

Toward maintainance of a commencial software: Defects model

HOUKPE TODJI Mahugnon Honore^{*†}, Nicolas Anquetil[‡]

^{*}University of Lille, France

[†]Inria Lille - Nord Europe, France

homahugnon@gmail.com, nicolas.anquetil@inria.fr

Abstract—Legacy systems are old software that style does useful tasks. In industrial software companies, legacy systems are often crucial for the company business model and represent a longterm business investment. Legacy systems are known to be hard to maintain. This is the case in a french company whose main product is twenty years old software written in PowerBuilder (we call it SPB). Our longterm goal is to help reengineer. But how to validate our intervention? Using moving average, and regression, we evaluate the maintenance state of SPB and produce a dashboard to monitor our future actions. We validate our results with the developer team. We In this paper, we present a lightweight defects model to help planning commercial software maintenance

Index Terms—Legacy system, Defect model .

I. INTRODUCTION

Software companies usually invest time and energy to improve the quality of the software they develop to respond to the raising market demands. Features are developed in hurry sometimes. As consequence they less allow resources for software refactoring to remove source code defects. As the software is out of control and it become hard to improve or maintain. Rewriting these software require time and a lot of resources. At this point one of the solution is to reverse-engineer them.

The main business product is SPB of our company . Our longterm goal is to reverse-engineer SPB. With over 3,000,000 lines of codes, continually updated since more than 20 years, SPB is not versioned and not unit tested. Bugs are registered in a database without correlation with which part of the system is responsible for the bug. Old versions of SPB are lost until 2012. The original developers of SPB are part of the developers team. So the current developers only know SPB partially. So we could not completely rely on developers' view of the system. In addition there is also misunderstanding between developer team and business team. As business team doesn't know the state of SPB. In this condition it is completely impossible to successfully reengineer SPB without 15 useful data models and our analytics.

Our data model and analytics have two main goals. First it provides a report on the state of SPB to the whole company and shows the business team the need to reengineer SPB. Secondly it will help us to monitor our future task on SPB. In this paper we will present the data model and the analytics we made.

This paper is structured as follows:

II. RELATED WORK

III. RESEARCH QUESTION

IV. BACKGROUND

A. Presentation of a PowerBuilder project

PowerBuilder is an enterprise development tool that allows you to build many types of applications and components. A PowerBuilder application component is grouped by libraries. A library can contain different types of Objects: Datawindow, User object, Global function, etc.

B. Ticket

A ticket is related to a task to do. This task can be fixing a bug, writing documentation, adding a new feature, etc. A ticket has the following characteristics:

- the libraries it is related to
- the creation date
- the closing date
- time spent by a developer
 - time to analyze
 - time to implement solution
 - time to test

V. METHODOLOGY

In our company, any task is registered as Ticket. In this section we will present details about the ticket database and our analytics.

A. Dataset

VI. RESULTS AND DISCUSSION

VII. SUMMARY AND CONCLUSIONS