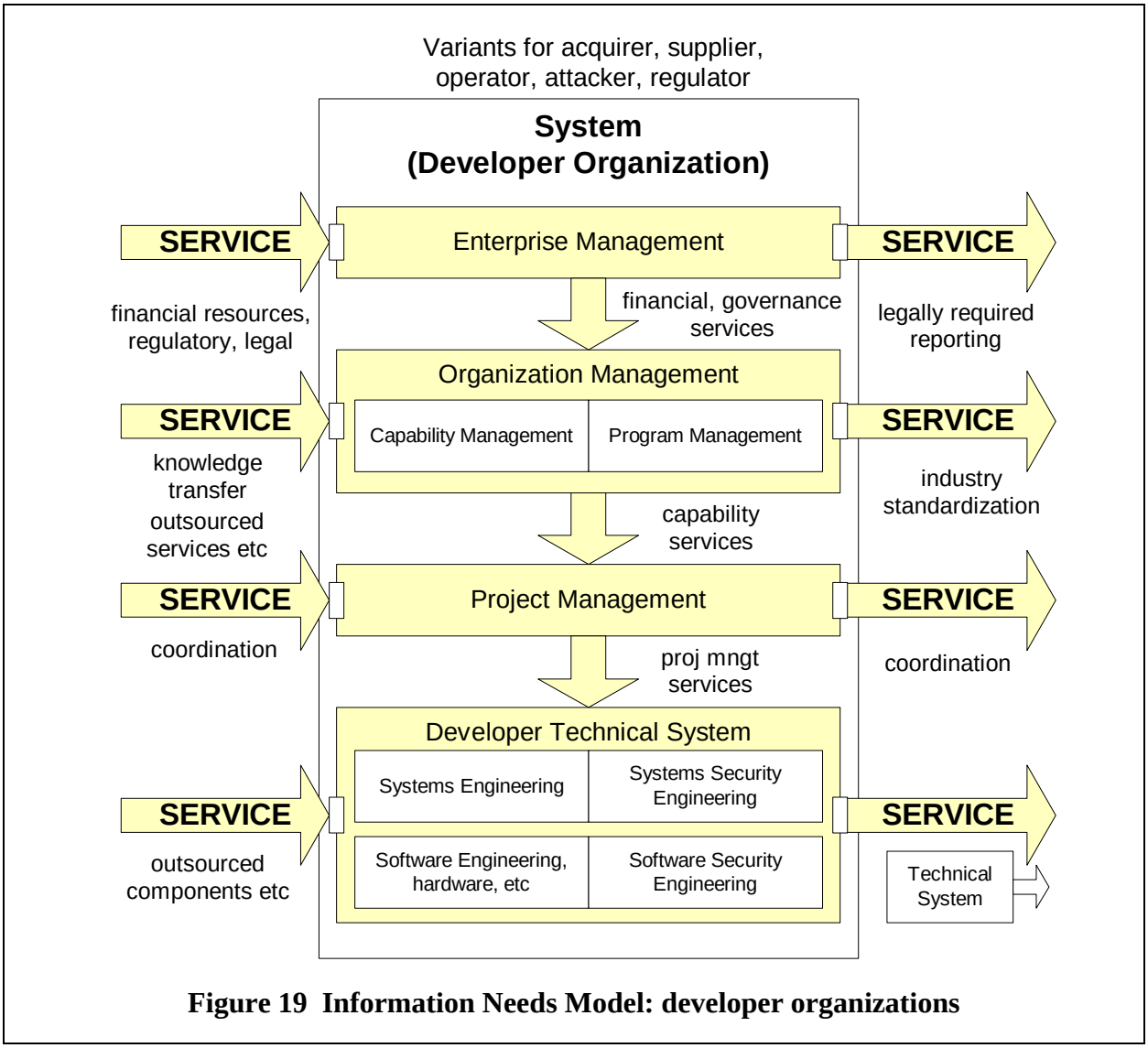


Figure 19 Information Needs Model: developer organizations

- Planning – setting targets, specifications, objectives
- Monitoring enactment of the plan – progress against plan (verification)
- Checking outcome of plan – effect of actions (validation).

Most decision-makers work under policy, regulatory or other constraints, implying an additional measurement need associated with *compliance*. The plan, the effects of actions and associated measurements are all subject to uncertainty. This is represented as a need for *risk assessment* and management. Finally, we add the heading of *assurance* to the model of Figure 20. Many decision-makers work in situations in which a customer or other party will be subject to risk arising from faults or uncertainties in the work done. Those parties will require assurance evidence to support assessments of their risk exposure.

The information needs of the decision-maker are considered under the headings of Figure 20. Detailed needs and measurement specifications will be determined by the decision type and subject domain. Some aspects of decision-making are common across all decision situations. The following are reviewed below:

- The viewpoint in time;
- Costs of actions and benefits of outcomes;
- Uncertainty.

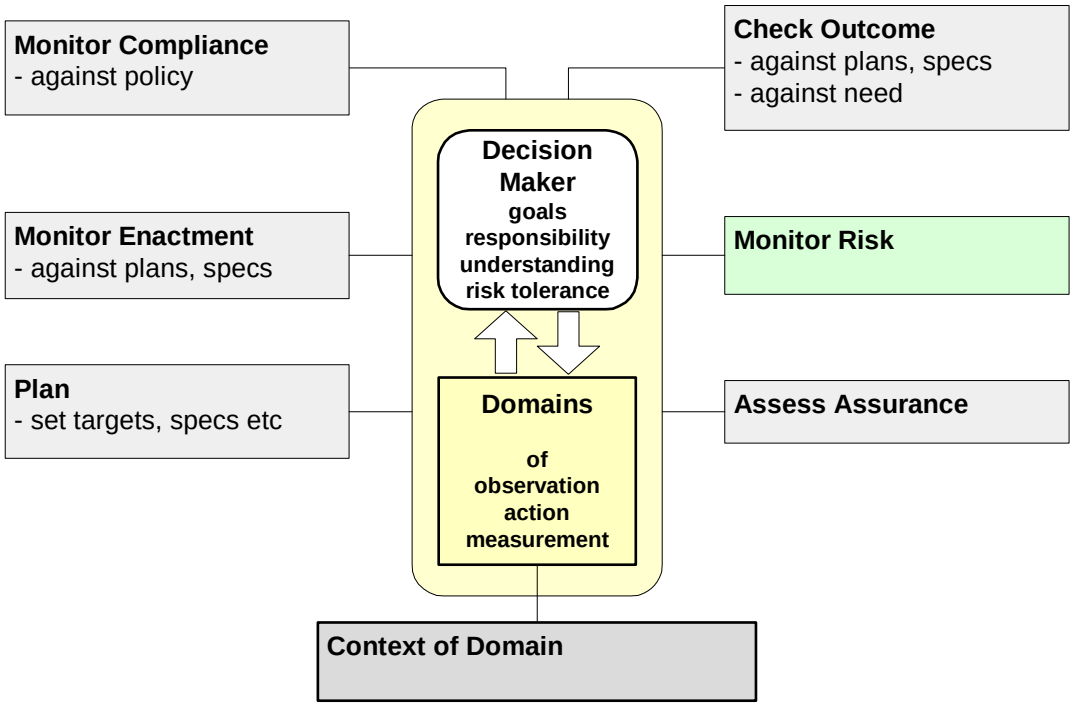


Figure 20 Generic information needs arising when a plan is enacted under uncertainty

5.1.1 Viewpoint in Time

Most decision situations involve a desired future state and a choice based on assessed past performance and current opportunities and constraints. Assessments are subject to uncertainty, giving rise to:

1. information needs about the past, recent or distant, for example:
    2.1. How secure has the system been, for example in terms of losses incurred ascribed to security incidents, including attempted intrusions? What is the uncertainty in this assessment?
    2.2. How efficient/effective have the security processes been, as enacted?
    2.3. What was the achieved performance compared with policy/objectives?
    2.4. What was the achieved performance in customer (other stakeholder) terms?

2. information needs about the present, for example:
    2.1. How secure is the system? What is the uncertainty in this assessment?
    2.2. What is the current performance of the system/ organization, compared with the security objectives?
    2.3 What are the achieved performance outcomes of the actions taken?
    2.4 What resources are actually being deployed?
    2.5 To what degree are policies, legal requirements and standards being complied with?
    2.6 What is the progress / status of security work and products in development and assurance processes

3. information needs about the future, for example:
    3.1 How secure will the system be (or what are the residual security risks), for different sources of risk? What is the uncertainty in this assessment?
    3.2 How much is it worth spending to reduce security risks? What are the most cost-effective actions to reduce security risks? What are the opportunity costs?
    3.3 How ready is the developer/operator organization to undertake security-critical work?
    3.4 How will the threat environment evolve in the future?
    3.5 What resources should be committed to the security work (money, time, capability)? And over what timescales (current project, medium term development)?

Decisions to act depend on confidence that the domain is adequately understood, that there is an acceptable likelihood that desired outcomes will result and that the risks of undesired outcomes are acceptably low. Outcomes may be intermediate ones, establishing progress towards a more distant final outcome. Intermediate outcomes may be viewed as providing options for future decisions. If a causal connection between action and outcome cannot be established with sufficient confidence, or if the downside risks are judged unacceptable, inaction will likely result.

Only past and current performance can be measured. The future has to be predicted, estimated or anticipated on the basis of past and current performance and the understanding of the decision-making agent. The degree to which the past is a dependable indicator of the future is an important consideration for the decision-maker.

5.1.2 Costs and benefits

Most decisions involve trade-offs between costs and benefits. Difficulties often arise in assessing the costs and benefits with sufficient confidence. Security-related decisions will involve typically the following questions:

1. What level of security in the component/system or service do I need?
2. How can I tell if the level of security is achieved or is likely to be achieved in the future?
3. What level of confidence do I need in assessment of security properties?
4. How can I tell if the level of confidence in assessment of security is achieved or likely to be achieved in the future?
5. What are the feasible ways of improving the security of the component, system or service? How can I trade-off the costs and benefits of alternative investments?
6. How can I judge the outcomes of investments made in terms of improved security and other performance properties of the component, system or service?
7. How effective and efficient are the security-related actions/ processes that are implemented?

Such questions involve both technical and management concerns and arise at all levels of system development.

