

Summary of the thesis presentation of Mr. Arash Zaryabi Langaroudi on:
“Resource Allocation and Optimal Release Time in Software System”

Like every project, software production project management involves quality, cost and timing control. For improving the effective management of these tasks, Mr.Zaryabi has worked and proposed new methods to control these factors, he has presented statistical tools to predict the cumulative number of software defects, optimized resource allocation in software production life cycle, and optimal release time and maintenance policy that can have a huge positive impact on making the software quality assurance easier. Also he has demonstrated the performance of the proposed algorithms on a variety of software defect datasets, compared and proposed techniques with the existing methods. Here some of the topics of his presentation are mentioned:

Software Quality:

Software quality is measured by the number of defects in software systems. As the software become bigger, manual inspection of software becomes more expensive.

Software Reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment. Although Software Reliability is defined as a probabilistic function, and comes with the notion of time, but it is different from traditional Hardware Reliability, Software Reliability is not a direct function of time. Electronic and mechanical parts may become “old” and wear out with time and usage, but software will not rust or wear-out during its life cycle. Software will not change over time unless intentionally changed or upgraded. Defect prediction is importance to project managers in allocating the limited resources effectively which provide the accurate estimation of project costs and schedules.

There are several methods for predicting the reliability of seawares based on Bayesian statistics and neural networks reliability growth. Knowledge about the number of expected defects in a software product at any stage provide essential information for decision making in many software development activities, such as cost analysis, resource allocation in testing and release decision time. The aim of software reliability growth modeling (SRGM) is to explain the behavior of software testing process caused by faults.

Mr. Zaryabi proposed statistical fault prediction modeling in his theses. He introduced a software defect prediction model based on the concept of operating characteristic curve and Laplace trend statistic. The idea is to use operating characteristic curves in statistical quality control to construct an efficient, fast, and accurate prediction method to estimate the cumulative number of software defects during the software development process.

He tested his proposed method on real software datasets that were taken from SAP development systems. These datasets contains monthly software defects that were recorded for a period of 60

months and found the probability of the failure. (He presented his results in several diagrams and compared them with the results based on other methods)

Resource allocation:

Most software quality measurement techniques are based on counting defects found in software system, resulting in their impracticability to estimate the human-resource cost of maintenance and predict the reliability of a future product. Mr. Zaryabi proposed a queuing theory based model for resource allocation in software development. The main objective is to model software management and maintenance during the system test, alpha test, and the beta test phases of a software system. He proposed model answers managerial questions related to project status and scheduling, and also provides a quantitative measure of the software.

It is well-known that more pre-release development and testing on systems can reduce future development costs and result in higher software quality.

In order to optimal software release and maintenance policy, a proper software resource allocation must be effected. For this manner a software defect prediction model using operating characteristic curves is presented. It is a sort of predictor to use geometric insight in helping construct an efficient prediction method to reliably predict the cumulative number of defects during the software development process. The widely used method for this manner is concept of queue models in communication systems and information processing systems, which is a resource allocation model who answers managerial questions related to project status and scheduling, is then introduced. Using the proposed allocation model, managers will be more certain about making resource allocation decisions as well as measuring the system reliability and the quality of service provided to customers in terms of the expected response time.

To test the performance of the proposed resource allocation model, He used a real software defect dataset that was taken from a SAP development system. This data contains monthly software defects that were recorded in the period of 60 months. The defects and fixes per month are chosen due to the fact that the variances of them in a day are much bigger than the variances in a month. He showed the distribution of Number of defects per week / Number of faults per month and realized that the number of defects is volatile. Also it will not be possible to fix more defects per period if we take into consideration the engineer capabilities of the developers.

He proposed resource allocation methodology helps estimate the need for new developers and resources for future projects. Applying queuing theory to model software management and maintenance helps verify the progress of the testing phase and estimate its cost. Also, decisions can be made about changes in the employees early rather than letting the product miss the schedule deadline. The selection of a queuing model is organization-based.

Cost control:

In any SRGM, the extracted information from the analysis could help management make decisions regarding a software development project. The focus is on decisions of when to release the software and when to stop maintaining the software after release. The cost of developing the software, as the most important aspect in software business, is used to make such decisions.

The total expected software cost, which can occur in both testing and operational phases, is an optimization problem. In the operational phase, it is considered two cost factors, namely the maintenance cost due to the software failure and the operational cost to keep the maintenance team. It should be noted that the operational environment after the release may differ from the debugging environment in the testing phase

Stochastic model is proposed to describe the cost behavior of the operation and estimate the optimal time by minimizing a cost function via artificial neural networks. A detailed analysis of software release time and maintenance decision is also presented.

The performance of the proposed approaches is validated on real data from actual SAP projects, and the experimental results, which is done by using a real software defect dataset from SAP, demonstrate a compelling motivation for improved software quality.

Conclusion:

A defect prediction model with source identification can give important ideas regarding the erroneous bottlenecks in the software development cycle. A defect prediction solution provides a guideline to the sources of defects that might be caused due to programmer's inability, failure in requirements collection or design mistakes. Efficiency focused software development units can benefit using defect cause information. They can take necessary precautions in a proactive manner. A defect focused prediction solution can also help to change the development methods. Such a solution or systematic approach can affect in a positive manner to produce less defected software. An important aspect of a defect prediction solution is that such a solution becomes necessary when there is a trade-off between to deliver earlier and to deliver with fewer defects. In today's software development industry, all companies and software development houses are in a severe competition that minimizing development time decreases the overall project cost. Less development and testing time also increases the defect density ratio in the final product. So a quantitative indicator is required for the software company to find the correct point in this balance. A defect prediction solution could provide the required quantitative metric to make a decision on the product delivery.