Security Evaluation Expert System

1. Executive Summary

Software is an important resource. It contains and controls data and other resources. Thus measures must be taken to protect that data and resources. Thus Software Metrics are measurement of some properties of a piece of software or its specifications. Metrics are very important in Software Quality Measurement. Since Tom De Marco rightly stated “You can’t control what you can’t measure.” Software control is the part of Software Risk Management Process. Software metrics are calculated on the basis of software risks encountered or which are imminent. Risk-aware organizations may choose proactively to specify, design, implement, operate and maintain their security controls, usually by assessing the risks and implementing a comprehensive security management framework. Many Information security standards promote good security practices and define frameworks or systems to structure the analysis and design for managing information security controls.

For Security control of a system, software is required which can evaluate security metrics and from the potential risks. IT security metrics are based on IT security performance goals and objectives. IT security performance goals state the desired results of a system security program implementation. IT security performance objectives enable accomplishment of goals by identifying practices defined by security policies and procedures that direct consistent implementation of security controls across the organization. Thus system which implement these goals and objectives can it evaluate and calibrate the risks which are imminent to the system. It can compare these risks with the Information security Goals and Objectives and thus determine what measures are to be taken to control those risks or evaluate how good the current system is to control or mitigate or manage risks.
2. Requirements- The Informal Statement.

A system is required to be built which can implement Risk management System. This system should be able to determine various Software metrics on the basis of different Software Risks obtained from the user. Software Metrics are nothing but measurement of these risks in statistical form. The system has some built in standards which are determined on the basis of these software metrics. Thus the security facts of the system, which are maintained, are then evaluated on the basis of these standards by comparing them to the available metrics obtained from the potential risks. Results thus obtained can then be used to take appropriate measures to improve the security control of the system.

Security Evaluation System is a risk assessment and security control and management system. This is a system which can help an organization to evaluate quality of security controls and help them identify appropriate measurements to be taken to improve them. Different inputs are to be provided by the user based on the current scenario of the organization, on the basis of which security metrics are calculated. These metrics are obtained by a basic formula. Formulae for the different metrics have been provided in the NIST document of security metrics guide. The metrics thus obtained are then compared with some standards and evaluated on the basis of the result. They thus determine the performance of the current security controls of the system and understand how effective they are or how they need to be changed or improve to meet the standards.
3. Specifications

The two main processes which are important in the development of IT security program are metrics development and metrics implementation.

- Metrics(types) Selection

  Types of metric selection primarily depend on the level of security control implementation. The focus of security metrics shift as the implementation of security control matures. There are five types of metrics at different levels.

  The types of metrics used in our system are rather of first type since it is still to develop. As the program might evolve and performance data becomes available, it would be able to focus on metrics like program efficiency, timeliness of security service delivery and effectiveness include operational results of security control implementation. And once it is integrated into an organization’s processes, the processes will become self-regenerating, measurement data collection will become fully automated, and the mission or business impact metrics of security-related actions and events can be determined by data correlation analysis.

- Identification\Selection of Metrics

  Generally metrics are determined on the basis and requirement of the people who are suppose to benefit from that particular IT Security program. Different people might be interested in different aspects of the security thus the metrics should be selected accordingly. Selection of metrics should be prioritized so that metrics selected for initial implementation consist of certain requirements. Metrics should facilitate improvement of high-priority security control implementation. It should use data that can be obtained from existing processes and data repositories. It measures processes that already exist and are relatively stable.

  Basic metrics have been implemented in our program since the program is yet to involve, more number of metrics shall be added eventually with the evolution of the program.
• Metric Documentation

Details of different metrics gathered could be stored in some format in some form of documentation.

• Goals and Definitions

NIST publication sp800-56 is used for the primary standards of goals to be met. There are many well-defined standards available for IT Security programs. Any one among these can be used according to the interest of stakeholders.

• Risk Assessment

This process includes Identification of different Risks in the system. Identifying and then also classifying them into a type where it can be associated with a particular software metrics.

• Data Collection – Manual or Automatic

This process includes the method of data collection for the appraisal of software metrics. There are two types of data collection methods Manual and Automatic. The choice of method of data collection primarily depends on the type of data resources and extraction tools available. Automated data extraction requires self-assessment tools, certification and accreditation (C&A) databases, incident reporting and response databases, and other data sources as a security program matures.

But since our security program is in an infant stage we have adopted manual method of data collection. Data is to be inputted by user through a Graphical User Interface. In this graphical interface various questions pertaining to Software Risks are being asked to the user. The data thus obtained can be used for appraisal. Gradually as the system develops automated tools for data collection can be used. When all the data is
gathered by using automated data sources without human involvement or intervention metrics data collection is fully automated.

- **Identify and apply Corrective Actions**

  After metrics development the data obtained can be used for identifying what corrective measures can be taken to improve the system. Corrective measures shall differ according to the system to which it applies and to whom it concerns. Corrective measures also need to be prioritized depending on the goal requirements of risk mitigation. Then these measures can be applied to its appropriate areas according to its requirement.

**Java Expert System Shell**

We have used JESS as Expert System Shell development tool. This has following features:

*Jess* is a rule engine and scripting environment that provide rule-based programming suitable for automating an expert system.

*Jess* is written entirely in Sun's Java language by Ernest Friedman-Hill at Sandia National Laboratories in Livermore, CA.

Its powerful scripting language gives you access to all of Java's APIs

*Jess* uses an enhanced version of Rete algorithm to process rules

*Jess* has many unique features including backwards chaining and working memory queries and can directly manipulate java objects

Rules, facts, templates and functions can be defined in JESS as well as since JESS supports JAVA Classes and their instances can also be defined in it.