Suppose we can estimate the reduction in security risks arising from a particular expenditure. The benefits arising from the expenditure on security can be viewed as the *Return On Security Investment* (ROSI) [30]. For a decision-maker to be able to trade-off costs and benefits, for example with other competing risk mitigations, some relationship is needed between risk reduction and cost (Figure 21).

Suppose a system is judged to be located at A_1, with respect a particular type of failure. An investment may yield an improvement to A_2, but at cost. An alternative might involve a change in design or technology that would place the system on the lower curve. A solution at B_2 would provide reduced risk at a reduced cost. Uncertainties in the assessment of risk and cost are usually too large to allow direct use of such models. However, they make explicit the needs of the decision-maker and provide an objective for measurement system design.

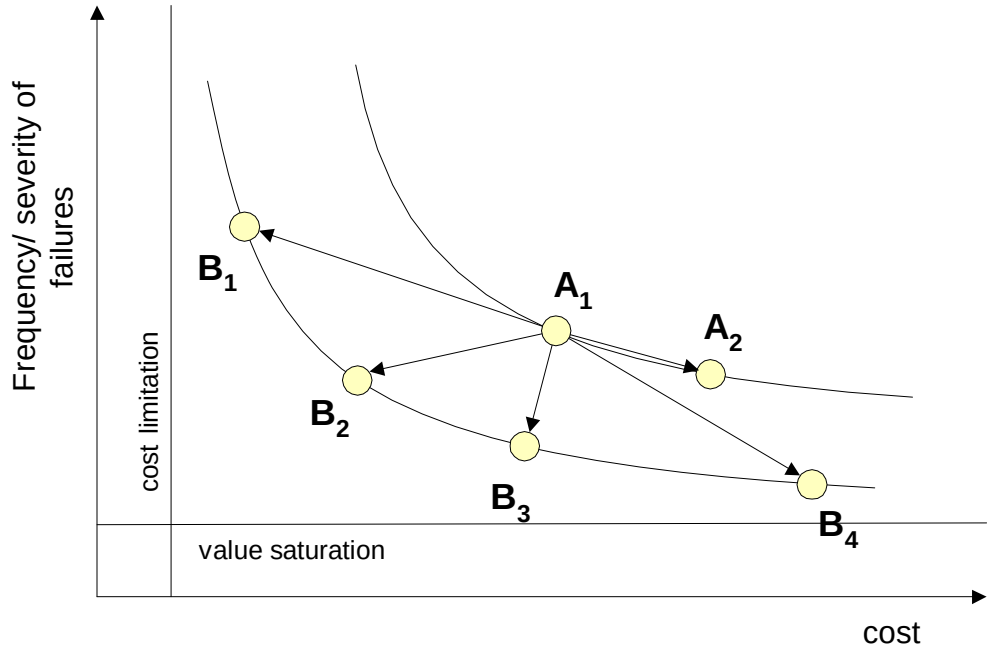


Figure 21 Dependency between failure frequency and severity, and costs invested in reduction of these [31].

Risk management, in different forms, is present at all levels of management. The following types of risk mitigation strategy are involved in security:

1. compliance with policy, reflecting regulatory requirements and standards;
2. investment in mitigations that lead to reductions in security-related losses; this appropriate where risk events are bearable but costly;
3. investment in mitigations that lead to evidence-based reductions in risk; this is appropriate where events are very costly and rare (c.f. safety risks); the ALARP principle (*As Low As Reasonably Practicable*) is invoked in the safety field to cover such risk acceptance decisions;
4. transfer of risk to partners or by insurance;
5. acceptance of risks; appropriate where events are rare and the losses acceptable.

5.1.3 Uncertainty

In many cases, we do not have sufficient data on which to objectively assess risks and the risk reductions achieved by security improvement actions. Decisions are made based on engineering and management judgment.

The best strategy for security measurement seems to be to start from this reality. Probability theory can be used to represent the assessments and confidence of security engineering, operations and management staff. This Bayesian interpretation of probability implies that we are systematizing subjective assessments, rather than supporting objective measurement. However, this approach is accepted in the project risk management field, for example, [15].

We can also establish a basis for developing objective measurements to serve the needs of the particular decision maker, supplementing subjective judgment. Over time, the decision might be based increasingly on objective data. Explicit consideration of uncertainties is helpful in many situations: even sparse data can reduce uncertainty sufficiently to enable a particular decision. The proposed systems-theoretic model for the application of measurement in the security field has the aim of evolving objective measurements to support decision-making that is initially mainly subjective.

A measurement approach to uncertainty would reduce the tendency to use single point estimates of likelihood and severity – effectively ignoring the variance around such expected values. It also recognizes that there are variable needs for accuracy – the value of increased accuracy in an estimate depends on the decision served.

An additional source of uncertainty in the security field is due to threats being *learning* agents, implying a lack of predictability. Three strategies are used:

1. Conventional planning (project development or operational plan) in which tasks are designed, resources deployed and progress monitored against plans (the ‘knowns’);
2. Risk management, in which contingencies are made to cover the occurrence of events that have been identified as possible (the ‘known unknowns’);
3. Awareness/readiness, in which resources are deployed to detect and respond to departures from expected operations (the ‘unknown unknowns’).

The best mix of these approaches depends on the flexibility available and the levels of uncertainty and risk involved. An example of an awareness approach is CISCO’s monitoring of unusual patterns of network use [25].

The relative time constants associated with the security (defensive) actions and the learning of the threat agents determines the type of appropriate response. In real-time operational situations, emphasis shifts to detection and rapid response. Time may not be available to improve accuracy or perform quantified trade-offs. Response will be based on an already-acquired knowledge and skills and on deployed measurement systems.

5.1.4 Acquisition and Trust

Many decisions concern acceptance of a product or service, a go/no-go decision based on criteria at a phase gate or review, an acceptance of a proposal or engagement of a supplier, an acceptance of an assessment of dependability or audit of a supplier process. Uncertainty and trust are involved such cases. Very often, there are significant gaps in expectation and understanding between different agents [31].

Assurance is required by an agent exposed to risks arising from faults introduced by a supplier or service provider. Assurance processes develop evidence on which risk acceptance decisions can be based. Various approaches to assurance are used, for example, process audits, product testing and third party certification. Traditional quality assurance was developed from experience in manufacturing. Repetitive processes enabled the application of statistical techniques such as quality control charts. The CMM approach to process improvement is based on a similar philosophy; develop repeatable processes that can be quantified and improved. Design processes, particularly for complex systems, are more difficult to treat in this way, but the principles are valid. Where there is insufficient stability or understanding of causal models, we cannot generate sufficient data on which to base statistical measurements. Instead, we have to move to an approach based on subjective judgment supplemented with selected measures.

Providing assurance evidence may represent an additional cost, depending on the information generated by the core development processes. There are trade-offs with other uses of such investments. Also the need for assurance varies, depending on the degree of dependence of the users on the provided services and their risk tolerance.

5.2 Enterprise Level

Senior executives and managers in organizations are responsible for regulatory compliance, establishing a security policy and setting up and monitoring internal security processes, ensuring governance in terms of representing the interests of various parties and providing assurance. These are essentially the headings of Figure 20 applied to a decision maker acting on an organization, viewed in enterprise terms. For example, based on the recommendations of the CISWG study [10], the following information needs can be identified at this level:

Decision Maker:	**Domain:**	**Context of Domain:**
Board Level	Enterprise (profit, not-for-profit)	Regulatory environment Threat environment Competitive environment
Information Need Heading	**Example Information Need**	
Plan	Policy and Objectives, security program, responsibility allocation	
Monitor Enactment of Plan	Policy compliance, program compliance	
Monitor Compliance	Sarbanes-Oxley, HIPAA, Gramm-Leach-Bliley	

Check Outcome	Monitor security losses, costs, program performance
Monitor Risk	Organizational risks, enterprise continuity
Assess Assurance	Internal and External Audits of security program

The top-level metrics recommended under these headings are given in [10]. Similar responsibilities would be expected for developer organizations, at this level, with the oversight of information security policy being supplemented with that of security policy and practices associated with product development.

In addition to the above, senior executives will be concerned with the costs involved in meeting security objectives and the effectiveness of investments made:

1. for developers of secure products, integrated performance of security efforts and the ROSI, as evidenced in security properties of developed products and services for client organizations;
2. for general information security, integrated performance and ROSI of security processes.

Traditional ROI calculations can be applied to security investments (Appendix 3). The inputs to ROI calculations are subject to considerable uncertainty in many cases.

In uncertain situations in which we are concerned about evolving threats, decision-making based on simple ROI assessments may be unwise. Choosing not to invest in a security action may lock us out of learning about evolving threats. An alternative decision model may be to use 'real options' theory – which has been applied to R&D management [32].

5.3 Organization Level

Managers at this level have the responsibility of establishing and improving security capabilities, so that the organizational policies and programs can be implemented. For example, the CISWG study [10] identifies the following responsibilities in the management of general information security:

1. Establish Information Security Management Policies and Controls and Monitor Compliance;
2. Assign Information Security Roles, Responsibilities, Required Skills, and Enforce Role-based Information Access Privileges;
3. Assess Information Risks;
4. Establish Risk Thresholds and Actively Manage Risk Mitigation;
5. Ensure Implementation of Information Security Requirements for Strategic Partners and Other Third-parties;
6. Identify and Classify Information Assets;
7. Implement and Test Business Continuity Plans;
8. Approve Information Systems Architecture during Acquisition, Development, Operations, and Maintenance;
9. Protect the Physical Environment;
10. Ensure Internal and External Audits of the Information Security Program with Timely Follow-up;
11. Collaborate with Security Staff to Specify the Information Security Metrics to be Reported to Management.

Various information needs are implied by these responsibilities. Some 39 metrics are recommended under these headings in [10], mainly monitoring compliance with the security policy. Measurements against policy may be more convincing in showing and predicting non-compliance than in establishing assurance.

Organizations that develop security-critical products have additional responsibilities associated with the security engineering capability required for development purposes. Two kinds of responsibility are involved:

1. the development of security engineering capability, as required by development projects;
2. the management of those resources across projects, possibly as part of a program or product line.

The SSE-CMM [9] practice areas provide a reference for security engineering capability and a basis for developing security metrics.

