What Makes A Good Information Security Policy: A Preliminary Framework For Evaluating Security Policy Quality.

S.B. Maynard [1], A.B Ruighaver [2]

Department of Information Systems,
University of Melbourne, Australia
[1] seanbm@unimelb.edu.au
[2] anthonie@unimelb.edu.au

Abstract

The level of quality of security policy is rarely discussed in any great depth in literature. Consequently, organizations often find it difficult to define quality in security policy terms. As the security policy field matures, however, the concept of quality is becoming more important for many of these organizations. This paper presents a model of security policy quality factors which has been developed from software development quality and data model quality. It briefly discusses the importance of these issues to organizations and gives some insight to their relevance by presenting some initial case study results.

1. Introduction

There is little information in security policy literature about quality measurement or improvement of policy. Instead, the literature concentrates on suggesting what should, or should not be included in a security policy (McMillan 1998, Tudor 2001, State of Oregon 1998), and often neglects to take into account other factors such as the psychology of stakeholders (Kabay 1994), or the security culture within the organization (Chia, Maynard and Ruighaver 2003).

The quality of strategic information security policy is concerned with improving current practice. In other words, investigating how organizations develop, implement, use and maintain strategic security policies and attempting to change organizational practice to improve the overall quality of the resultant policies. As with many organizational initiatives, this will be influenced by stakeholder behavior and beliefs. Importantly, acknowledging that different stakeholders will hold different beliefs about quality, and allowing the addressing of these beliefs will be critical in improving security policy quality.

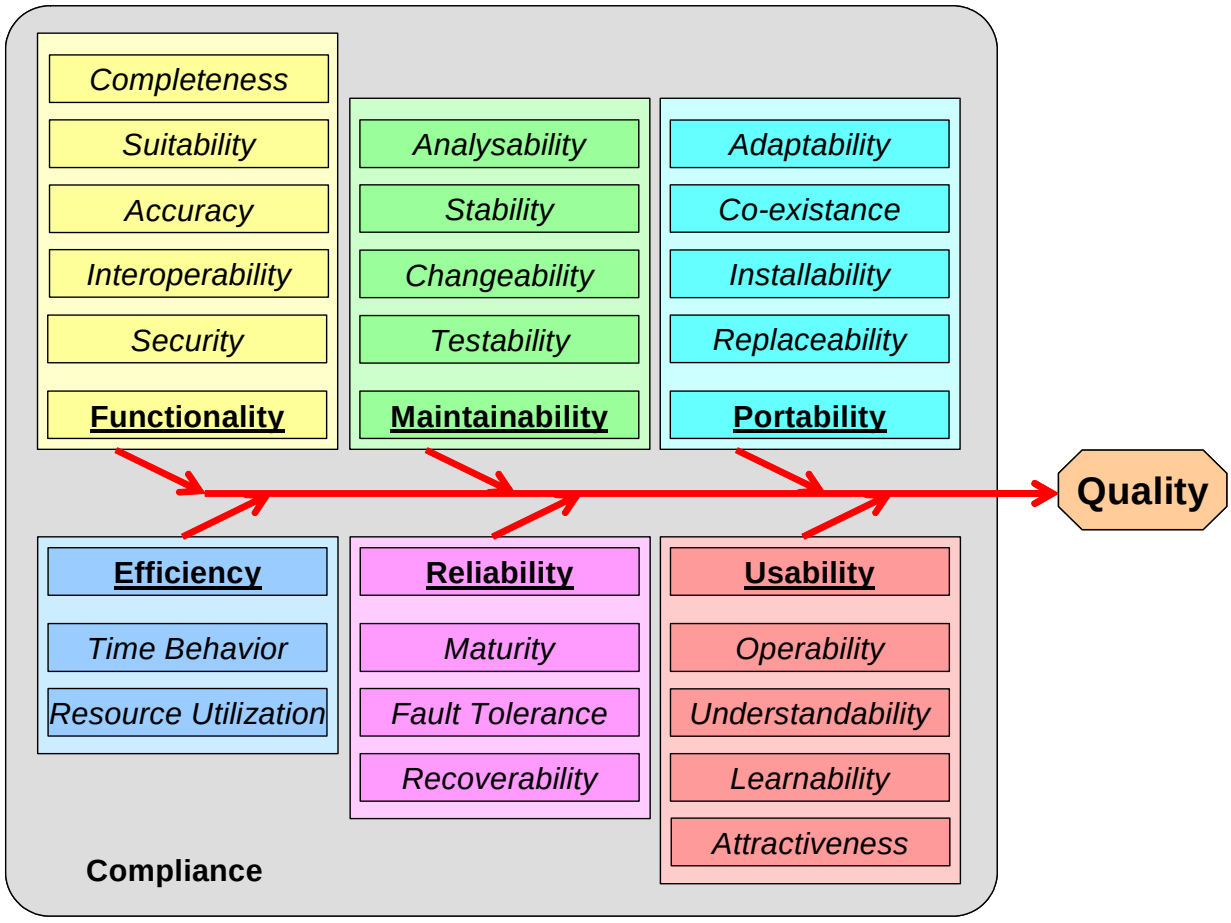
This paper focuses on the development of a set of factors that can be used to determine the quality of an information security policy within an organization. The aim of the research is to determine what factors contribute to the quality of information system strategic security policies for an organization. We did a thorough theoretical search through the literature concerning security research, to try and identify those factors that were deemed to be important to quality. Although a number of ideas of quality were formed, unfortunately, it was impossible to produce a framework for quality from this research. We next did a search of other IT related fields, including auditing, software development and data quality and found many richer frameworks for quality in these areas. This paper reports on the adaptation of these frameworks to the information security policy domain.

