Using Time Series Models for Defect Prediction

in Software Release Planning

A Thesis

Presented to

The Graduate Faculty

Central Washington University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Computational Science

by

James W. Tunnell

June 2015

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hereby approve the thesis of

James W. Tunnell

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

Dr. John K. Anvik, Committee Chair

Dr. Yvonne Chueh

Dr. Kathryn Temple

Dean of Graduate Studies

ABSTRACT

Using Time Series Models for Defect Prediction

in Software Release Planning

by

James W. Tunnell

June 2015

To produce a high-quality software release, sufficient time should be allowed for testing and fixing defects. Otherwise, there is a risk of slip in the development schedule and/or software quality. A time series model is used to predict the number of bugs created during development. The model depends on the previous numbers of bugs created. The model also depends, in an exogenous manner, on the previous numbers of new features resolved and improvements resolved. This model structure would allow hypothetical release plans to be compared by assessing their predicted impact on testing and defect-fixing time. The VARX time series model was selected as a reasonable approach. The accuracy of the model varies for different sampling periods, window sizes, and degree of differencing.

ACKNOWLEDGEMENTS

The author is grateful to Dr. John Anvik, for his advice and patience, to Dr. Yvonne Chueh for her help with exploratory data analysis, and to Dr. Kathryn Temple for her guidance with time series modeling.

TABLE OF CONTENTS

Chapter Page

I INTRODUCTION 1

II LITERATURE REVIEW 3

III MOTIVATION 6

The Next Release Problem 7

The Gap between Abstraction and Reality 9

IV BACKGROUND 12

Time Series Models 12

Trends and Stationarity 13

V METHODS 16

Data Methods 16

Modeling Methods 21

VI RESULTS 26

Data Results 26

Modeling Results 28

VII DISCUSSION 43

VIII THREATS TO VALIDITY 45

IX FUTURE WORK 46

X CONCLUSION 49

REFERENCES 50

APPENDIXES 49

Appendix A—Time Series Data Plots 52

Appendix B—Stationarity Testing Results 58

Appendix C—Exploratory Modeling Results 65

LIST OF TABLES[[1]](#footnote-1)

Table Page

1 Results of sampling example issue data 19

2 Date ranges of data collected, and the number issues that resulted. 26

3 Sliding windows sizes to be used for each sampling period 27

4 Parameter values selected for final modeling. 32

5 A comparison of the final modeling results across datasets 42

LIST OF FIGURES[[2]](#footnote-2)

Figure Page

1 An explanatory model 7

2 Applying defect prediction in the Next Release Problem 10

3 An overview of data methods 16

4 Sampling issue data 19

5 An illustration of time-windowing 21

6 Plot of the none-valid proportion 30

7 Plot of the 95% in-interval proportion 31

8 Distributions for actual and predicted number of bugs,

MongoDB *core server* dataset 34

9 Histogram of prediction errors,

MongoDB *core server* dataset 34

10 Q-Q plot of forecast mean errors,

MongoDB *core server* dataset 35

11 Distributions for actual and predicted number of bugs,

Hibernate *orm* dataset 36

12 Histogram of prediction errors,

Hibernate *orm* dataset 37

13 Q-Q plot of forecast mean errors,

Hibernate *orm* dataset 37

14 Distributions for actual and predicted number of bugs,

NetBeans *platform* dataset 38

15 Histogram of prediction errors,

NetBeans *platform* dataset 39

16 Q-Q plot of forecast mean errors,

NetBeans *platform* dataset 39

17 Distributions for actual and predicted number of bugs,

NetBeans *java* dataset 41

18 Histogram of prediction errors,

NetBeans *java* dataset 41

19 Q-Q plot of forecast mean errors,

NetBeans *java* dataset 42

20 Undifferenced time series data from the Hibernate *orm* dataset 47

1. The tables shown in the Appendixes are not listed here. [↑](#footnote-ref-1)
2. The figures shown in the Appendixes are not listed here. [↑](#footnote-ref-2)