As an example, the model of Figure 20 might be applied to a decision-maker responsible for security capability as follows:

Decision Maker:	**Domain:**	**Context of Domain:**
Security Capability Manager	Developer Organization	Enterprise Policy and Security Program Regulatory environment Supply Chain Environment
Information Need Heading	**Example Information Need**	
Plan	Security Process and Resource Plan: performance objectives	
Monitor Enactment of Plan	Process enactment on the part of development projects; resource, schedule, costs, effectiveness (faults introduced, detected etc); monitor resource development, training	
Monitor Compliance	Enterprise Policy, regulations	
Check Outcome	Monitor security losses arising from use of product, costs, process performance, efficiency, effectiveness, ROI	
Monitor Risk	Capability risks – tracking changing technology, threats	
Assess Assurance	Internal and External Audits of security process	

Information needs arise in terms of establishing continuous improvement in technical processes. A distinction is drawn between:

1. the measurement of cost, performance and risk in a vertically integrated way (through the organizational hierarchy/ responsibility chain) and
2. the measurement of end-to-end process performance in a horizontally integrated way (the process view).

Both views are important. The first is directed more at assessing the effectiveness of security investments and risk management (with less attention on how the work is structured); the second more towards assessing the efficiency of end-to-end security processes (to support process improvement). 'Vertical' measurement of performance and risk does not itself require work areas to be structured as processes, but is compatible with a process approach. However work is organized, it seems important to ground measurement as much as possible in the technical/ operational practice level i.e. at the level where risks are detected and where inventiveness and creativity are deployed in their mitigation. This orientation may help to tailor/anchor best practice models based on process maturity to specific project and operational situations [21].

The estimation of likely future costs of security is being addressed by the parametric cost modeling community [33].

5.4 Project Level

The management of a project gives rise to a set of information needs that are focused on the progress of development of a particular product. This is the main concern of the PSM / ISO 15939 measurement framework.

Decision Maker:	**Domain:**	**Context of Domain:**
Project Manager (Security)	Development project of a security-critical system	Developer organization and security capability Supply chain environment Regulatory Environment Enterprise Policy and Program
Information Need Heading	**Example Information Need**	
Plan	Project Plan, development security policy, performance objectives	
Monitor Enactment of Plan	Security requirements and compliance Security properties of components and their verification status Progress and costs of mitigation actions, against plans and risk tracking, keyed with system development life-cycle Integrated security performance, balancing investment between identified risks Trade-offs between security and other system performances Readiness/ awareness (readiness to respond to events not foreseen in plans and risk assessments)	
Monitor Compliance	Enterprise Policy and Security Program Process models and applicable standards Regulations	
Check Outcome	Integrated past performance of security activity Efficiency and effectiveness of security actions.	
Monitor Risk	Identified security risks (identified threats and vulnerabilities)	
Assess Assurance	Tracking security assurance activities, progress and costs.	

The information categories in the PSM model follow from the headings when applied to general software project management. The PSM Information Categories are [3]:

1. Schedule and Progress: compliance with project plan, with meeting engineering specifications;
2. Resources and Cost: compliance with project plan, available resource and consumption;
3. Product Size and Stability: changing requirements and design;
4. Product Quality: assurance and product performance outcome;
5. Process Performance: process compliance and monitoring performance;
6. Technology Effectiveness: design and implementation monitoring at component level;
7. Customer Satisfaction: service performance outcome.

5.5 Technical/Professional Specialty Level

A wide range of technical specialties are involved in security engineering and operations. For example, the CISWG study [10] lists the following elements of an information security program, at technical operations level:

1. User Identification and Authentication
2. User Account Management
3. User Privileges
4. Configuration Management

5. Event and Activity Logging and Monitoring
6. Communications, Email, and Remote Access Security
7. Malicious Code Protection
8. Software Change Management, including Patching
9. Firewalls
10. Data Encryption
11. Backup and Recovery
12. Incident and Vulnerability Detection and Response

The selection of topics on this list may be expected to change as new types of threats emerge.

These responsibilities are mainly concerned with the operation of technical features existing in commercially available IT systems and COTS components. The security policy is implemented as a set of decisions on how to deploy 'security controls' (e.g. automatic logging off of users after a selected idle time). The recommended metrics generally reflect this orientation.

In the case of product development organizations, the scope of technical specialization involved will depend on the product type and technologies involved. Decision-makers will be concerned with the preventive, constructive and tolerance aspects of security faults, in addition to the development of operational policies and counter-measures.

Software-intensive systems involve the specialist fields of network security, computer security, specialist security components and technologies, software security and associated hardware security (e.g. tamper-proofing). These specialties have their own practices and measurement concepts.

Information needs at technical development levels are mainly concerned with assessing and predicting the performance of designed, implemented and deployed products. Measurement is conducted with reference to requirements and specifications and in the context of a system development life cycle. Product measurement overlaps with the quality assurance field (e.g. [34] for software) at this level.

Both top-down (identification of potential service failures and actions to mitigate the risks) and bottom-up (identification of potential faults and actions to avoid or otherwise mitigate them) approaches are used. Specialist practices address improvements to reduce the introduction of faults and their detection and mitigation if introduced. As discussed in Section 4.3, security concerns will typically lead to improvements in core product development processes (e.g. software engineering), as well as to specific security practices (e.g. development of intrusion detection systems).

Table 1 summarizes the information needs arising at the management levels discussed here and in relation to current, past and future performance.

Security systems engineering will typically carry responsibility for integrating security aspects over total systems. The aggregation of security risks over a system is discussed in Section 2.4.

	Information Needs about:		
	Future Performance (planning)	**Current Performance (monitoring)**	**Achieved Past Performance (assessing)**
Enterprise	Enterprise risk Public (externalized) risk Threat Environment Legal/ governance environment	Current security performance Current Enterprise risk Public risk exposure Current expenditure Current resource allocations Compliance with legal, policy, governance	Costs of security Delivered security performance Effects on profitability & productivity Opportunity costs, ROSI Learning curves Effectiveness of policies, legal environment
Organization	Future process risk Predicted security performance & risk Future threats Future benefits Organizational governance environment. Standard practices. Capability model	Current security capability Maturity Benchmarking Current investment in development – product and process Current outcomes Responsiveness, flexibility, awareness Competence Compliance with processes, stds, capability	Delivered security performance Effectiveness and efficiency of security processes, security management system, policy – process performance Actual costs Achieved Process maturity Effectiveness of standards. Efficiency of processes and standards
Project/ Operations	Estimation of Resources and Costs of security development and operations Project Risk, Planning Applicable Regulatory and standards environment. Development and Operations Policies	Schedule & Progress of work against plan Progress of risk mitigations Progress of contingency actions, Costs Outcomes of tasks in terms of risk and performance Event response Monitor compliance with policies, standards and regulations	Actual delivered security effectiveness of integrated product or service Assurance. Customer satisfaction Integrated efficiency and effectiveness at project/ operations level Effectiveness and efficiency of project-enacted management work.
Technical	Predicted technical security risks Estimates, Product size, Planning Required security performance Anticipation Technology Effectiveness Applicable Regulatory and standards environment. Development and Operations Policies Requirements and specifications	Current residual security risks Progress of risk mitigations Current performance – response, recovery Current costs; stability. Product Quality Relative merit of different risk mitigation options Monitor compliance with policies, standards and regulations at technical levels Monitor application of lessons learnt, checklists etc	Integrated costs of security work Integrated effectiveness and efficiency at technical level Achieved security performance – product quality Security-related damages Assurance Effectiveness and efficiency of project-enacted technical work.

Table 1 Example information needs of managers of security-related work

6 Security Measurement Map

Security measurement needs arise in a wide range of decision situations. We would like to have an entry point to measurement guidance that is applicable to all security-related decision situations that arise in the development and operation of software-intensive systems.

A set of categories of measurements is developed that aims to be complete, in the sense that all issues that arise in security are covered by the single categorization. Subsequent research and practical experience will then develop guidance linked into this framework, including the development of constructs from measurable concepts etc. If the categorization is found to be incomplete, it can be extended. This approach is based on the PSM ICM table [3], developed in the domain of software engineering, project management and acquisition.

The systems-theoretic approach starts with the identification of a decision-maker (assumed to have the information need) and the type of information need involved, which amounts to an identification of the type of decision process and measurement. The generic model of Figure 20, based on the classic PDCA cycle, with the addition of compliance, risk and assurance, is proposed as the starting point.

The headings illustrated in Figure 22 are proposed as a useful reference map in the case of a

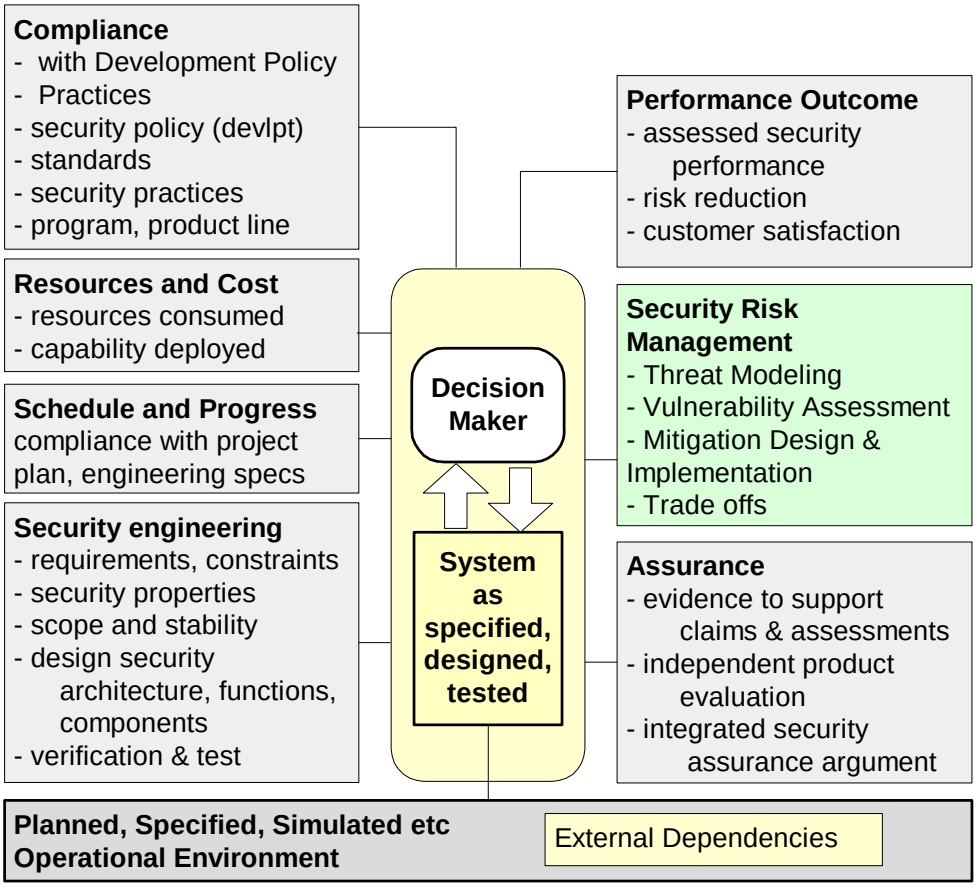


Figure 22 Security Measurement Map - system development

decision-making during system development, at project management level. The following seven categories are proposed:

1. Security Engineering;
2. Schedule & Progress;
3. Resources and Cost;
4. Compliance;
5. Performance Outcomes;
6. Security Risk Management;
7. Assurance.