The research draws on two research disciplines, data model quality and software quality, to present an initial framework for the quality of information security policy. Whilst a number of case studies are being undertaken as part of this research, we thought it more important to concentrate on the presentation of the quality framework, rather than on presenting the full case analysis as this is still continuing. We do,

however, provide some insights to the quality framework through using some of the interesting findings from the case studies which we wish to present in full elsewhere.
We will now introduce a framework for security policy quality, discussing each of the quality factors and how they relate to security policy. We then provide some insights as to whether these quality factors are found in organizations using some initial results from the case studies currently being conducted.

2. A Framework for Security Policy Quality

As mentioned earlier, this research utilizes the fields of data model quality and software quality to develop a framework for the quality of information security policy (see Figure 1). A well accepted model of software quality, proposed by McCall, Richards and Walters 1977, consists of eleven dimensions of quality across three categories. Whilst these dimensions are fairly inclusive, issues such as perception and aesthetics were not addressed (Tyrrell 2000). Other, more inclusive models of software quality are the Australian and New Zealand standard 4216 (Standards Australia 1994) and ISO 9126 (International Standards Office 2005). They list 6 main software quality factors that take into account those defined earlier by McCall et al. (1977).



Quality Factors are represented in **bold underlined**
Characteristics are represented in *italics*

Figure 1: A Framework for the Quality of Information Security Policy

From the perspective of data model quality, Moody and Shanks (1998b) propose a model for evaluating and improving the quality of entity relationship models. Their model was developed with reference to various discipline areas including data modelling, software quality, and architecture. They propose 7 factors which also are similar to the software quality factors identified above. Whilst there are potentially many other models that can be utilized (Arthur 1993, Dunn 1990, Humphrey 1989, Budgen 1994, Saunders and Curran 1994, Yourdon 1992), their inclusion would be superfluous as they do not add any factors or concepts that have not already been considered.

In comparing these quality models it is interesting to note that whilst similar characteristics exist between them, the focus of the application of those characteristics in some cases is quite different. Looking at Moody and Shanks' (1998b) model, the application of the quality characteristics is purely on the output of the model building process, "the E-R model", whereas with software quality, there is also a focus on the evaluation of the process undertaken to develop the software. Furthermore in software quality there is an overarching concept of compliance for each of the quality factors that takes into account any issues focusing on standards and conventions.

In the remainder of this paper we present the framework of quality factors as shown in Figure 1. We discuss each of the six quality factors breaking them into a number of characteristics which are used to describe that particular factor. Although the model presented here is extensive, it is likely that some of the factors and characteristics may not be currently used within organizations for the evaluation of security policy quality because, as previously mentioned, the security policy field is relatively immature. Likewise, it is possible that further characteristics specifically related to security policy quality may be identified in the currently still ongoing case study research.

3. The Quality factors and measures

3.1. Functionality

The functionality factor, as defined in the Software Quality standards, focuses on the existence of the functions required for software to perform its tasks as defined by the stated or defined needs for that software. Moody and Shanks (2002) use a similar concept, which they call completeness, to determine if data models contain all the information to meet the user requirements. In the case of software quality, and also for security policy quality, functionality however is more than just completeness.

For software quality, there is an inherent assumption that the software is designed based on documented software requirements that have a high quality. Likewise, for security policy quality we should assume that the organization has developed a set of strategic policy requirements which adhere to the organizations mission statement and the business strategies of the organization. However, in practice few organizations will have explicitly identified what their strategic security requirements are, separately from the policy.

In order to determine the functionality of the information security policy a number of characteristics of functionality can be used. From analyzing the software quality standards, Moody and Shanks (1998b) and McCall et al. (1977) the following characteristics will be used for determining functionality: suitability, accuracy, interoperability, completeness and security. Each of these characteristics will now be described.

3.1.1. Suitability

The suitability characteristic, in software quality, focuses on whether a set of functions is present and the appropriateness of those functions in a software product (ISO9126 2005). In security policy, the concept of a given set of functions being present is not new and literature often discusses the structure of security policy and how to determine what should be in a security policy. For example, a security policy must

include details about who (a person or more likely a position title) is responsible for parts of the policy and that it should be clear about actions to be taken when breaches occur.

In information security policy quality, the suitability characteristic focuses on the appropriateness of the information security policy for the organization. For instance, if the organization has explicit security goals, specifically derived for this particular organization, then the suitability of the policy could be determined by ascertaining whether the security policy meets the stated goals or objectives of the organization and does so in an appropriate way (LeClerc 2001, Fulford and Doherty 2003). However, in many organizations, security goal's are not derived specifically for the individual organization, or may never be explicitly defined at all (Control Data 1999). In such cases, stakeholder opinions could be used to determine if the security policy is suitable. Other aspects of suitability may be that the organization does not enforce the security policy, or perhaps certain aspects of the security policy, in which case why are those aspects of the policy there. Are they suitable?

3.1.2. Accuracy

Accuracy is commonly defined as whether or not something conforms to the truth, whether it is free from error and whether it conforms to standards. In the ISO9126 standard, accuracy has been defined as the attributes of software that bare on the provision of right or agreed results or effects. McCall et al. (1977) previously used correctness to describe a similar phenomenon: the extent to which the program satisfies its specification and functional objectives.

