Chapter - 2

Literature Review

☑️ Alshammari, Bandar and Fidge, Colin J. and Corney, Diane (2009)

These literatures show that

- Security metric is not considered as much as other quality attributes such as complexity metrics. Also, most security studies concentrate on the level of individual program statements. Such type of approach makes it hard and expensive to discover and fix vulnerabilities caused by design errors in the existing system.

- In future work, let focus on the security design of an existing object oriented application and define security metrics. These metrics allow designers (developer or system analyzer) to find out and fix security vulnerabilities at an early stage of the re-engineering process and it helps to designer review the security metrics to make particular decision about security into re-engineering approach.

☑️ Ashwin B. Tomar and Dr.Vilas. M. Thakare (2011)

This paper provides information about

- Quality of software is very important into the computer science because quality is nothing but future requirements characteristics of end user or customer that is in measurable form. Basically, quality can be those product characteristics which meet end users requirements and thereby, provide product satisfaction. Hence, it is concluded that quality should be contain all properties (or characteristics or features) and vital attributes of a product which should satisfy the given requirements.

- Therefore this paper aims to provide a basis for software quality model research.


This thesis gives the clear idea about

- How external software properties like cohesion and coupling and external properties like performance and security are correlated. It defines different security metrics to measure external property by using internal property like coupling. This thesis also suggests that future design review and testing should focus on coupling properties when security is specified as an important factor in the requirements specification.

- This gives idea about how external properties and internal properties are correlated .And clear direction regarding use of coupling feature for measuring security.

☑️ B. Alshammari, C. J. Fidge, and D. Corney, (2009)

This paper describes

- Different approaches which aim to measure the quality of various object-oriented design attributes such as reusability, flexibility, and functionality based on their relevance to certain quality design properties. But there is little attention on security of
software. Most of the studies focus on security at system level or at the code level. But to measure a security at code level is quite expensive. Paper also provides alternate designs by applying refactoring. It uses UMLsec for representing input artifacts i.e. class diagram.

➢ Idea extracted from this paper for proposed tool is to generate more than one design which gives different level of security based on the requirements. Again different features of OO Design can be used to find out the security from design artifacts.

❖ **Bela Ujhazi, Rudolf Ferenc, Denys Poshyvanyk and Tibor Gyimothy (2009)**

This paper presents

➢ Two new conceptual metrics for measuring coupling and cohesion in software systems. Our first metric is Conceptual Coupling between Object classes (CCBO) which is based on the well-known CBO coupling metric while the other metric is Conceptual Lack of Cohesion on Methods (CLCOM) which is based on the LCOM cohesion metric.

➢ One advantage of the proposed conceptual metrics is that they can be computed in a simpler way as compared to some of the structural metrics.

➢ Empirically study of CCBO and CLCOM for predicting fault-proneness of classes in a large open source system and compared these metrics with a host of existing structural and conceptual metrics for the same task.

❖ **Chowdhury, B. Chan, and M. Zulkernine, (2008).**

This paper have conducted

➢ An exhaustive survey of the commonly reported security metrics. It proposed three metrics: stall ratio, coupling corruption propagation, and critical element ratio. By conducting two case studies it has been shown that how the proposed metrics can be effectively used for different measurements. The interesting observation is the correlations among the metrics demonstrate some consistent patterns. The main aim of this paper is to come up with more code-level metrics as it has been observed that relatively fewer metrics assess security at code level.

This paper helps to understand the effect of coupling and how corruption in coupling can propagate in different modules within software package. Important part that can be used from this paper is security or the metrics are mostly depends on data collection methodology used for that software.

❖ **G.S. Anandha Mala, J. Jayaradika, and G. V. Uma (2006)**

This paper presents
NL-OOML presents an approach to extract the elements of the required system by subjecting its problem statement to object oriented analysis. This approach starts with assigning the parts of speech tags to each word in the given input document. The text thus tagged is restructured into a normalized subject-verb-object form. Further, to resolve the ambiguity posed by the pronouns, the pronoun resolutions are performed before normalizing the text. Finally the elements of the object-oriented system namely the classes, the attributes, methods and relationships between the classes, the use-cases and actors are identified by mapping the ‘parts of speech-tagged’ words of the natural language text onto the Object Oriented Modeling Language elements using mapping rules. But approximately 12.4% of additional classes and 7.4% of additional methods are identified in all the samples taken each of around 500 words.