The rationale for the seven headings is as follows. The managed domain involves security engineering; measurement constructs will map to metrics in that domain. From a management perspective, the engineering level will assist with planning and monitoring progress in terms of technical progress towards meeting specifications and mitigating risks. Schedule & Progress and Resources & Costs are PSM headings covering project management aspects. Compliance refers to monitoring the degree to which organizational policies, standards and legal requirements are being met. Performance outcomes refer to the checking the effectiveness of the managed work in customer terms i.e. validation. Security risk management covers information needs associated with uncertainties in the technical and project plans. Assurance covers information needs associated with the provision of assurance evidence for customer or regulatory agents.

Similar headings will be appropriate for decision-makers in operations settings.

The model of Figure 22 is still general; a particular application will involve specialization to the particular type of system, its architecture and components and to the security practices deployed.

7 Developing Security Measurements

PSM and ISO 15939 [2] are based on the concept of *measurement constructs* (Figure 23) that link the information needs of managers with base measures of artifacts present in the managed domain. Measurement constructs embody understanding about the measured system and how measurements relate to management responsibilities. Such understanding covers:

1. *who* is involved in the domain; what are their roles, responsibilities, goals and values, leading to information needs? These questions are addressed by means of an **Information Needs Model**, based on the PSM layered management model;

2. *what* is the 'target' system, asset, service, or operation that is subject to management effort? What is it that has the security properties that are being engineered and maintained? What are the development and/or operational Environment of this system? These questions are addressed by developing a **Target System Model**. This is useful because there are significant differences between the kinds of systems of concern;

3. *what* are the security performances of concern? How is security performance manifested? How does pursuance of this property interact with the other performance attributes of an integrated system or service? These questions are addressed by means of a **Security Concept Model,** developed to be compatible with the safety concept model [4]. The objective is to provide a 'bridge' between security professionals and managers; the model provides a basis for decomposing information needs into measurable concepts in a top-down fashion;

4. *how* is the property pursued or enhanced by the specialty engineering and operations communities? What practices and work products are involved? We address these questions by developing a **Representative Practices Model**, based on published best practice and standards in the specialty;

5. finally, a **Measurable Entities Model** is developed; work products associated with security engineering and operations management are identified and measurable attributes identified. This model provides a basis for synthesizing potential measures in a bottom-up fashion.

The work reported in this paper has addressed several of these models. Information needs have been explored with reference to decision-makers at different management levels within development and operations organizations. Security has been modeled conceptually in terms of the dependability-related definition, as a risk management activity. The measurement target system has been described in general terms, with reference to architecture and components. Security practices have been discussed and reference made to the SSE-CMM practice areas. Measurable artifacts have not been explored in any detail, other than to mention risk tracking, costs and related generic concepts.

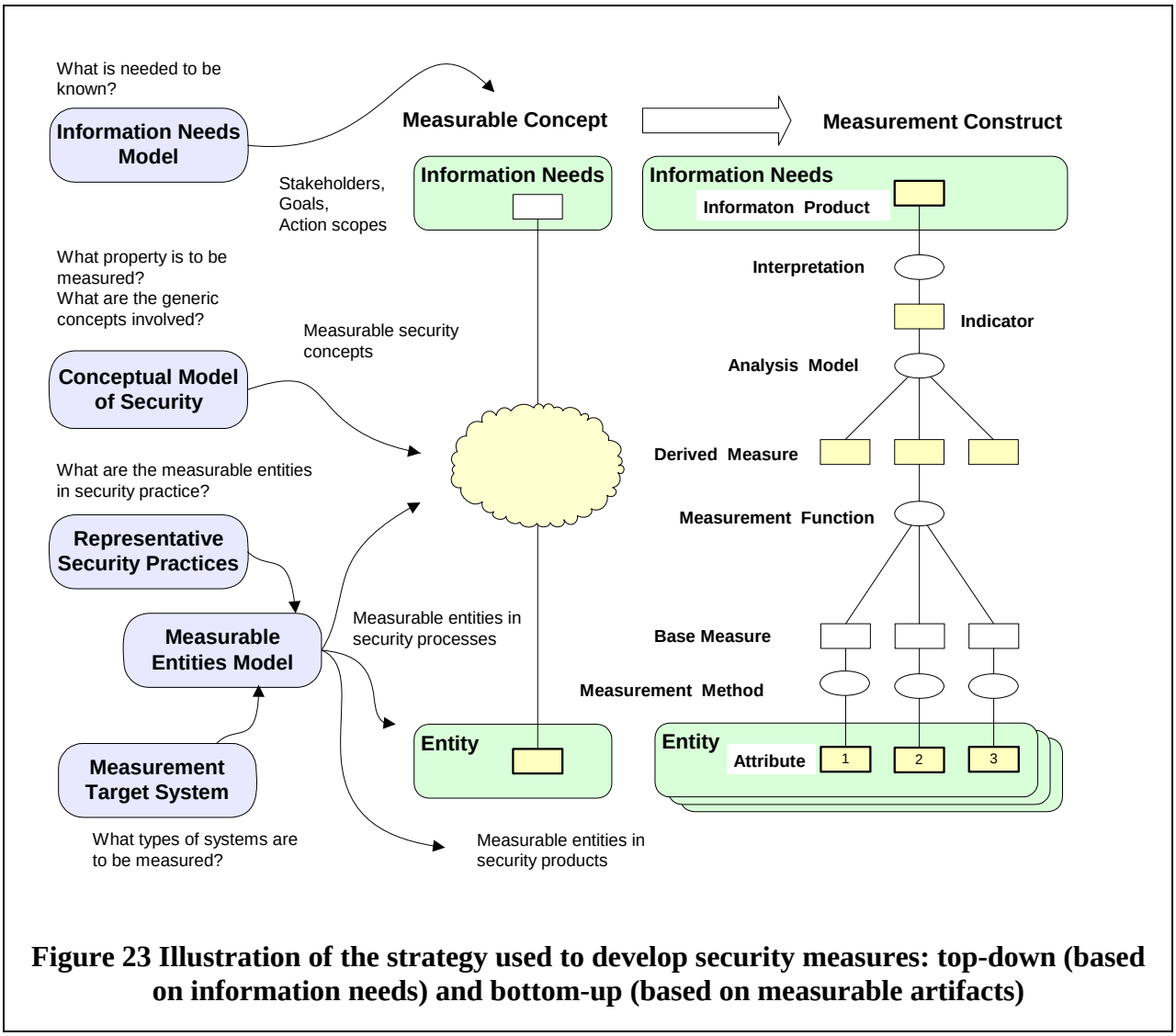


Figure 23 Illustration of the strategy used to develop security measures: top-down (based on information needs) and bottom-up (based on measurable artifacts)

Practical measurement guidance will require the development of example measurement constructs and specifications. This will require applying the general measurement headings to more specific security engineering and operational situations.

The most challenging aspect of this work is to support the management need for assessments of aggregated security risk and cost/effectiveness trade-offs arising from alternative security strategies. An approach based on fault paths and assessed risks has been proposed. The 'systems theoretic' model for measurement recognizes the lack of empirical data needed as a basis for objective probabilistic reasoning. The proposed concept seeks to foster the development of objective measurements, starting from the subjective assessments that may be feasible to begin with.

Figure 24 summarizes the areas of application of measurements involved in developing the security of a service (developed from Figure 12). The labels M1, M2 .. in Figure 24 indicate the following:

M#	Measurement Area	Types of Measurement
M1	Enterprise Level Management	Monitoring compliance with legal requirements e.g. by monitoring audit shortfalls Financial Performance: return on investment, market share, options Investment Decisions: trade-off and selection between competing strategies and investments
M2	Capability Management Program Management	Process Performance: effectiveness and efficiency Practice Areas (CMM models): maturity assessment through audits Resource Management across project portfolios: trade-offs between alternative deployments of limited resources
M3	Project Management	Planning, progress monitoring, resource consumption, outcomes; risk management
M4	Operations Management	Planning, performance monitoring, resource consumption, outcomes; risk management
M5	Technical Management Engineering	Fault Management, Fault Trees Technical Risk Management Product Properties Application of specialist methods
M6	Acquired and provided technical components and systems	Product, System, Service Properties Assurance
M7	Threat Monitoring	Threats to development processes: agent properties
M8	Threat Monitoring	Threats to operations: agent properties
M9	Technical Management Maintenance	Corrective maintenance during operations Product Updates
M10	Service Failures	Security events that affect the provided service Frequency Recovery costs
M11	User System	Damage costs caused by service failures