For security policy, the concern is whether or not the policy is performing as people think it should, or in other words whether the policy statements are having the desired effect. The accuracy characteristic focuses on the extent to which the information policy is defined according to requirements. If an organization has explicit stated security goals, accuracy could be determined by whether or not the stated goals were really the goals of the organization, or were just the explicit written gown goals, where the actual organization security goals were different. Once again where explicit goals are not known, it would be the opinions of stakeholders that could be used to determine if the policy was accurate.

Another issue dealing with accuracy for security policy is the difference between the written policy and the "real" policy. Often in organizations there is a written policy that is there to ensure that the policy is documented, however, the way in which the organization operates means that what really happens may not necessarily correspond to the policies of the organization. So another way to determine accuracy of the policy is to observe whether or not the written policies are implemented and are operating as they were intended.

Finally, accuracy may also deal with other areas where differences occur between what is documented in the policies and in practice within the organization. For example, whether responsibilities in the policy are actually true in real life (Warman 1992, Regan 2001), whether resource ownership as documented is in fact correct in real life (LeClerc 2001) and whether the risks the policy states are addressed are in fact addressed (Anton and Earp 2001).

3.1.3. Interoperability

The interoperability characteristic focuses on how well policies interact. In data modeling, Moody and Shanks (1998b) describe this as integration: how well the data model fits within the rest of the organizational data. From software quality this is the extent to which the software interacts with other systems (McCall et al. 1977).

From the information security policy quality perspective the fit of the information security policy within the organization is a good indicator of interoperability. This may focus on whether policies do not contradict each other, and that their wording and structure is similar, or perhaps on whether the development process used for security policy development is similar, or the same as that used for other policy development within the organization.

In practice, with a focus again on explicitly stated security goals, to determine the interoperability of the security policy, the alignment between the security goals and the organizations business objectives could be investigated. Like previous characteristics, when the organization does not have explicit security goals, different stakeholders could be used to determine interoperability.

Furthermore, it is important for the security policy to be consistent and compatible with other policies within the organization (Guttman and Bagwill 1999).

3.1.4. Security

In software quality, as defined by the ISO 9126 standard, security is concerned with the construction of the software to ensure that the incidence of unauthorized access to programs and data is reduced. McCall et al. (1977) offers a similar construct that they call integrity which focuses on reducing the unauthorized access to software and data access and to ensure that data will be verifiable throughout its life.

Of all the aspects of quality this one is probably the most difficult to describe for security policies. From the security perspective though, security can be thought of having 3 main facets: confidentiality, integrity and availability. For confidentiality, issues such as is the policy written in such a way that it can be published, and whether or not policies could be re-written to make them less sensitive are possible areas of interest for organizations. Integrity is concerned with whether changes to one part of the policy affect other areas of the policy, in other words, how well is the policy structured. However, this is not of a concern here as it is covered under stability.

Availability, from a security policy context, focuses on the trade off between having the policy available to people or unavailable. Regan (2001) states that policies that have significant technical security information should be safeguarded. This is important, but often there is a tendency with security policy to consider that all the information contained within it is too sensitive and thus deciding not to make it available, even to those people who it should be available to. It is a better solution to remove the technical information from the policy and to allow the policy to be available to those who need it.

3.1.5. Completeness

From the descriptions of the characteristics of functionality within software development it is apparent that the concept of completeness has not been fully described by any of the characteristics. Interestingly when addressing data model quality Moody and Shanks (2002) added completeness to their model to address similar issues as found here. As a result, it is believed that the model of security policy quality may benefit from the inclusion of completeness.

Whilst there are some aspects of completeness discussed in suitability, it is not clear that this captures the essence of completeness within security policy quality. From the suitability characteristic perspective, determining whether a set of functions is present only partially satisfies a security policy's completeness. Where functions are not present it must be made explicit why they are not considered. Completeness goes further than this and in some instances could be thought of as the breadth of coverage of the policy. As a result, while there is some overlap the characteristic completeness is included in the quality model.

3.2. Reliability

The reliability factor as defined in the AS/NZS 4216 standard focuses on the capability of software to maintain performance given stated conditions. McCall et al. (1977) defines reliability as the extent to which the software is expected to carry out its functions with the required precision and that it is available when it is required. From the perspective of information security policy reliability deals with whether the security policy functions consistently and is able to reliably perform its intended tasks (which may vary from organization to organization). Based on the AS/NZS 4216 standard for software quality the

characteristics of reliability are maturity, fault tolerance and recoverability. Each of these characteristics will now be described.

3.2.1. Maturity

Maturity is commonly defined as the state of being fully grown or developed. In software quality, the ISO 9126 standard defines maturity as the frequency of failure resulting from faults in the software. This is making an assumption that the more mature the software becomes the less likely it would fail as faults will have been detected and fixed. From an information security policy quality perspective, the concept of maturity can be viewed from two perspectives. From a policy lifecycle viewpoint, an assumption can be made that if a security policy has gone through a lifecycle several times, then it must have a higher quality. More importantly, maturity may be determined by the decreasing number of security incidents that result in changes being made to the policy, in essence this is determining how many faults there are in the policy.

For instance, within an organization that has a good security culture one would expect to see a mature security policy that undergoes few changes as the result of incidents and that includes the defined security principles of the organization. One would expect that the security policy has procedures in place to ensure its validity and currency.