In this paper the authors state

This is a natural language-based CASE tool called CM-Builder which aims at supporting the Analysis stage of development in an Object-Oriented framework. CM-Builder uses robust Natural Language Processing techniques to analyze software requirements texts written in English and build an integrated discourse model of the processed text, represented in a Semantic Network. This Semantic Network is then used to automatically construct an initial UML Class Model. The initial model can be directly input to a graphical CASE tool for further refinements by a human analyst.

CM-Builder analyzes the requirements text and build initial class diagram only. This model can be visualized in graphical case tool by converting it into standard data interchange format where human analyst can make further refinements to generate final class model. Also CM-builder makes the extensive use of NLP techniques.


The authors proposes

"GOOAL" (Graphic Object Oriented Analysis Laboratory) receives a natural language (NL) description of problem and produces the object models taking decisions sentence by sentence. The user realizes the consequences of the analysis of every sentence in real time. Unique features of this tool are the underlying methodology and the production of dynamic object models. GOOAL produces the class diagram by considering the validation threshold of 50% and its coverage accuracy (Precision matrices) is very minimum that is 78%.

Istehad Chowdhury, Mohammad Zulkernine (2010).

Paper gives brief description
About software metrics which are often used to assess the ability of software to achieve a predefined goal. Software metric is a measure of some property of a piece of software. Complexity, coupling, and cohesion (CCC) related metrics can be measured during the software development phases (such as design or coding) and used to evaluate the quality of software. Because high complexity and coupling and low cohesion make understanding, developing, testing, and maintaining software difficult, they may lead to introduction of vulnerabilities. It provides empirical evidences that complex, coupled and non-cohesive software entities are often less secure by conducting study on history of Mozilla Firefox.

This review helps to provide basis for finding security of software by considering Coupling as one of the major OO design feature. Because coupling is nothing but the connection between different modules in a software package.


Proposed paper provides

Quantitative measurement for the degree of trustworthiness for software systems by defining software security metrics based on vulnerabilities included in the software systems and their impacts on software quality. It use the Common Vulnerabilities and Exposures (CVE), an industry standard for vulnerability and exposure names, and the Common Vulnerability Scoring System (CVSS), a vulnerability scoring system designed to provide an open and standardized method for rating software vulnerabilities. It tries to correlate internal weakness Contribution of this paper in proposed tool is it suggests the need of identifying vulnerabilities and accordingly priorities it based on their severity.

Rudiger Lincke, Jonas Lundberg and Welf Lowe (2008)

This paper discuss that

Software engineering developer and system analyst should be based on the tools for implementing these metrics to support them in quality evaluation and ensure tasks to allow to measure software quality and to deliver the information needed as input for their decision making and engineering processes.

Currently a large body of software metrics tools exists. But these are not the tools which have been used to evaluate the software metrics.

Problems with this technique:

1) It has high coupling values and less cohesion values.
2) Also it ignores security metrics such as Data and Operation Access Metrics.
3) It analysis only few source code language file such as for java.

S. Arun Kumar, T. Arun Kumar and P. Swarnalatha (2010)
The authors introduce

- Various standard definition of quality-1) quality contains all characteristics and significant features of a product or an activity which related to the satisfying of given requirements, 2) quality is nothing but totality of features and characteristics of product or a service that bears on its ability in satisfy the given need, 3) conformation to requirements, 4) meeting user requirements etc.

- The important role of software process improvement is evaluating the current status of the software and decides the improvement proprieties.

Therefore, it should the focus on the software process improvement that requires the need of software measure i.e. measurement of software quality metrics.


In this paper the authors describes

- A methodology and a prototype tool called Linguistic assistant for Domain Analysis (LIDA), which provide linguistic assistance in the model development process. It presents a methodology to conceptual modelling through linguistic analysis. Then gives overview of LIDA’s functionality and present its technical design and the functionality of its components. Finally, it presents an example of how LIDA is used in a conceptual modelling task.