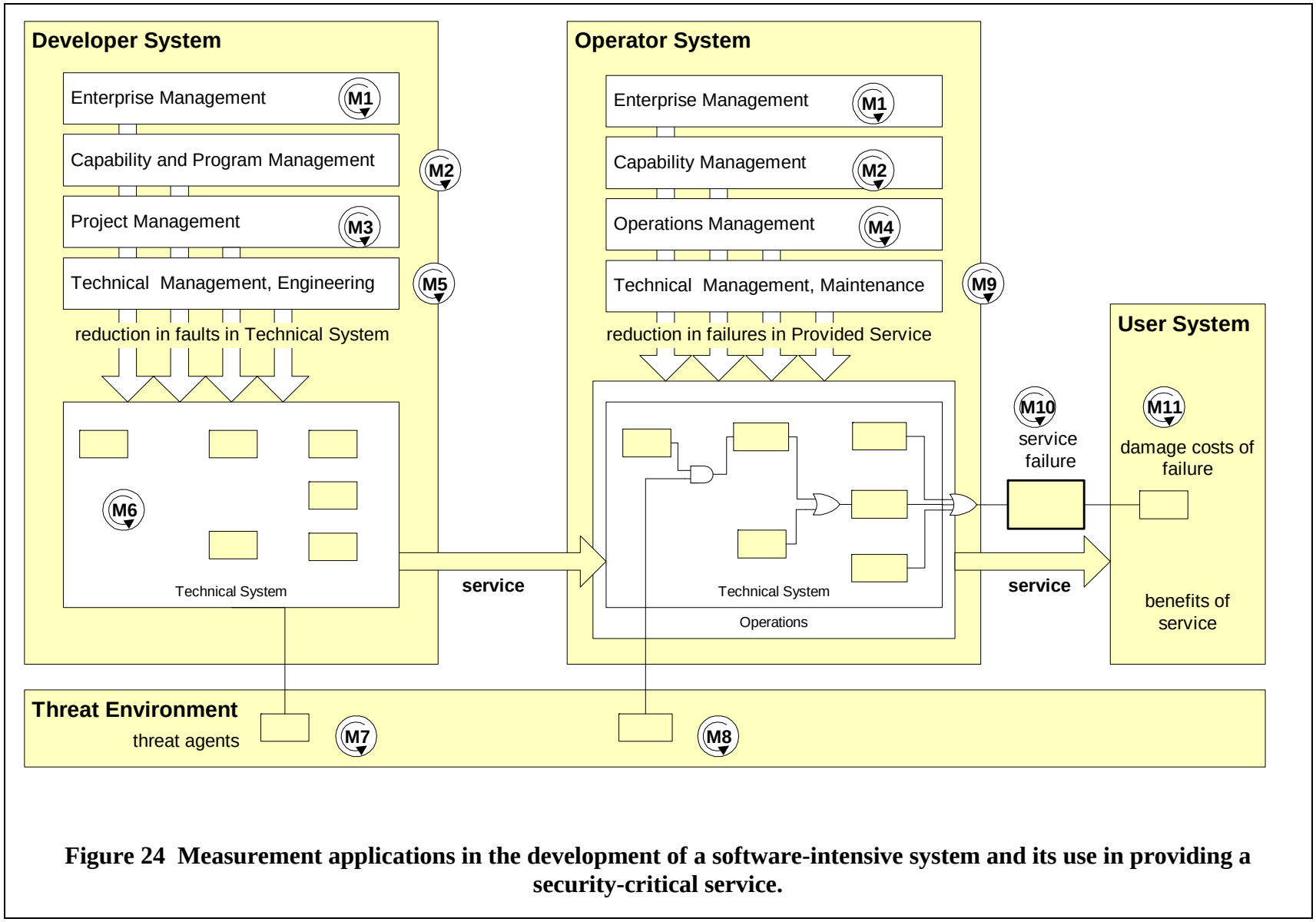


Figure 24 Measurement applications in the development of a software-intensive system and its use in providing a security-critical service.

The next steps for developing security measurement guidance are as follows:

1. Select representative decision-making roles and project situations (e.g. development of secure software components);
2. Select representative information needs, drawn from the generic models developed in this report;
3. Select representative measurable entities from the decision domain (e.g. security requirements, risks, test results);
4. Develop indicators to serve information needs, based on identified base measures;
5. Develop measurement specifications applicable to similar situations and illustrative for other situations.

8 Conclusion

The security field is diverse and evolving rapidly to meet the dual challenges of net-centric systems and increasingly capable threat agents. Measurement concepts are needed that bridge between specialist technical domains and integrated systems management. We need indicators of security properties that are *good enough* to support the types of decision that have to be made at aggregated levels of systems and services. At the same time, 'rolled up' approximations of security properties have to be amenable to being 'unrolled' as particular concerns and threats evolve.

An integrated approach to security measurement has been proposed, drawing on measurement, risk management and systems concepts. The PSM TWG is bringing forward practical guidance materials on security measurement, informed by the concepts developed in this report.

Collaborative work is also in hand with the Measurement Working Group of ISSEA, in the context of the SSE-CMM, recent NIST measurement guidance (NIST SP 800-55 [5]) and the ongoing development of ISO/IEC 27004.

9 References

1. McGarry, J., *et al., Practical Software Measurement; objective information for decision makers*. 2001, Boston: Addison-Wesley. 277.
2. ISO/IEC, **ISO/IEC 15939:2002(E),** *ISO/IEC 15939:2002(E) International Standard Software engineering - Software measurement process*, 2002-07-15, ISO/IEC.
3. PSM, *PSM Security Measurement Guidance.* 2006(v1).
4. Murdoch, J., **v 2.0,** *Safety and Security Measurement*, February 2004, PSM.
5. Swanson, M., *et al.*, **NIST Special Publication 800-55,** *Security Metrics Guide for Information Technology Systems*, July 2003, NIST.
6. ISO/IEC, *ISO/IEC 27004 WD Information Technology - Security techniques - Information security metrics and measurements*, 30 June 2005, ISO/IEC.
7. ISSEA, *Metrics Working Group Report, 4th Annual ISSEA Conference*, International Systems Security Engineering Association.
8. SSE-CMM, *Systems Security Engineering Capability Maturity Model® SSE-CMM® Model Description Document Version 3.0*, 15 June 2003,
9. ISO/IEC, **ISO/IEC 21827 v3.0,** *ISO/IEC 21827 Systems Security Engineering Capability Maturity Model (SSE-CMM)*, ISO/IEC.
10. CS1/05-0005, **CS1/05-0005,** *Corporate Information Security Working Group: report of the best practices and metrics teams*, 17 November 2004, 10 January 2005, Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, Government Reform Committee, US House of Representatives.
11. DHS, *Software Assurance Program, .* 2005, US Department of Homeland Security, Cyber Security Division
12. ISO/IEC, *ISO/IEC 16085 Risk Management. .*
13. Avizienis, A., *et al., Basic concepts and taxonomy of dependable and secure computing.* IEEE Trans. Dependable and Secure Computing, 2004. **1**(1): p. 11-33.
14. Roberts, N.H., *et al.*, **NUREG -,** *Fault Tree Handbook*, Systems and Reliability Research Office, US Nuclear Regulatory Commission.
15. Chapman, C. and S. Ward, *Project Risk Management: processes, techniques and insights*. 2nd ed. 2003, Chichester: John Wiley. 389.
16. Landwehr, C.E., *Computer Security.* International Journal on Information Security (IJIS), 2001. **1**: p. 3-13.
17. Williams, J.R. and G.F. Gelen, **ATR 97043,** *A Framework for Reasoning about Assurance*, 23 April 1998, Arca systems Inc.
18. ISO/IEC, **ISO/IEC 15408 v3,,** *ISO/IEC 15408 Common Criteria for Information Technology Security Evaluation*, Common Criteria Project Sponsoring Organizations.
19. ISO/IEC, *ISO/IEC 15026 Assurance.* 2005.
20. ISO/IEC, **ISO/IEC 17799:2005,** *ISO/IEC 17799:2005 Information technology - Code of practice for information security management*, 2005, ISO/IEC.
21. Charette, R., L.M. Dwinnell, and J. McGarry, *Understanding the Roots of Process Performance Failure.* CrossTalk, 2004. **August 2004**: p. 18-22.
22. Leveson, N.G., *A systems-theoretic approach to safety in software-intensive systems.* IEEE Trans. Dependable and Secure Computing, 2004. **1**(1): p. 66-86.
23. Ibrahim, L., *et al., Safety and Security Extensions for Integrated Capability Maturity Models*, September 2004, US FAA & DoD.

24. DoDAF, *DoD Architecture Framework Version 1.0*, 9th February 2004, US Department of Defense, Architecture Framework Working Group.
25. Paquet, C. and W. Saxe, *The Business Case for Network Security: advocacy, governance and ROI*. 2005, Indianapolis: Cisco Press. 381.
26. Swiderski, F. and W. Snyder, *Threat Modelling*. 2004, Redmond, Washington: Microsoft. 259.
27. Stoneburner, G., **NIST SP 800-33,** *Underlying Technical Models for Information Technology Security*, December 2001, NIST.
28. Jones, A.n. and D. Ashenden, *Risk Management for Computer Security: protecting your network and information assets*. 2005, Oxford: Elsevier. 274.
29. Howard, M. and D. LeBlanc, *Writing Secure Code*. 2003: Microsoft Press International. 800.
30. Berinato, S., *Finally, a Real Return on Security Spending*, in *CIO Magazine*. 2002
31. Larsen, G. *Strategic Observations and Thoughts on a System Model for Security Metrics/Measurements: Why can't we get traction?* in PSM TWG. 23 March 2005 . Herndon VA. 2005.
32. Rouse, W.B. and K.R. Boff, *Value-Centered R&D Organizations: ten principles for characterizing, assessing and managing value.* Systems Engineering, 2004. **7**(2): p. 167-185.
33. Colbert, E. and e. al, *Costing Secure Systems: 5th Workshop*, 21 March 2004, USC-CSE.
34. ISO/IEC, *ISO 9126 Information technology - Software Product Evaluation - Quality characteristics and guidelines for their use*, ISO/IEC.
35. ISO/IEC, **ISO/IEC 15408-1:1999(E),** *ISO/IEC 15408-1:1999(E) Information technology — Security techniques — Evaluation criteria for IT security*, 1 December 1999, ISO/IEC.
36. PMI, *A Guide to the Project Management Body of Knowledge: PMBOK 2000*, Project Management Institute.
37. APM, *PRAM Project Risk Analysis and Management Guide*, Association for Project Management.
38. 882B, M.S., *System Safety Program Requirements*, . 1984, US Department of Defense: Washington DC
39. IEC, *IEC 1508-Functional Safety: Safety-Related System (Draft)*, International Electrotechnical Commission.
40. Weick, K. and K. Sutcliffe, *Managing the Unexpected: Assuring High Performance in an Age of Complexity*. 2001: John Wiley. 224.
41. Henning, R. *Workshop on Information Security System Scoring and Ranking*. May 21-23, 2001 . Williamsburg, VA: Applied Computer Security Associates & MITRE. 2001.
42. BSI, **BS 7799-2:2002,** *BS 7799-2:2002 Information security management systems - Specification with guidance for use*, 5 September 2002, British Standards Institute.
43. Deming, W.E., *The New Economics: for industry, government, education*. 2nd ed. 2000, Cambridge, Mass.: The MIT Press. 247.
44. Alberts, C.J. and A.J. Dorofee, **CMU/SEI-2001-TR-016,** *OCTAVE Criteria, Version 2.0*, December 2001, SEI CMU.