3.2.2. Fault Tolerance

Fault tolerance is described in software quality as the ability of the software to continue operations when things start to fail. In practice, attaining fault tolerance often requires redundancy in many parts of the "system". From an information security policy perspective, the concept of fault tolerance has a focus on redundancy within security policies and in those controls implemented to enforce policy.

In security, fault tolerance is taken seriously and often the concept of overlapping controls is implemented within organizations (Whitman and Mattord 2005). Likewise in strategic security policy it is vitally important that there is no one single point of failure. For instance, within an organization where a comprehensive set of security policies exist, if one policy fails then the next higher level policy in the policy structure will be utilized. Furthermore, there should be evidence of overlapping controls to handle situations where key personnel are ill, or on holidays. This is not suggesting overlapping responsibilities, but does focus on oversight issues.

3.2.3. Recoverability

A common understanding of recoverability is the ability for something to be restored to a normal state. In software quality, the ISO 9126 standard defines recoverability as "the capability of the software product to re-establish a specific level of performance and recover data directly affected in the case of failure". In software development when there is an incident it is likely that data integrity may be compromised and recoverability in this case means getting the data back.

For security policy however, unlike in software development, when an incident occurs the policy its self is not usually damaged. So a direct comparison between software development and security policy is not possible. However, there are certain aspects of recoverability that are covered in the changeability characteristic, for instance when you recover from a security incident and must change the policy.

3.3. Usability

An important construct in assessing the quality of most things is to look at their usability. In software quality, McCall et al. (1977) defines usability in terms of the time and resources required to effectively use the software. The ISO9126 standard uses similar terminology in its definition, and focuses on the effort needed to use and on the individual assessment of use by a set of users.

From the information security policy perspective, the usability of the security policy focuses on mainly the implementation and understandability of the policy. A good description of usability is the effort required for constituents to create, use and update the information security policy. The communication of the policy (Guttman and Bagwill 1999) is important if stakeholders are going to be able use the policy. Furthermore, the policy is only going to be useful if the stakeholders are aware and educated about the policy (Leinfuss 1996, Control Data 1999). The characteristics used to determine usability are understandability, learnability, operability and attractiveness (ISO 2005, Moody and Shanks 1999b). Each of these characteristics will now be described.

3.3.1. Understandability

Moody and Shanks (1998b), in terms of data modelling, define understandability as the ease with which the concepts and structures of the data model can be understood by others. Likewise, in software quality, understandability has a focus on the effort required by users to recognize the logical concept and its applicability of the software.

In information security policy quality understandability can be described as the ease of comprehensibility of the information security policy. In practice, within a large organization this may be extended to the comprehensibility of each policy and of the linkages between the policies. Understandability may be determined through examining employee's understanding of security and of their responsibilities implicitly or explicitly described within the policy. Also, depending on the roles and responsibilities of a stakeholder, the understandability of the strategic security policy could relate back to which controls are implemented and for what reasons. Clearly a misconception about any of these issues may result in a security breach within the organization.

Furthermore understandability of the policy for stakeholders will be affected if the policy uses jargon, or technical terminology (Warman 1995, McMillan 1998, Regan 2001), or if it contains ambiguities which make the policy unclear (Tudor 2001, Control Data 1999, State of Oregon 1998).

3.3.2. Learnability

The ease of being able to fix something in memory is a common definition for learnability. From a software quality perspective, the ISO9126 standard focuses on the amount of effort required for the user to learn the application in terms of input, output and operation.

In terms of information security policy, learnability focuses on the effort constituents need to use to learn the information security policy so that they are able to abide by it. In practical terms within an organization, one way of determining learnability is through the training of new stakeholders and focusing on how they learn the policy. This characteristic takes into account awareness training, but goes further to encompass the simplicity or complexity of the security policy. Learnability is affected by the understandability characteristic to a degree because if understandability is a problem for stakeholders, then it is likely that they will have problems learning the policies and their consequences.

3.3.3. Operability

Operability, in information security policy terms, concerns how easy psychologically from the individual's perspective it is to work within the policy guidelines once the policy has been learnt and understood. In software quality, this characteristic was used to test the effort of users to operate the software in question. Failure to provide a simple understandable security policy will ultimately cause it to be hard to abide by and therefore may increase the chances of it being breached as employees will consciously or sub-consciously not adhere to parts of the policy. Also, developing a policy that asks stakeholders to go against normal psychological norms will cause problems from an operability perspective as they will be more likely to follow the norms, rather than abiding by the policy (Kabay

1994). Other issues that may impact on the operability of the policy are how the policy impacts on a stakeholders work habits, and how difficult the policy is to enforce.

3.3.4. Attractiveness

The concept of attractiveness is proposed by ISO 9126 and is the ability for the software to be attractive, in terms of interface design, to the user. In terms of software quality, the attractiveness of the interface is often paramount to the success of the product and, understandably is a critical issue to the success of many commercial products. However, for security policy there, in essence, is no real "interface" as such, except for the paper on which the physical representation of the policy is written. An exception to this may be where the policies are available to stakeholders via an organizations intranet, in which case the presentation of the policy may have some impact on stakeholder's perceptions of quality. In practice, whilst many organizations make their policies available via their intranets, they often do so in the form of an Adobe PDF or Microsoft Word file, rather than making use of HTML or equivalent and as such the attractiveness of the policy will be the same as the hard copy document. If organizations were to make use of HTML technologies, policies could be designed to be more interactive, rather than just being paper based and in such cases policy attractiveness (let alone any of the other usability characteristics) will be more important to stakeholders.

