CHAPTER 3

RELATED WORK

This chapter presents a classification of software metrics and summarizes the work that software companies and organizations have performed in this context.

3.1 CLASSIFICATION OF SOFTWARE METRICS

Software metrics are standards to determine the size of an attribute of a software product and a way to evaluate it. They can also be applied to the software process, Awais Rashid et al (2002). Several books present different classification of software metrics, most of them agree on the following:

a) **Software product metrics**: These metrics measure the software product at any stage of its development. They are often classified according to size, complexity, quality and data dependency.

b) **Software process metrics**: These metrics measure the process in regard to the time that the project will take, cost, and methodology followed and how the experience of the team members can affect these values. They can be classified as empirical, statistical, theory based and composite models.

This research project focuses on presenting software metrics as they could be applied for estimating the effort from the Unified Modeling Language (UML), Junichi et al (1999) and also a new estimating strategy by integrating the function point estimation with quality standards was proposed.
Comparison analyses have been generated by comparing the result with the already existing strategies with the developed new technique, Brian Nixon, (1992). The UML is an industry – standard language that allows software organizations to clearly communicate requirements, architectures and designs.

3.2 EXAMPLES OF SOFTWARE METRICS SYSTEMS

Brian Nixon, (1998) have dealt with the idea of concentrating software metrics that could be used during the software process, even though not all software metrics have been organized in the same manner. This section presents some examples of software metrics sets that present and implement different software metrics. Table 3.1 depicts the related work carried out in software metrics in an organization.

3.2.1 Software Measurement Laboratory

There is a very extensive and comprehensive presentation of software metrics and tools at http://irb.cs.uni-magdeburg.de/sw-eng/us/index.shtml posted by the Software Measurement Laboratory (SMlab) at the University of Magdeburg, Germany. The SMlab’s teams, members led by Prof. Reiner R. Dumke, have done a very good job concentrating different useful community activities related to software metrics. They present a wide range of tools and topics related to software metrics and have created a large community of participants that comprises members all around the world.

3.2.2 ZD – MIS

ZD-MIS stands for Zuse / Drabe Measure-Information-System, (http://home.tonline.de/home/horst.zuse/zdmis.html) which provides a 'comprehensive software test framework'. The project was initiated by Horst Zuse and Karin Drave. It comprises a large set of software metrics and a book
with the fundamentals. This project is currently offered as a product that can be purchased.

Table 3.1 Software Metrics Related Work

<table>
<thead>
<tr>
<th>Sl.No.</th>
<th>Company/Organization</th>
<th>Brief Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>SMLab</td>
<td>Software Measurement Laboratory at the University of Magdeburg, Germany.</td>
</tr>
<tr>
<td>2.</td>
<td>ZD-MIS</td>
<td>Zuse / Drabe Measure – Information System private company.</td>
</tr>
<tr>
<td>3.</td>
<td>Power Software</td>
<td>Private company.</td>
</tr>
<tr>
<td>5.</td>
<td>QSM</td>
<td>Quantitative Software Management is a private company.</td>
</tr>
<tr>
<td>6.</td>
<td>CMM</td>
<td>Capability Maturity Model of Software of Software Engineering Institute at Carnegie Mellon University.</td>
</tr>
<tr>
<td>8.</td>
<td>Total Metrics</td>
<td>Consulting services</td>
</tr>
</tbody>
</table>

3.2.3 Power Software

Power Software is a company (http://www.powersoftware.com) that provides different tools of software metrics. The tools that Power Software provides go from counting lines of code to Cyclomatic Complexity, Halstead product metrics, and Object Oriented metrics, Junichi et al (1999). The
company also provides tools to measure the effort and project management metrics.

### 3.2.4 Charismatek Software Metrics

Charismatek is a company (http://www.charismatek.com.au/) that provides Metrics Software Tools and consulting services. They have developed a Function Point Tool: ‘WORKBENCH™’ which has been receiving good ratings by a user satisfaction survey, Chung (1993). This company also has other software programs to aid in the software management process.

### 3.2.5 QSM

Quantitative Software Management (QSM) can be found at http://www.qsm.com/. QSM also specializes on developing software metric tools for project management and they have developed SLIM-Metrics. SLIM-Data Manager software tools that graphically allow users to see resources spent and estimation of quality for the project, Galin (2001). Both have a database system integrated to track changes and see the history of the project across the time.

### 3.2.6 Quality Models

There are many Quality Models provided by different organizations that give guidelines of software product/process improvement, Chung (1993). One notable work is the Capability Maturity Model of Software (CMM) of the Software Engineering Institute (SEI) at Carnegie Mellon University), Chung et al (2000). The CMM suggests a software company evolution improvement that goes from an Initial Level of software development process in which there is no organization, to an Optimizing Level, in which there is a
continuously improving process. In the Optimizing level, the software development process and products are constantly monitored and the results are predictable. Another important work is the International Organization for Standardization, which has a standard for quality management systems (http://www.iso.ch/iso/en/iso9000-14000/), Roetzheim et al (1998). Many companies follow different standards to achieve certifications.

3.3 SUMMARY

This chapter has presented a classification of software metrics. Software Metrics gives a precise knowledge of the status of an attribute of the software and helps to evaluate this software attribute in an objective way. Software Metrics also helps to later make plans for modifications that need to be implemented in the future. In addition, it saves the values obtained as history for further reference. What motivates this study of Software Metrics is to develop a software metrics framework that has a set of metrics that could be used as a standalone tool for estimation. The next chapter deals with the importance of software engineering with various other disciplines.