Appendix 1 Glossary

Acceptance	Agreement to receive and use, that contract terms are met, to take on risk
Asset	Information or resources to be protected by the countermeasures of a system. Part of a system which, if subject to a security failure, would cause damage. Adapted from ISO/IEC 15408-1 [35]
Assurance	The basis on which trust is placed in a system or service. The provision of the basis, usually in the form of evidence and analysis
Attack Goal	The objective of an attacker.
Attack Tree or Threat Tree	The means by which the goal of an attacker can be achieved, decomposed recursively as sub-goals in AND/OR relations. Set of alternative attack paths by which a top-level attack goal can be achieved.
Dependability	The ability to deliver a service that can justifiably be trusted. (calls for a justification of trust). The ability to avoid service failures that are more frequent and more severe than is acceptable (implies criteria for deciding whether a service is dependable). Dependability properties comprise availability, reliability, safety, integrity and maintainability.
Dependence	Of a system on a service; the reliance of a system's operations on a provided service.
Error	Deviation in actual system state from correct or intended state. Errors are caused by faults and give rise to service failures.
Failure	In a provided service; the service is not as intended, causing a fault in the user system.
Fault	The adjudged or hypothesized cause of an error. A fault may or may not have security implications. Also called a defect, flaw.
Information Security	*Information security* is characterized as the preservation of: 1. confidentiality: ensuring that information is accessible only to those authorized to have access; 2. integrity: safeguarding the accuracy and completeness of information and processing methods; 3. availability: ensuring that authorized users have access to information and associated assets when required. ISO/IEC 17799 Information technology — Code of practice for information security management [20]
Mitigation	Reduction in risk achieved by some action. Security risks during development can be reduced by better requirements, design, improved manufacture and test and countermeasures. During operation, security risks can be reduced by improved policies, better enactment and countermeasures.

Return on Security Investment	Benefit achieved, usually expressed in financial terms, arising from expenditure on security. (ROSI)
Risk Assessment	Assessment of threats to, impacts on and vulnerabilities of information and information processing facilities and the likelihood of their occurrence. ISO/IEC 17799 Information technology — Code of practice for information security management [20]
Risk Management	Process of identifying, controlling and minimizing or eliminating (security) risks that may affect information systems, for an acceptable cost. ISO/IEC 17799 Information technology — Code of practice for information security management [20]
Security	The ability to avoid service security failures that are more frequent and more severe than is acceptable (implies criteria for deciding whether a service is secure.) Security properties comprise confidentiality, integrity, availability, authenticity and non-repudiability. Work that involves ensuring the confidentiality, integrity, and availability of systems, networks, and data through the planning, analysis, development, implementation, maintenance, and enhancement of information systems security programs, policies, procedures, and tools. [www.opm.gov/fedclass/text/GS-2200.htm]
System	A general term indicating an entity that provides some useful functionality or service and that is developed and operated. The provided service may require the system to hold assets that are to be protected from attack. A specific IT installation, with a particular purpose and operational environment. – ISO/IEC 15408.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation. ISO/IEC 15408-1 [35]
Threat	A potential security failure. The means through which the ability or intent of a threat agent to adversely affect a system, facility, or operation can be manifest. An attack goal of a threat agent.
Threat Agent or Attacker	An individual, group or agency that has (security) attack goals against some asset.
Threat Profile	The set of threats presented to a system www.ee.oulu.fi/research/ouspg/sage/glossary/
Vulnerability	An internal fault in a system that may cause a security failure or enable an attacker to exploit some asset. An attack path in an attack tree that is insufficiently mitigated. The concept of vulnerability is meaningful only with respect to a defined threat, adjudged or hypothesized.

Appendix 2 Security Risk

The concept of risk carries different meanings in different professional communities. Project managers define risk traditionally as:

> An uncertain event or condition that, if it occurs, has a positive or negative effect on a project objective [36]
>
> An uncertain event or set of circumstances that, should it occur, will have an effect on the achievement of the project's objectives [37].

Some writers argue that project risk should be defined as an *uncertain effect* on a project's performance, rather than as a *cause* of an uncertain effect, as implied by the above. More specifically, in this view, risk is defined as [15]:

> the implications of uncertainty about the level of performance achievable by a project.

The advantage of this approach is that it opens up a wider range of issues as potential sources of uncertainty; events, conditions and sets of circumstances are viewed as subsets of the potential sources.

This second approach is more in line with the definition of risk used in the financial investment community. Risk is defined as:

> the downside variability of the level of performance achievable relative to expected outcome. [Markowitz 1959, quoted in [15].

This concept plays a key role in the mean variance approach to portfolio investment. Many project management decisions can be viewed in similar terms; good decisions lead to (1) expected performance outcomes that meet specifications and (2) variances of performances around the expected values that are acceptably small. Risk is associated with the variances, rather than the expected performances themselves.

The safety engineering community defines safety risk as [38]:

> A combination of the frequency or probability of a specified hazardous event, and its consequence.

Risks are classified in terms of severity, determined by assessed likelihood and consequences. The acceptability of a safety risk is judged with reference to the benefits provided by the system and the costs associated with risk reduction. The ALARP Principle (*As Low As Reasonably Practicable*) [39] is an example decision framework. Government agencies are often involved in such judgments. Insurance companies view such risks as *pure* risks, to distinguish them from the speculative risks borne by investors.

It follows that safety risk is associated with an expected performance level of a system (as experienced by those exposed to the risks). The variance around that expectation would be an additional source of safety risk. Attitudes to risks are role-dependent. An operator may not

view the expected performance level as a financial risk, if insurance and other provisions have been made.

It has been argued [15] that safety risk assessment would be refined by recognizing the uncertainty in both the likelihood and the consequence components of safety risk. Single figure estimates would be replaced by probability distributions.

Security risk is defined like safety risk:

> A combination of the frequency or probability of a specified security event, and its consequence.

A security-critical system may be designed and operated to achieve a specified level of expected security performance, with a variance. For the service provider, financial risk would be associated only with the variance, provided appropriate contingencies have been made to cover the consequences of expected performance. This implies that consequences of expected security events are bearable. For the service user, the planned system performance (expectation and variance) would involve exposure to a level of security risk. The user's security risk may also be translated into financial terms. The user's trade-off would be that the benefits arising from using the services outweigh the risks involved. The user of the system is not an investor in it, merely a purchaser of offered services, with no wish to be exposed to risks. Risks are considered acceptable provided they are as low as reasonably practicable, given the price of the service. Legal systems provide the ultimate test of risk interpretations where disputes arise.

For system developers and operators, security risks have characteristics that seem to challenge traditional management practices:

1. some security threats may be *learning* (or opportunistic) agents. The possibility that threats may change over time seems to be the main challenge (the degree of malicious intent is, arguably, a secondary issue). A dynamic threat Environment is difficult to predict and plan for. This leads to more emphasis on continuous, adaptive management, to maintain established security performance;
2. the components and systems infrastructure of security systems must nevertheless be developed and deployed, i.e. subject to traditional project processes. Design and implementation commitments have to be made to enable infra-structural systems to be realized. The basic challenge seems to be to enable such commitments to be made while keeping an eye on the provision of adaptive security functionality at operations level. Software components enable adaptation (modification of earlier commitments), but at cost and risk, and not in all situations;
3. vulnerabilities in some systems are discovered during operations, possibly as a result of successful intrusions etc., resulting in a dynamic, responsive characteristic in vulnerability management;
4. there is large scope for security countermeasures, especially in the information systems domain. This has led to standardization of security requirements, functions and evaluation criteria in the IT products and services and systems sector (Common Criteria);
5. the damage arising from failures in security can take a variety of forms; information-related damages may not be local to the managed system in time or space;
6. because threats are often human agents, socio-technical considerations play a part.

It has been proposed [17] to view assurance as the degree of confidence in a risk assessment, i.e. the variance around an expected security or safety risk. This approach is useful because it

recognizes the trade-off that might be available between spending resources on risk reduction and on reducing uncertainty.

The following are tentative proposals / concepts for informing decision makers about security risks.

Justification of Security Investment based on Outcome Observability

A difficulty with all preventative actions (also in the safety domain) is that the successful outcome is a null result – no problems arise. Skeptics will always question the need for investment that does not seem to have any tangible outcome.

A concept called the *observability* of a performance outcome is proposed to tackle this problem. Suppose the security manager is working to reduce the number of unauthorized, successful accesses per week to an information system. A change in procedures is introduced that increases costs but is successful in reducing the rate of unauthorized accesses. The outcome is observable and understandable to senior managers (who sign the checks) and the investment is recognised as successful.

Now suppose the security manager is aware that some of those who gain access to the IS are attempting to make money transfers that are potentially very damaging to the organization. None has yet been successful. The security manager introduces an additional firewall and reduces the associated security risk. But the *observable* measurable performance outcome is unchanged, as far as the senior manager is concerned.

To justify the investment, the security manager needs two things: (1) objective observable measures that change when the security action is taken and (2) an understandable and convincing model that causally connects the observables with security performance outcomes.

This approach calls for the development of observable performance measurements as part of introducing a change.

Generalizing this concept, we can imagine a set of increasingly 'deep' observables, that require increasingly sophisticated models that link them to end performances. We obtain a spectrum of measurements that link surface measures, close to the desired performance of a system (and deemed objective by the external observer) with deep ones, based on observables distant from the external performance (and deemed subjective by an external observer, unless the causal models are agreed upon). This concept is linked to the role of models in the cognition of learning agents.