3.4. Efficiency

The concept of efficiency has been defined in software quality terms by McCall et al. (1977) as the amount of resources and interactions required to perform the required functions. This was extended in the AS/NZS 4216 standard and was defined as the relationship between the level of performance of the software and the amount of resources used. From an information security policy perspective, efficiency focuses on the cost and effort of developing and implementing the security policy from an organizational perspective in terms of the amount of time and resources required. Efficiency can be viewed based on both time behavior and resource utilization (ISO 9126 standard). Thinking of security policy in this manner is thought to be rare, if it occurs at all. This may be attributed to the immaturity of the field. Also, in terms of resource and time behavior, it is not clear how to write high quality policies. Therefore, evidence is not expected to be observed of these characteristics in our study. Even so, these two characteristics will be discussed and issues presented that will have bearing on the quality of the policy development process and of the end product.

3.4.1. Time Behavior

The focus in software quality on the time behavior characteristic is on the response time of the software and on throughput rates in the functions that it performs. In information security policy quality, time behavior focuses on the time required for policy development, implementation and ongoing upkeep. In practice, within organizations issues such as the ease of being able to update the policy or the ease with which the policy can be implemented in terms of time are useful indicators of time behavior.

An interesting issue that will also relate to resource behavior is when an organization can implement policies, or policy controls in more that one way to protect their assets. A decision will be made by the organization of which strategy to implement, which in turn may have implications on the time required for development, implementation, and in turn resource behavior.

3.4.2. Resource Utilization

Resource utilization has a focus on the amount of resources required within the organization for conducting a task. In terms of software quality this means the amount of resources required for the software to carry out its functionality. Likewise, in strategic security policy resource utilization focuses on the amount of resources required to implement, enforce, maintain and use the policy. The amount of

duplication of responsibilities and the amount of resources required for enforcing the policy may be useful within organizations when determining the amount of resources required for the information security policy. Furthermore, the ease of auditing the policy and of training regarding the policy may impact on resource utilization.

3.5. Maintainability

The maintainability factor in information software quality according to McCall et al. (1977) focuses on the ease to maintain the system. In the ISO9126 standard this is similarly defined focusing on the effort required to make specified modifications to the software. From an information security policy perspective, maintainability is the effort required for upkeep of the information security policy. Based on the software quality standards and Moody and Shanks (1998b) work in data modelling, the characteristics for maintainability are analyzability, changeability, stability and testability. Each of these characteristics will now be described.

3.5.1. Analyzability

Analyzability, from the software quality perspective focuses on the ability to diagnose deficiencies or causes of failures, or identify the parts of the system to be modelled. More scientifically, analyzability is defined as the ability to examine methodically by separating into parts and studying their interrelations. Moody and Shanks (1998b) carefully define simplicity in data model quality which also relates to the analyzability of the data model. They define it as "The data model contains the minimum possible entities and relationships". It is not, as they point out whether or not the model is expressed well or understandable this is *understandability*.

For information security policy quality, the characteristic analyzability is the ease of examination for deficiencies in the information security policy. For instance, in a large organization the analysis may focus on the risks to the organization and whether the information security policy and its associated security measures mitigate those risks.

3.5.2. Changeability

Changeability is an important characteristic of maintainability within any project within organizations as it ensures that the project can adjust with the ebbs and flows of the organization. In software quality changeability focuses on the effort required to change software due to modifications, fault removal, or to cater for changes in the environment. Likewise in data modelling there is a characteristic called flexibility (Moody and Shanks 2002) which is the ease to which the data model can cope with business change. In information security policy, changeability focuses on the ability for the information security policy to adapt to changes to technology, or to changes in the organizational or external environments.

Changeability is a characteristic that is consistently emphasized in the literature in terms of ensuring that the policy is designed so that it can be changed as policies are meant to be "living documents" (Hayes 2003, Bayuk 1996). Whilst practitioners may not directly use "changeability" as a term, this characteristic is one of the most obvious characteristics that organizations will refer to when assessing the quality of their security policy. For instance, the ability for the policy to be changed in accordance with changes to government legislation, standards or to business processes would be an indication that from the Changeability characteristic perspective that the process, and policy document artifact are of high quality.

3.5.3. Stability

Stability is commonly defined as being the quality of being free from change or variation. In a software quality context stability is defined as "attributes of software that bare on the risk of unexpected effect of modifications" (AS/NZS 4216 standard). Stability, from an information security policy context, would thus be concerned about the introduction of security "holes" as a result of updates to the security policy.

Moreover stability in strategic security policy focuses on the impact of changes to the security policy on the overall security of the organization. For instance, the modification of the security policy within an organization to cater for a new threat may inadvertently cause a problem with the security policy in terms of depth of coverage, or may even introduce a hole in the policy that was previously not exposed.

3.5.4. Testability

In software development testability is defined as the effort required for validating modified software to determine any consequences of those software modifications (ISO9126 2005). For strategic security policy, the ability to physically test changes to policy before implementation would be useful. Unfortunately due to the nature of security policy, it is nearly impossible to accomplish this because, unlike in a software development environment, we do not have a test bed on which we can implement our new policy. In practical terms, however, in some organizations it may be possible to roll out security policy updates and changes incrementally, department by department, or country by country. This may provide some aspects of testability as enjoyed by software quality.

