Chapter 9

Conclusion

9.1 Research Contributions

In this thesis, we have argued for the importance of software metrics to enable quantitative management decision-making. Our research effort has focused on the identification of a core set of 14 test and evaluation metrics, which should be collected throughout the software project life cycle, and stored in a database. In addition to the collection and use of software metrics data in its raw form for simple management queries, we have also argued for the analysis of raw metrics data to produce useful information. This will enable management to obtain more information to control the current projects, as well as to identify trends and patterns, which would enable process and product improvements in the future. We have also identified software metrics analysis techniques that can be used for risk and quality management, each with their advantages and drawbacks. The results of the analysis should also be stored in the metrics database in order to aid future management decision-making.

Perhaps the most overlooked area in metrics database research is the analysis of the efficiency of various data models based on metrics queries, which are focused on risk and quality management. Both practitioners and researchers often take a certain model (say, the relational model) for granted and then try to phrase their queries to suit the peculiarities of the model. More often than not, they try to limit their queries to what is supported by the data model. Therefore, since the relational model in its pure form handles recursion poorly, both practitioners and researchers attempt not to pose any queries that require extensive temporal processing and recursion on the relational data model. Rather than make the query fit the model, we feel that the metrics database research should take the opposite approach of making the model fit the query.

Following are the major contributions that have resulted throughout the course of research discussed in this thesis.

- We have proposed a set of 14 core metrics that can most effectively be used across various

organizations for risk and quality management. These metrics have been referred to as the Test and Evaluation metrics. The set includes 8 critical metrics of NASA's metric data set, and has been selected based on the principal component analysis.

For risk and quality management we have presented an analytical framework using influence diagrams, accompanied by various analytical and classification techniques. The crux of this framework is to carry out diagnosis and probing in case some risks are identified in the early part of the software development process life-cycle. We have also presented how prediction can help to identify resources over a period of time in order to achieve some desired quality software product. In summary, we have shown that many existing data analysis techniques can be effectively used for diagnostic and prognostic probing with the Test and Evaluation metrics.

For modeling a software development environment, we have proposed two mathematical models for analyzing the effects of productivity/learning capabilities of software development team and the CASE tools on the development of software development process. These models are based on a more realistic set of assumptions than the ones used in previous models, such as Putnam/Norden model. Furthermore, we have extended these models by using a cost model proposed by Dr. Mizuno (NEC) that considers various parameters including the size of software being developed. A detailed validation of these models reveal that they are superior over the existing models, such as the Putnam/Norden model.

Another unique and original contribution of our research is the development of a Petri-net based formal framework for temporal modeling of metrics data. For this framework we have proposed a set of n-ary temporal relations which can be effectively used to develop various abstractions of quality/risk using an object-oriented model. The reason for proposing a Petri-net model is that considerable semantic heterogeneity may exist among the users of the metrics data. Hence, various analysis techniques need to be invoked to support such abstractions. As mentioned earlier, we believe that the use of the correct data model has a direct impact on the efficiency of complex metrics queries and the maintainability of metrics databases. For the purpose of implementation of this object-oriented model, we have provided a comparative

assessment between the relational model and the graph data model (GDM). Our benchmarks have shown that GDM is a more favorable candidate for implementation than the relational model.

The final contribution of this thesis is a unique methodology based on the theory of recursive graphs (R-graphs) that can be used to model different levels of abstractions for risk and quality management of software project in a systematic manner. In particular, the proposed methodology allows transformation of views to object-oriented paradigm. Based on the research presented in this thesis, we have proposed a pragmatic architecture for software metrics database system.

9.2 Future Extensions

The collection, analysis, and storage of metrics data in a metrics database is only the first step to make the use of software metrics widely accepted by management and other users. Automated tools should be developed for the collection of metrics data in order to reduce the overhead of metrics data collection. Furthermore, since metrics data is only useful if it is analyzed effectively and provide timely tools for management, we also need to look at methods to promote ease of use, as well as tools to analyze metrics data.

As data is being increasingly realized as a valuable strategic resource, organizations are using ever more sophisticated data collection, storage, and analysis techniques. This is evident from the almost exponential growth in the adaptation of data warehousing, on-line analytical processing (OLAP), and data mining technologies in the corporate environment. A number of commercial tools have become available for building and storing warehouses, and for performing analyses on them. As observed in this thesis, data about the process and product, useful for project management, has a very strong temporal component. Further, to monitor, analyze, and subsequently improve the process, it is important to capture this temporal nature explicitly. A natural approach to do this is to capture such information in a data warehouse. The main database

used for project management can be seen as the operational (transactional) data store which supports the project, while data is continuously extracted from it and fed into a data warehouse. The operational data store captures the most recent state of the project, and the warehouse records the entire history of states that the operational data store has progressed through. On-line analytical processing (OLAP) is a powerful set of techniques for analyzing data along various dimensions, and can be used for the analysis of this data. Data mining is the approach of automatically extracting heretofore unknown, yet significant, patterns in data. These can also be used to analyze the project management data warehouse. Overall, it is proposed that the highly successful techniques of data warehousing, OLAP, and data mining be used to store and analyze project management data.

Recent research in data mining techniques can have high-level impact on managing software product development. These techniques have the potential of extracting useful patterns and behaviors from software metrics data and generating information on risk and quality management. Such techniques will enable the generation of the most useful data and knowledge representations for such management. These techniques when applied to the proposed metrics can allow the measurement of such abstract criteria as relevance, value, and potential utility of information. Efficiency and scalability issues in data mining and gathering statistics over massive data sets is a promising research avenue that can be explored based on the software metrics proposed in this thesis. In particular, future research can be focussed on seeking solutions for efficient and scalable classification, clustering, probabilistic modeling, prediction and estimation, and dependency analysis. All these factors will help greatly in performing optimized searches of large data stores using automatic discovery agents. Future research can also be undertaken for developing tools and techniques which may also be employed in multi-dimensional probing and exploration of software metrics data, including information pre-processing, efficient sampling, and search domain reduction methods.

Other issues for future research can focus on rational software development decisions and develop a framework for completely automated software systems. Rational decisions on software

development must take into consideration the economic consequences of the development in time. Recent progress in valuation methods shows that not only the value of a single software development project can be determined, but the price of a commercial software package can also be determined by developing sophisticated quantitative supply and demand models.