For very high-risk events (more common in the safety domain), observable performances are more difficult to find. Near misses and similar events are very important.

Safety and Security Risk Compared

Security engineering is a type of *risk management*, and this is the main characteristic shared with safety engineering. Security engineering seeks to reduce the likelihood of future security incidents and the severity of their consequences should they occur. A range of security analytic techniques and risk reduction strategies are deployed to achieve this. The current performance of

a system, in terms of security incidents reported, is monitored and used as an input to future strategies. Measures of risk (future performance in terms of likelihood of occurrence and effects) and past-achieved performance are the core measures of security. There are similarities with safety engineering, but also differences of emphasis.

Safety engineering is mainly concerned with hazards arising from weaknesses in the system design, development and operation. Issues external to the system are considered, including Environmental effects and exposure times, but these are viewed as relatively static. The main concerns tend to focus on failure scenarios that start with component failures within the system and lead relatively rapidly (i.e. uncontrollably) to accidents, for given operational contexts. The traditional approach has placed emphasis on developing a safe product, having it certified as acceptably safe for operation, and then operating it within defined constraints. Current trends are moving towards a more through-life approach in which a Safety Management System, used in the development phase of a system, is transferred to an operational support role, providing continuous learning and improvement of safety performance.

Security engineering has to deal with a more dynamic threat Environment and this affects the risk management approach in two ways:

1. The harmful effects of a security incident may be felt across a range of different timescales and remote from the site of the security incident;
2. Threats evolve in time; concerns are dominated by threat agents that learn and adapt to system vulnerabilities (c.f. the relatively static Environmental threats to system safety).

There is greater emphasis on real time response to newly emerging threats. At the large system (and networked systems) end of the scale, predictive analyses, although an important part of planning, cannot be expected to provide for every security risk. The systems involved are too complex and the threat Environment changes too rapidly. Managing in this Environment requires feedback and resources to respond to unfolding events. These are also the characteristics of *high reliability organizations*, as explored by [40].

The challenge seems to be to commit to security design features that result in systems that are operationally feasible. The concept of a *local security environment* for an entity seems important in this regard; it enables an entity to be designed to a fixed threat specification, while placing responsibility on other parts of the system (and on operations) to maintain the local Environment.

Appendix 3 Representative Practices

Return on Security Investment Calculation

Traditional ROI calculations can be applied to security investments: the following table reflects the approach of [25].

Total		
Asset Value (AV) of an information asset	=	
		Cost of replacing information
	+	cost of replacing sw, hw
	+	cost of reconfiguration
	+	cost of loss of availability

	+	associated costs (loss of data confidentiality and integrity)
Exposure Factor (EF) of asset	=	fraction of asset value removed by a particular attack
Single Loss Expectancy (SLE)	=	financial loss expected from a successful attack
	=	AV x EF
Probability of an attack of a particular type in a one year period	=	Pr(attack)
Annual Loss Expectancy (ALE)	=	SLE x Pr(attack)
Net Present Value of a security appliance that stops the annual losses = discounted ALE over selected number of years		
Total Cost of Ownership (TCO) of a security appliance = procurement cost		
	+	non-recurring costs
	+	discounted recurring costs
Return on Investment in security appliance = (NPV of avoiding losses – NPV TCO) / (NPV TCO)		

Table 2 Traditional ROI calculation based on discounted cash flows, from [25]

ISO/IEC 15408 Common Criteria

The Common Criteria (now established as ISO/IEC 15408 [ISO/IEC, 1999 #485]) provide a framework for the independent evaluation of the security performance of IT products and systems. The evaluation process involves:

1. the identification of security objectives and requirements, constituting a *Security Target* (ST);
2. the optional use of a standard *Protection Profile* (PP), representing typical sets of security functions;
3. the identification of a *Target of Evaluation (*TOE);
4. the evaluation of the TOE against the PP and security requirements;
5. several evaluation levels (EAL 1 through EAL 7), providing different levels of evaluation rigor, and therefore confidence in the performance.

The Common Criteria (CC) approach provides a means for a system developer to establish assurance that a product or system meets identified security performance standards. Security risk is reduced by assessment against internationally agreed performance standards. The CC framework is built around catalogs of PPs and evaluated products. Extended requirements and evaluation criteria, not in the standard models, can be included.

The CC approach has been used as a guide in developing the proposed PSM model – particularly the concept of integrating assured components into assured systems. Certified components and systems may still contain vulnerabilities, so additional security risk management would remain necessary. Defense systems may require stronger assurance techniques than ‘standard’

commercial IT applications. A security process following a CC approach would present measurable artifacts and attributes (e.g. scope and progress of assurance activities, costs, security risk reductions and improvements in confidence intervals of these). The assurance activity is itself a form of measurement.

Security Process Maturity: ISO/IEC 21827 SSE-CMM

The SSE-CMM [9] comprises the following Process Areas:

Process Areas	Goals
PA01: Administer Security Controls	Security controls are properly configured and used.
PA02: Assess Impact	The security impacts of risks to the system are identified and characterized.
PA03: Assess Security Risk	An understanding of the security risk associated with operating the system within a defined environment is achieved.
	Risks are prioritized according to a defined methodology.
PA04: Assess Threat	Threats to the security of the system are identified and characterized.
PA05: Assess Vulnerability	An understanding of system security vulnerabilities within a defined environment is achieved.
PA06: Build Assurance Argument	The work products and processes clearly provide the evidence that the customer's security needs have been met.
PA07: Coordinate Security	All members of the project team are aware of and involved with security engineering activities to the extent necessary to perform their functions.
	Decisions and recommendations related to security are communicated and coordinated.
PA08: Monitor Security Posture	Both internal and external security related events are detected and tracked.
	Incidents are responded to in accordance with policy.
	Changes to the operational security posture are identified and handled in accordance with the security objectives.
PA09: Provide Security Input	All system issues are reviewed for security implications and are resolved in accordance with security goals.
	All members of the project team have an understanding of security so they can perform their functions.
	The solution reflects the security input provided.
PA10: Specify Security Needs	A common understanding of security needs is reached between all parties, including the customer.
PA11: Verify and Validate Security	Solutions meet security requirements.
	Solutions meet the customer's operational security needs.
PA12: Ensure Quality	Process quality is defined and measured.
	Expected work product quality achieved.
PA13: Manage Configurations	Control over work product configurations is maintained.
PA14: Manage Project Risk	Risks to the program are identified, understood, and mitigated.
PA15: Monitor and Control Technical Effort	The technical effort is monitored and controlled.
PA16: Plan Technical Effort	All aspects of the technical effort are planned.
PA17: Define Organization's Security Engineering Process	A standard systems engineering process is defined for the organization
PA18: Improve Organization's Security Engineering Processes	Improvements to the standard systems engineering process are planned and implemented.
PA19: Manage Product Line Evolution	Product lines are evolved towards their ultimate objectives.
PA20: Manage Systems Engineering Support Environment	The systems engineering support environment maximizes process effectiveness.
PA21: Provide Ongoing Skills and	The organization has the skills necessary to achieve project and

Knowledge	organizational objectives.
PA22: Coordinate with Suppliers	Effective suppliers are selected and used.

There are similarities with the four sub-domains proposed in the PSM model; differences reflect different choices about how to group activities.

Safety and Security Extensions to the iCMM and CMMI Models

Safety and security extensions to the iCMM and CMMI models have been published recently [23]. A *Safety and Security Application Area* (AA) has been introduced that identifies goals and standards-based *Application Practices* (APs) directed at establishing and maintaining a safety and security capability, define and manage requirements based on risks attributable to threats, hazards, and vulnerabilities, and assure that products and services are safe and secure throughout their life cycle. Goals and practices of the application area are:

Goal 1 An infrastructure for safety and security is established and maintained
- AP 01.01 Ensure Safety and Security Competency
- AP 01.02 Establish Qualified Work Environment
- AP 01.03 Ensure Integrity of Safety and Security Information
- AP 01.04 Monitor Operations and Report Incidents
- AP 01.05 Ensure Business Continuity

Goal 2 Safety and security risks are identified and managed
- AP 01.06 Identify Safety and Security Risks
- AP 01.07 Analyze and Prioritize Risks
- AP 01.08 Determine, Implement, and Monitor Risk Mitigation Plan

Goal 3 Safety and security requirements are satisfied
- AP 01.09 Determine Regulatory Requirements, Laws, and Standards
- AP 01.10 Develop and Deploy Safe and Secure Products and Services
- AP 01.11 Objectively Evaluate Products
- AP 01.12 Establish Safety and Security Assurance Arguments

Goal 4 Activities and products are managed to achieve safety and security requirements and objectives
- AP 01.13 Establish Independent Safety and Security Reporting
- AP 01.14 Establish a Safety and Security Plan
- AP 01.15 Select and Manage Suppliers, Products, and Services
- AP 01.16 Monitor and Control Activities and Products

The proposed measurement framework is broadly compatible with the recommendations of the Application Areas. For example, AP 01.06, 07 and 08 place risk assessment at the center of safety and security practice, as does the proposed measurement framework. AP 01.11 involves the objective evaluation of products, covered by the *assurance* and *performance* measurements of the proposed framework. The concept of a 'managed domain' proposed in this paper is similar to an *Application Area*, or a set of *Application Practices*.

An intention of the *managed domain* concept is that it should make minimum assumptions about how work is organized. One aspect of security and safety performance relates to awareness and flexibility of response. It is assumed that these aspects are addressed in a managed domain by having resources deployed that can respond to unexpected events. The classic process maturity

view is appropriate when processes are repeatable and attention can be directed towards evolutionary improvements in efficiency. A managed domain may then be treated mainly as a process or set of processes.

ISO/IEC 17799 Information Security

Widespread concerns about the security of general business IT systems has resulted in the development of standards in this field [41]. *Information security* is defined as the preservation of confidentiality, integrity and availability of information:

- *Confidentiality*; Ensuring that information is accessible only to those authorized to have access;
- *Integrity*; Safeguarding the accuracy and completeness of information and processing methods;
- *Availability*; Ensuring that authorized users have access to information and associated assets when required.