- This tool identifies model elements through assisted text analysis and validates by refining the text descriptions of the developing model. LIDA needs extensive user interaction while generating models because it identifies only a list of candidate nouns, verbs and adjectives, which need to be categorized into classes, attributes or operations based on user’s domain knowledge.

- S. Zhou, H. Yang and P. Luker, William C. Chu

The authors have suggested

- Software metrics are very useful in a forward engineering of re-engineering process of existing software system. Also, they are absolutely necessary in re-engineering process. They show exactness, clear picture and understanding of the existing software system. It should be the first step in transferring effective re-engineering process.

- Let’s use further metrics to choose the most practical and powerful metrics from existing metrics and develop or generating new metrics and then add them to the new complexity and security categories.

This specification defines

- The vocabulary and rules for documenting the semantics of business vocabularies, business facts, and business rules; as well as an XMI schema for the interchange of business vocabularies and business rules among organizations and between software tools.
- In software modelling, SBVR is a modern and an improved way of capturing requirement specifications in natural languages that is not only easy to read for human beings but also simple to machine process. A typical SBVR representation such as the SBVR business rules is simple to machine process due to the higher order logic foundation of SBVR.

Using SBVR, one can generate a shared domain model (based on business vocabularies and rules). Both constituents of a standardized SBVR representation are explained below:

**SBVR Business Vocabulary**

SBVR Business Vocabulary is the collection of business entities, their instances and relationships between them, which can be used by any organization in their writing and talking during the course of their business.

- **Terms**: These are the noun or group of words which can be collectively used for the designation of a business entity. For example: “bank” or “investment bank”
- **Name**: These are the words which are used to represent the instance of a particular term. For Example: “SBI” which is an instance of bank.
- **Fact Type**: These are the sentences which represent the relationship between terms.

We are using the template term-verb-term to establish the relation between two terms, as it is very obvious that a mutual relationship between 3 entities can be easily break down to maximum of 3 binary relations. For example, the fact type “customer owns account is member” states that a customer is related with account and account is related with member and a person who owns an account will be a member. This relationship can be breakdown to two relations as described by the two fact type like “customer owns account” and “customer is member”.

Following five types of SBVR vocabulary are explained below:

- **Object Type**: A general concept that exhibits a set of characteristics to distinguish that object type from all other object types” e.g. library, student, etc.
- **Individual Noun**: A qualified noun that corresponds to only one object e.g. Pune is a famous city in Maharashtra.
Verb Concept: A verb concept specifies the relationships among noun concepts e.g. Vehicle has engine.

Characteristic: An abstraction of a property of an object e.g. name of city is Pune, here name is characteristic.

Fact Type: A fact type can be binary fact type e.g. “student borrows book”. Other possible forms of fact types are associative fact type, partitive fact types, categorization fact types, etc.

**SBVR Business Rules**

These are the sentences under business jurisdiction which guide the structure and behaviour of an organization. The rules guiding the structure are known as Structural Rules and the rules guiding behaviour are known as Operative Rules.

In SBVR 1.0, the formal representation of a business entities structure or behavior under a business jurisdiction is called a SBVR business rule. A business rule typically expresses structure or operation of a particular business entity in a specified business domain. Each SBVR business rule is based on at least one fact type.

The SBVR rules can be of two types:

- **SBVR Structural Rule**: Such rules are used to define an organizations setup
- **SBVR Behavioral Rule**: Such rules are employed to express the conduct of a business entity

**Timothy M. Meyers and David Binkley (2007)**

- Software reengineering is an expensive process due to the ambiguity of where to focus reengineering effort. Coupling and Cohesion metrics are complexity metrics out of particularly cohesion metrics have the potential to help in this identification and to measure progress. The most extensive work on such metrics is with cohesion metrics. It should use of dependence information that make them an excellent choice for cohesion measurement.

- In future study, it should be raise the most important question such as does a software developer or analyst which could be access to complexity metric values for the program do a better job of restructuring the program?