From a meta-policy perspective, it is important that the meta-policy be tested for effectiveness and validity (Baskerville and Siponen 2002). Meta policies could be tested using an incremental implementation approach as described above to give some qualities of testability.

3.6. Portability

Portability, in terms of software quality, focuses on the ability for the software to be transferred from one environment to another. In information security policies the portability of policies (especially between organizations) is not generally considered as being accepted practice as policies should be tailored towards the individual organization. Furthermore, it is impractical for an organization to use other organizations policies as they may not have a direct match, or even provide protection for the organization when they are implemented outside the original development location. So, for normal information security policies within organizations the portability quality factor is not relevant.

However, in large organizations where meta-policies exist, portability becomes an issue. As a result, portability, in large organizations, focuses on the ability of the meta-policies to be utilized throughout the organization. To determine the portability of the meta-policies a number of characteristics of portability can be used. As stated by the ISO9126 standard these characteristics are adaptability, installability, co-existence and replacability. Each of these characteristics will now be described with relation to meta-policies.

3.6.1. Adaptability

In software quality, adaptability focuses on how easy the software can be tailored to different environments without any further development. Adaptability, from a meta-policy perspective, concerns the effort required to use the meta-policy to implement security policy throughout the organization. When a meta-policy is defined well, it tends to be easier to implement and less requests for exceptions from policy will occur.

Even in organizations that do not have security meta-policies the adaptability characteristic may still be useful. For instance, in large decentralized multinational organizations where many business units exist, the organization may allow each business unit to write its own policies. In this case, adaptability would focus on how well the process used by one business unit could be adapted for another to also use. The problem with this scenario is that it is output driven – the creation of the policy artifact. Or, as Maynard and Ruighaver (2003) discuss, it is the lack of documentation that is inherently collected and created as part of many policy development exercises. If documentation concerning decisions made and so forth was present, as in software development, then the policy development processes used would be more likely to be adaptable to other situations. For smaller organizations, or organizations with a centralized structure

this would not be so relevant. It is not expected that these issues will be identified in the research as it is unlikely that organizations have thought about security policy quality in this manner due to the maturity of the field.

3.6.2. Installability

Installability, in terms of software quality, focuses on the effort required to install the software in a specified environment (ISO 9126 standard). For security meta-policy, the focus would be on how easy implementation of the meta-policy is across the organization. This characteristic helps to focus quality on the cultural differences between organizational units based in different countries, not only in terms of national culture, but also organizational culture. In many instances, a meta policy attempts to maximize compliance without outlawing non-compliance given good reasons (Baskerville and Siponen 2002). Thus, if a component of the meta-policy can not be implemented in a country due to cultural or legislative restrictions then that part of the meta-policy could be ignored, or implemented in a different way.

3.6.3. Co-existence

Co-existence, in software quality terms, focuses on whether software adheres to conventions or standards relating to portability within the environment where it is used. For security policy, co-existence has a slightly different bent. When meta-policies are created, it is expected that all units of the organization must be able to adhere to the policy. However, in practical terms it is often impossible for this to actually occur. Often, due to cultural differences, different government legislation or even different technological requirements it is impossible to meet the meta-policy guidelines in full. Co-existence, in real terms then is the ability for the meta-policy to be adapted to the requirements of local situations. In practice, within organizations where full co-existence can not be achieved it is possible to formally request exemption from, or changes to particular areas of the meta-policy to ensure that maximum co-existence is reached, and to document areas and reasons where it is not. This process is often referred to as a "policy deviation" or an "exception" by organizations.

3.6.4. Replaceability

In software quality, the characteristic replaceability is used to investigate the amount of opportunity and effort required for using a piece of software in place of specified other software within that same environment. In other words, this characteristic focuses on the effect of replacing a current software product with a new one. In policy quality, this characteristic may be important when policies are replaced with other policies, for example during a major organizational restructure.

In security policy replaceability is ability for the security policy to be used as a replacement for an existing security policy. For instance, in an organization that is taken over what will be the effects of replacing their organizational security policies with those of the takeover organization? In many cases we expect that organizations do not concern themselves much with how replaceable their policies are, nor whether their policies can replace other policies of organizations that they take over. However, when this does occur we would expect that there would be some kind of compromise between old policies and new policies, with the new policies being modified within the new organization to cater to different organizational norms, or regulatory requirements.

3.7. Compliance

In software quality compliance is defined as whether or not the software adheres to government legislation, or organizationally adopted standards (ISO9126 2005) and is included as a characteristic within each of the six quality factors. We have moved this and made it an overarching factor for our model in security policy due to the importance of compliance to many organizations in terms of security

policy. The reasoning behind this is that when complying with standards and legislation it is likely that different aspects of the compliance requirements will cut across specific quality factors.

In information security policy quality, compliance focuses on whether the information security policy abides by the various standards of the organization whilst also meeting legislative requirements. This is probably the most important area of information security policy quality, and is the one that should obviously be found currently within organizations as organizations must comply with a number of regulations and standards (for example the privacy legislation or Sarbanes-Oxley) and this compliance will be reflected within policy statements. Also, it is likely that this is one of the few areas of security policy quality within organizations that is actively tested for, usually in the form of audits.