ISO/IEC 17799 provides a code of practice for information security management under the following headings:

- Security Policy
- Organizational Security
- Asset Classification And Control
- Personnel Security
- Physical And Environmental Security
- Communications And Operations Management
- Access Control
- Systems Development And Maintenance
- Business Continuity Management
- Compliance

The ISO standard views security requirements as arising from three sources:

1. assessment of risks to the organization;
2. legal, statutory, regulatory and contractual requirements;
3. particular set of principles, objectives and requirements for information processing that an organization has developed to support its operations.

Security risks are reduced by the implementation of *security controls* (types of action, as defined in this paper). C.f. NIST SP 800-53, FDIS ISO/IEC 27001.
An associated standard, BS 7799-2:2002 *Information security management systems - Specification with guidance for use* [42] is directed at business managers and defines the concept of an *Information Security Management System* (ISMS). A process-based approach for establishing, implementing, operating, monitoring, maintaining and improving the effectiveness of an ISMS is described. This standard uses the PDCA cycle [43] as a reference for the management of the ISMS. This concept has been adapted for the proposed model.

NIST has developed measurement guidance with reference to security program maturity in the IT domain [5].

It is intended that the proposed measurement approach is compatible with these standards. Their main limitation is that they do not engage with the development or operation of secure systems at

detailed technical levels. An objective of the proposed measurement approach is to achieve 'vertical integration' between technical risk assessment and management decision-making.

Risk Management Tools

Risk management tools include [28]:

1. CRAMM
2. FIRM
3. SARA and SPRINT
4. COBRA
5. OCTAVE [44]

Tracking Particular Security Risks

The proposed measurement approach includes the concept of performance and risk tracking systems in each of the four sub-domains identified, combined with tracking of integrated performances and risks. The following are applicable to each sub-domain:

1. counts of identified risks in risk tracking systems and their time-evolving status ('rows' in the tracking systems);
2. measurements associated with performance observables in performance tracking systems;
3. resources deployed and progress of actions;
4. scopes of plans, risks and awareness;
5. outputs of tasks;
6. outcomes of tasks risk management and performance;
7. assurance task progress, costs;
8. competence deployed.

The use of tracking systems, analogous to the *Hazard Tracking System* used in safety engineering [4], and risk tracking systems used in project risk management, seems an obvious approach.

Threat Environment Management

This view of security involves measurement and actions within the entity Environment and the triggering of actions in the other sub-domains. Actions available in the Environment would depend on the type of entity involved. For publicly accessible IT systems, actions might be directed at reducing motivation and monitoring usage. Defense systems operate under wider permitted ranges of action. There is a link with *Damage Management* in the area of recovering damages, for example, by using legal systems. Some threats (e.g. natural threats) are internal to the entity and have similarity with safety concerns.

The monitoring and assessment of attackers is the principal role, enabling responses to be made in system design and operation. During the development phase, emphasis is on predictive assessment to inform design commitments. During operations, emphasis is on rapid detection and response within the 'space' created by the designs.

A Threat Tracking or Management System would enable counts of numbers of actual and potential attackers in different categories and the status of actions that have been triggered by them. Examples of categories include:

1. Potential/ actual status; success of attacker (in penetrating the security assets, deriving benefit, causing damage);
2. Capability of threat agent;
3. Intention of threat agent;
4. Numbers of potential attackers in each type;
5. Priority indicator, based on risk (involves other sub-domains);
6. Scope of threat (in terms of parts of system attacked, identified vulnerabilities);
7. Number of threat vectors in a threat type; (e.g. ADDER score [29])
8. Time rates of appearance and capability/ learning rates.

Table 6 shows example sketches of tracked counts of threats, vulnerabilities and events.

Vulnerability Management

This view of security involves actions within the entity itself, including both entity design and operations/ policy actions. The designs and policies influence and constrain the actions available in the Event and Damage sub-domains.

Many different kinds of action are possible, depending on the type of entity involved, and whether the context is a development project or an operational system/ organization. The actions of this sub-domain are generally preventative (pro-active) in nature, in terms of the delivery of security. For 'standard' IT product and systems, well-recognized countermeasures to known kinds of attack have been developed. The Common Criteria approach provides an internationally recognized process for independently evaluating the assurance of IT products and systems against standard security functions. Such an approach enables a market in evaluated standard security function products. Assurance levels provide confidence in the security performance of products and systems and are one way to reduce risk. Other kinds of non-standard system will require more specific analyses and assurances.

Some applications have well-developed approaches to vulnerability management. For example, a Vulnerability Management System (VMS) is described as assigning one of four severity categories to a *Potential Discrepancy Item*:

- Category I findings are any vulnerability that provide an attacker immediate access into a machine, gain super-user access, or bypass a firewall;
- Category II findings are any vulnerability that provides information that has a high potential of giving access to an intruder;
- Category III findings are any vulnerability that provides information that potentially could lead to compromise;
- Category IV vulnerabilities, when resolved, will prevent the possibility of degraded security.

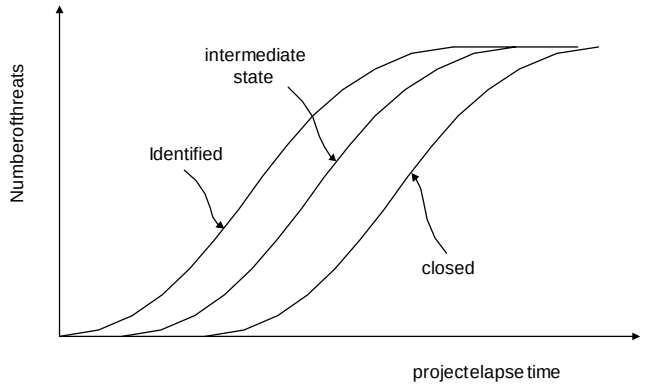
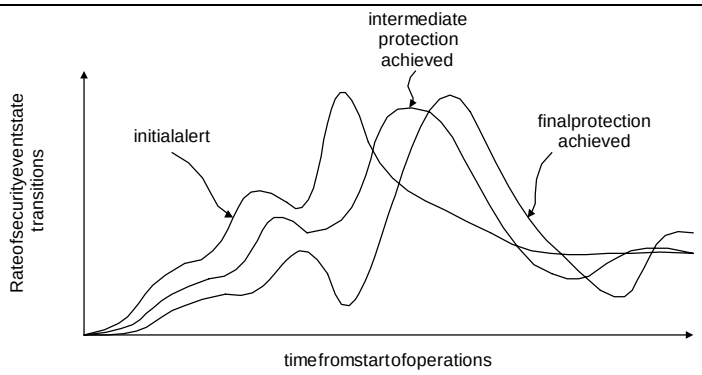
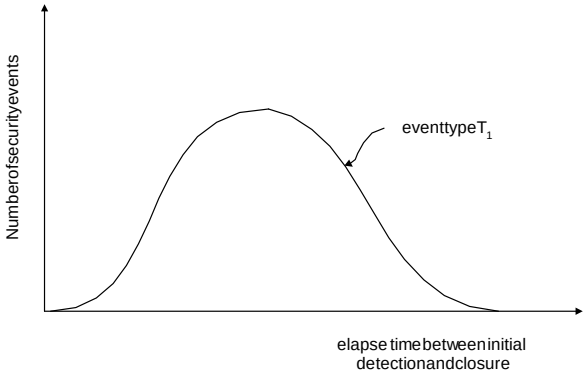
Measure	Tracking
Identified threats to an entity; plot of number of threats against project elapse time. Threats are managed in terms of initial detection, intermediate protection and final closure. Applicable to development phase; number of threats levels off at a maximum. Separate charts for threats of different severities. Similar charts would be used for high-priority vulnerabilities.	
Rate of security event state transitions during operations phase. Rates of initial detection, intermediate protection and closure should be equal, asymptotically. Areas under curves should be equal.	
Statistical data – time to 'process' security events i.e. elapse time between initial detection and closure, for many events of type T_1.	

Table 3 Example tracking of security threats and events

A Vulnerability Tracking System would enable counts of numbers of identified vulnerabilities in different categories and the status of risk mitigation actions triggered by them. A vulnerability being managed by a Common Criteria approach would be tracked with reference to a management system tailored to the tasks involved.

Many techniques and technologies are involved in removing vulnerabilities and reducing associated security risk. Examples in the software security domain include:

1. Language-based security
2. Operating Systems Security
3. Secure Middleware
4. Malicious Code Detection
5. Intrusion Tolerance

6. Trust Management
7. Program Analysis

Vulnerabilities in an entity are reduced by two means: (1) application of known best practice methods, tools etc., based on shared domain understanding and (2) identification of particular vulnerabilities for the entity of concern. Explicit identification and tracking of vulnerabilities is directed at the second of these.

Security Event Management

This view of security involves actions that respond to attack events (and actions that prepare for them). The detection and annunciation (signaling) of events is included in this view. Security functions of interest in this sub-domain are those that involve fast response to events. Actions arising in this sub-domain include preventative / pro-active and reactive actions.

Security Event Tracking provides a source of objective performance measurement. The form of security event will vary depending on the type of entity involved. Many events will be of in the form of an attack scenario; a successful intrusion will involve a sequence of states or conditions, some of which might be observable. The actions taken in response to security events are also measurable.

A security event may be modeled as a multi-stage scenario; this can comprise deterministic and probabilistic steps. Measurements may be available to detect the transition of an entity or threat agent to an intermediate state i.e. a state prior to a successful intrusion or an occurrence visible to an end-user. Such observable events enable assessment of security performance and risk reduction based on objective data, but without necessarily incurring actual security breaches. Probability tree representations support the use of event detection to revise risk assessments. Security actions triggered by such measurements can be represented as modifications to event trees.

Damage Management

This view of security involves actions that respond to damage arising from attack events (and actions that prepare for managing damage). This domain also covers potential damage assessment for the purposes of assessing the value of the security entities. Also of interest is the design of systems and policies (e.g. interfaces, boundaries, role/responsibilities) that can reduce the risk of damage propagation, given an intrusion. Damage effects may not lie exclusively within the fields of action and measurement of the other security sub-domains, depending on the type of entity involved.

A Damage Tracking System would enable the recording of the effects of successful attacks, responses to them and the achieved outcomes. The damage sustained by a system or organization arising from security attacks, whether intentional, opportunistic or accidental, is the final objective test of the success of investments in security.