It supports modular development of the project.
4. Feasibility Study

Modular Approach:

In the given security evaluation system, different classes are created that represent different areas of risk and security control. The attributes of these classes are then stored in the form of shadow facts about the organization in the Jess database. Jess engine then fires various rules based on these shadow facts.

Structural Approach:

Another approach can be to add facts about the organization by using the Facts class present in the java libraries for Jess. By creating objects of this class and specifying various attributes of the fact we can directly store so created fact into the jess database in the form of ordered or unordered facts. In such case creating respective classes for respective areas of risk and security control is not required. All the facts go into the database as the objects of one class Facts. The jess engine then uses these facts to fires various rules.

Comparison:

<table>
<thead>
<tr>
<th></th>
<th>Modular Approach</th>
<th>Structural Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>use</td>
<td>can be used when facts needs to be arranged in modules based on their underlying properties</td>
<td>can be used when facts don’t have underlying properties to be grouped together in modules</td>
</tr>
<tr>
<td>Effect on the</td>
<td>Could increase the number of classes</td>
<td>No extra classes are required</td>
</tr>
<tr>
<td>implementation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facts-rule</td>
<td>Easy to identify what facts influence what rules because of their underlying properties</td>
<td>Hard to determine what facts influence what rules</td>
</tr>
<tr>
<td>relationship</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. Implementation

Data base is represented by various facts provided by the user by answering various questions asked regarding the current scenario of the organization. The answers given to these questions are then stored in the expert system’s database in the form of shadow facts. Various rules governing these facts are pre-loaded into the jess engine. These rules get fired when specific facts get entered, and thus various metrics get calculated by the expert system which is given back to the user in the form of feedback. In the end the expert system provides number of suggestion as to where the organization have risk and security control issues and what remedial measures to be taken to improve the security issues present in the organization.

For example, if the organization needs to determine whether the risk in various systems are periodically assessed or not, then it is asked to enter various facts about the current risk potentials present in the organization through various questions like

1) How many systems have risk assessment performed and documented?
2) How many systems have gone through risk assessment in various time frames like within 2 yrs, 3 yrs etc.
3) How many systems have not gone through risk assessments?

Answers to all these questions, given by the user are then stored in the data base in the form of facts. Rules governing these facts get fired and the corresponding metric that calculates the percentage of systems that have risk assessment performed documented. This percentage is then used in the determination of how organization fairs in the risk assessment of various systems under the scale of bad, average or good. In the end the expert system provides suggestions as what necessary measures needed to undertake in order to mitigate the risk factor.
Structure and content:
Various critical elements that are of importance to the organization are structured in the given system in the form of various classes. Another single class creates the user interface for user inputs and instantiates objects of these classes that are used to add various facts into the database based on the inputs provided by the users and in the end the same class displays the user the analysis done and necessary measures to be taken.

User Interface:
User interface provided by the given security evaluation expert system consists of various forms which consist of sets of questions focused on particular areas of organizational interest. These forms are displayed one after the other till all the questions are answered. Once the user submit the forms by pressing the submit button on the form, the expert system displays the result of analysis done and the appropriate measures required.

Limitations:
The given expert system is not been completely optimized in terms of time complexity. Also there is a scope of adding many more metrics into the systems.

Hardware Requirement:
operating system: any operating system like windows 98/2000/XP, linux, unix, solaris.

Software Requirement:
- Any version if sun’s Java Development ToolKit (JDK) for compiling java code.
- Any version of Java Expert System Shell (JESS) which provide data store and rule engine.
6. User Guide and GUI

The project files consist of six java files and one .clp file containing templates of classes and rules. The java file UserInterface.java is the principal file.

Steps to run the program

1) Copy all the files in the working directory and have the classpath of JDK set to this working directory.
2) Compile the UserInterface.java file as “javac UserInterface.java”
3) Run the same file as “java UserInterface”
4) You shall see a graphical user interface forms. Input the required information through the GUI.
5) After filling up the required information you shall see the results.
6) And then you might receive suggestions to overcome the problems.

Simple questions will be asked for Risk Assessment like

1. How many systems are there in your agency (or agency component, as Applicable)?

2. How many IT systems have been assessed for risk during the last reporting period?

3. How many risk findings were discovered for all risk assessments conducted in the reporting period?
It is not necessary to have precise field data you can speculate the values for testing purpose.

Program follows very simple GUI as having complex GUI is not part of the expert system and is no way required. Input data is given by the user through form like layouts and results are displayed through equally simple layouts in verbal form.

7. Development process documentation

Member 1: Amit Pillay

- Feasibility Study - He studied the problem and pondered over different possibilities of solutions. He did a comparative study of all the feasible solution and then selected the best approach to solve the problem.

- Programming in JESS and JAVA – He selected JESS as Expert System Shell and did major part of programming in it. He also selected JAVA and fairly contributed programming in it.

- Documentation – He supported the documentation the way he supported the project work i.e. he did the documentation part of feasibility study, Implementation, User guide and GUI description.
Member 2: Bhavik Gandhi

- **Metrics Identification** – He studied the details of the metrics. He also studied the development and implementation process of various metrics. He also took an overview of the various metrics and selected appropriate metrics to be implemented.

- **Design and Programming in JAVA** – He represented the problem in computer domain. He made decision of which classes to be included and also did programming in JAVA.

- **Documentation** – He composed different parts of the project documentation like Executive Summary, Requirements, Specifications and Development process documentation.