4. Discussion

Before further discussing the model a number of issues dealing with the differences between security policy and software development must be addressed. Software development is a very mature field, having been researched extensively over the last 30 plus years. However security policy research, particularly when dealing with quality issues is by comparison only in its infancy. As a result there are clear differences between the fields that may influence the quality model and the manner in which it is utilized. For instance, in software development there is a clear distinction between the development of the software artifact and the development of the requirements for the software. In security policy development this distinction is not nearly as clear, and may not exist at all, which results in the design and development of the security policy being carried out simultaneously.

Furthermore, in software development one would expect that the quality factors are expected to be found in all software quality evaluations. With security policy however, due to its immaturity, the quality factors may not always be found in organizations. The reasoning behind this is that in many organizations, the assessment of security policy quality is often ad hoc and at a level that is appropriate with the maturity of the field. It should be noted though, that looking at security policy quality in this manner will give an in depth understanding of the quality of security policy in organizations, and may also indicate what the quality of security policy could be in organizations as the security policy quality area matures.

As mentioned previously, we are conducting a number of case studies within this research. At present we have concentrated our research on medium size organizations and are attempting to gain access to study a large organization. Whilst it is impossible to fit a complete set of case studies and to present the quality model, we intend to use some insights that we gained from these case studies in this section. From these case studies it is clear that some of the quality factors developed in this paper are not identified. Table 1 shows a summary of the initial results so far, identifying whether factors are identified and by which stakeholders.

Interesting findings from our initial case study results are that none of the quality characteristics for portability, and only one from maintainability and reliability have been identified by stakeholders. There may be a number of reasons for this. Firstly, none of the organizations reported that their policies were of a type where portability would be a factor, in that none of them were large enough to be utilizing meta-policies. In large organizations, where case study research has been limited in this study due to access problems, there is some indication that the portability factor quality characteristics may be of importance to stakeholders. Secondly, the lack of maturity of the security policy lifecycles in the organizations studied may cause maintainability and portability issues to be overlooked as the policies themselves may be thought of as throw away artifacts, with a limited lifespan, or alternatively are used for long periods of time without modification.

For each of the other factors (functionality, usability and efficiency) there is evidence that one or a number of stakeholders are interested in one or more of the quality characteristics. Interestingly the main stakeholders who had the greatest focus on compliance were the security and ICT specialists, with a lesser

focus from executive management, suggesting an internal focus within the ICT areas on ensuring that policies are compliant with legislation and other guidelines whilst having executive oversight. Furthermore they, along with executive management also have a focus on the interoperability of the security policy with other policies in the organization. This is not surprising as ensuring that differing polices are not contradicting each other and can work together is important to the collegiality of the organization.

Table 1: Quality Factors Identified by Stakeholders: Initial Case Study Results

Factor	**Characteristic**	**Currently Identified in Research**	**Interested Stakeholders**
Functionality	Suitability	Yes	Security Specialist, ICT Specialist, Executive Management (this factor was not explicitly mentioned)
	Accuracy	Yes	Security Specialist, ICT Specialist, Executive Management (this factor was not explicitly mentioned)
	Interoperability	Yes	Security Specialist, ICT Specialist, Executive Management
	Security	Yes	ICT Specialist, Executive Management
	Completeness	Yes	Human Resources, Security Specialist, Business Unit Representatives, User Community
Reliability	Maturity	Yes	Security Specialist, Business Unit Representatives, Executive Management
	Fault Tolerance	No	
	Recoverability	No	
Usability	Understandability	Yes	Human Resources, Security Specialist, ICT Specialist, Executive Management, User Community
	Learnability	Yes	Human Resources, Business Unit Representatives, Executive Management
	Operability	Yes	Human Resources, Executive Management, User Community
	Attractiveness	No	
Efficiency	Time Behavior	Yes	Human Resources, Security Specialist, User Community
	Resource Utilization	Yes	Human Resources, Security Specialist
Maintainability	Analyzability	No	
	Changeability	Yes	Human Resources, Security Specialist
	Stability	No	
	Testability	No	
Portability	Adaptability	No	
	Installability	No	
	Co-existence	No	
	Replaceability	No	
Compliance		Yes	Security Specialist, ICT Specialist, Executive Management

Only those stakeholders with governance responsibilities (in these case studies the ICT specialists and executive management) were concerned with the security policy security and this is demonstrated through the rigorous manner in which requests to view the policies are handled within the organization. Unsurprisingly, the most discussed factor of quality was usability.

From the ICT specialist stakeholder's perspective the focus was on the understandability of the policy, whereas for many of the non ICT stakeholders, whilst this was also important, the learnability of the policy was the major focus. However, as we suggested earlier, none of the case study participants thought that attractiveness was an issue.

5. Conclusion

As the maturity of the security policy development lifecycle increases within organizations, the ability for those organizations to assess the quality of security policy becomes important. In this paper we presented a framework for security policy quality. This quality framework is a starting point for organizations to better understand the concepts of security policy quality. Whilst not all of the factors have been identified so far in the case studies there is evidence that many of them can be identified in practice. For those where no evidence has yet been found, there may be good reasons, for instance the immaturity of the field when compared to the reference disciplines.

The model, in its form presented here, does not take into account the differing opinions of stakeholders. For security policies to succeed within an organization it is well known that stakeholders should be involved within the policy lifecycle process. As a result their differing views towards quality, as alluded to in the discussion, are important. Different stakeholders will focus on different aspects of the security policy lifecycle and may in fact have different views about quality.

