Chapter 1.18
Intelligent Analysis of Software Maintenance Data

Marek Reformat
University of Alberta, Canada

Petr Musilek
University of Alberta, Canada

Efe Igbide
University of Alberta, Canada

ABSTRACT

Amount of software engineering data gathered by software companies amplifies importance of tools and techniques dedicated to processing and analysis of data. More and more methods are being developed to extract knowledge from data and build data models. In such cases, selection of the most suitable data processing methods and quality of extracted knowledge is of great importance. Software maintenance is one of the most time and effort-consuming tasks among all phases of a software life cycle. Maintenance managers and personnel look for methods and tools supporting analysis of software maintenance data in order to gain knowledge needed to prepare better plans and schedules of software maintenance activities. Software engineering data models should provide quantitative as well as qualitative outputs. It is desirable to build these models based on a well-delineated logic structure. Such models would enhance maintainers' understanding of factors which influence maintenance efforts. This chapter focuses on defect-related activities that are the core of corrective maintenance. Two aspects of these activities are considered: a number of software components that have to be examined during a defect removing process, and time needed to remove a single defect. Analysis of the available datasets leads to development of data models, extraction of IF-THEN rules from these models, and construction of ensemble-based prediction systems that are built based on these data models. The data models are developed using well-known tools such as See5/C5.0 and 4cRuleBuilder, and a new multi-level evolutionary-based algorithm.

Single data models are put together into ensemble prediction systems that use elements of evidence theory for the purpose of inference about a degree of belief in the final prediction.

INTRODUCTION

Many organizations want to prepare reliable schedules of maintenance tasks. Such schedules would lead to on-time realization of these tasks and better management of resources. This is an important issue, especially in the case where maintenance tasks account for more than half of a typical software budget (Glass, 1989; Smith, 1999). Because of that, the software industry is exhibiting an increased interest in improving software maintenance processes. Software engineers use a number of different tools to support maintenance activities and make them more efficient. The most commonly used tools are tools for model-based software component analysis, metrics extraction, measurements presentation, and statistical analysis and evaluation. Besides that, software maintainers need tools that would help them to understand relationships between attributes of software components and maintenance tasks. Knowledge gained in this way would increase understanding of influence of software component attributes, such as size of code, complexity, functionality, and so forth, on efforts associated with realization of maintenance tasks.

There are four different categories of software maintenance: *corrective*—it involves changing software to remove defects; *adaptive*—it leads to changing software due to changes in software operating environment; *perfective*—it embraces activates that lead to improvement of maintainability, performance, or other software quality attributes; and *preventive*—it is defined as maintenance performed for the purpose of preventing problems before they happen. The corrective software maintenance is associated with activities related to elimination of software defects. This process is a key factor in ensuring timely releases of software and its updates, and high quality of software. Different tools and systems are used to support activities that are directly related to correction of defects. However, there is also a need to build systems that support decision-making tasks and lead to preparation of schedules and plans for defect removal processes. These systems should not only provide quantitative predictions but also give indications about plausibility of these predictions. Additionally, they should provide maintenance engineers with knowledge about defect removal efforts that explain obtained predictions. In summary, it is desirable to have a tool equipped with the ability to retrieve knowledge about relationships between attributes describing software and factors that directly or indirectly influence defect elimination activities.

Some of the important questions asked by managers and software maintenance engineers regarding removal of defects from software systems are:

- Does a defect removal process depend on functionality of software components?
- Does a defect removal process depend on the time when a defect entered the system?
- What are the factors that influence time needed to correct a single defect?
- What kind of relations between software component attributes and time needed to remove a defect can be found from software maintenance data?
- How confident can someone be about dependencies that have been found between a defect removal process and attributes of software components?

This chapter focuses on building software maintenance data models and their analysis. The aim is to build a prediction system that is able to provide software maintenance engineers with predictions regarding defect elimination efforts, knowledge about factors that influence these