It should be noted that the influences that each of the quality factors may have on each other may also be important. Whilst this has yet been unable to be reliably measured, it is envisioned that differing factors may be affected by other factors in the model in positive and negative directions. Furthermore, as different stakeholder perceptions are captured by the model, these interactions may become clearer.

From these early case study results it is clear that many of the quality factors utilized within the quality framework may be of interest to organizations when focusing on the improvement of their security policy practices. Whilst gaining access to larger organizations has been problematic the authors hope to gain some insights from these types of organizations regarding some of the quality factors not yet found in medium sized organizations. It is thought that one of the reasons that some of these factors have not been identified may not be a result of the size of the organization, but rather on the maturity of the security policy field when compared to software quality. Issues such as the stability and testability of policies may become more important as the security policy lifecycle matures.

Further research is currently being conducted using this quality factor model. A number of case studies are, and have been conducted. Results from these case studies are currently being utilized in order to improve the model. It is envisaged that the model will be further validated using focus groups to comment on its utility.

6. References

Anton, A. I. and J. B. Earp (2001). Strategies for Developing Policies and Requirements for Secure E-Commerce Systems. Recent Advances in E-Commerce Security and Privacy, Kluwer Academic Publishers**:** 29-46.

Arthur, L. J. (1993). Improving Software Quality: An Insiders Guide to TQM. New York, John Wiley.

Baskerville, R. and M. T. Siponen (2002). "An Information Security Meta-Policy for Emergent Organizations." Journal of Logistics Information Management.

Bayuk, J., L. (1996). Security Through Process Management. Morristown, NJ, Price Waterhouse.

Budgen, D. (1994). Software Design. Wokingham, Addison Wesley.

Chia, P., S. Maynard and A. B. Ruighaver, Eds. (2003). Understanding Organizational Security Culture. Information Systems: The Challenges of Theory and Practice. Las Vegas, USA, Information Institute.

Control Data. (1999). "Why Security Policies Fail." from http://www.securityfocus.com/data/library/Why_Security_Policies_Fail.pdf.
Dunn, R. (1990). "SQA: a management perspective." American Programmer.
Fulford, H. and N. F. Doherty (2003). "The Application of Information Security Policies in Large UK - Based Organizations: an Exploratory Investigation." Information Management and Computer Security **11**(3): 106-114.
Guttman, B. and R. Bagwill (1999). Internet Security Policy: A Technical Guide, NIST.
Hayes, B. (2003). Conducting a Security Audit: An Introductory Overview, Security Focus**:** 5.
Humphrey, W. S. (1989). Managing the Software Process. Reading M.A, Addison-Wesley.
International Standards Office (2005) International Standard 9126.1:2005: Software Engineering - Product Quality, Part 1: Quality Model.
Kabay, M. (1994). "Psychological Factors in the Implementation of Information Security Policy." EDPACS - EDP Audit, Control and Security Newsletter **21**(10): 1.
LeClerc, R. (2001). Audit and Security Control Issues when Conducting Information Security Reviews, SANS Institute**:** 10.
Leinfuss, E. (1996). "Policy over Policing." Infoworld **18**(34): 55.
Maynard, S. and A. B. Ruighaver (2003). Development and Evaluation of Information System Security Policies. Information Systems: The Challenges of Theory and Practice. M. G. a. D. Hunter, K. K. (eds). Las Vegas, USA, Information Institute**:** 366 - 393.
McCall, J. A., P. K. Richards and G. F. Walters (1977). Factors in Software Quality, U.S. Department of Commerce.
McMillan, R. (1998). "Site Security Policy Development." Retrieved 16 Aug 1998, from http://secinf.net/info/policy/AusCERT.html.
Moody, D. L. and G. G. Shanks (1998b). "Improving the Quality of Entity Relationship Models: An Action Research Programme." The Australian Computer Journal **30**(4): 129-138.
Moody, D. L. and G. G. Shanks (2002). "Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework." Journal of Information Systems.
Regan, M. A. (2001). The Computer Security Threat to Small and Medium Sized Businesses - A Manager's Primer, SANS Institute**:** 12.
Saunders, J. and E. Curran (1994). Software Quality. Wokingham, Addison-Wesley.
Standards Australia (1994). AS/NZS 4216 - Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.
State of Oregon. (1998). "Guideline for Developing an Agency Information Systems Security Policy." 20 Feb 1998, from http://www.state.or.us/IRMD/guidelin/secpol.htm.
Tudor, J. K. (2001). Security Policies, Standards, and Procedures. Information Security Architecture: an integrated approach to security in the organization. Florida, USA, CRC Press LLC**:** 79-99.
Tyrrell, S. (2000). "The Many Dimensions of the Software Process." Retrieved 6 April, 2005, from http://www.acm.org/crossroads/xrds6-4/software.html.
Warman, A. R. (1992). "Organizational Computer Security Policies: The Reality." European Journal of Information Systems **1**(5): 305-310.
Warman, A. R. (1995). Developing Policies, Procedures and Information Security Systems. Information Security the Next Decade: Proceedings of the IFIP TC11 eleventh international conference on information security IFIP 95. J. H. P. Eloff and S. H. von Solms**:** 464-476.
Whitman, M. E. and H. J. Mattord (2005). Principles of Information Security. Boston MA, Thompson Course Technology.
Yourdon, E. (1992). Decline and Fall of the American Programmer. Englewood Clifs, NJ, Yourdon Press.