# Reallocation of Resources during Releases for Better Outcome in Software Development

£Md. Tajmilur Rahman

£Department of Engineering and Computer Science, Concordia University, Montreal Quebec, Canada

# Abstract

...

# I. Introduction

Software projects are notorious for going over budget and schedule. Rush periods are often gets seen before a major release that turns the developers into dinosaurs as Frederick Brooks likens in his benchmark study “*The Mythical Man Month*” [1]. This “Rush To Release (RTR)” can be prompted either by external forces such as decision by management to include new features in the release or to release earlier to beat a competitor. Alternatively, the rush may simply be due to inappropriate or unrealistic scheduling. Whatever the reason is it is an obvious. Regardless of the causes, the rush to release stresses developers and often requires developers to work on unusual, high priority or critical areas of the system. In this paper we study how RTR effects project organization and introduces technical debt. Our main research questions are as follows:

1) Do developer work on different areas of the system around the time of release?

2) Are there certain areas of the system that receive increased attention (i. e. do developers focus on a smaller set of files around releases).

3) Do the areas of code that are modified around the time of release have higher defect densities than code that is modified during normal development?

Very few research works have been performed regarding the re-allocation of resources. Robert van Engelen worked for similar kind of a research to understand the resource allocation dynamics across the software projects [2]. He mainly tried to reallocate development resources amongst projects for increasing the satisfactory level of consumer or customer while we are focusing on the impact on code-base like the complexity of script files. Robert proposed a project entropy metric in his work to understand if there is any limit for a particular reallocation does not lead to user satisfaction. Here entropy is to represent disorder and chaos to understand degradation of software and its inherent complexity. In his work resources may not just be the developers but also can be any other resources necessary for a software project development.

This paper is organized as follows. In Section II, we describe some background and motivation. Section III will describe about the ownership of files and ownership of a set of files or a directory. We will try to understand how native a code-base is to a developer or a development team. In Section IV, some analysis to determine reallocation has been performed in a release or where reallocation needs to be performed will be presented. What changes in nativeness ∆ƞ occurs after the reallocation. Section V will give us the result to show how change in ∆ƞ puts impact on the outcome of a software. Finally section VI will give us an idea of our future work and followed by the section VII Conclusion.

# II. Background and Motivation

…

# III. Ownership and Native-Code

…

# IV. Research and Analysis

…

# V. Results and Discussion

…

# VI. Future Work

…

# VII. Conclusion

…

# References

1. F.B. Brooks, J. The MythicalMan Month, Addison-Wesley, Reading, Mass., 1975
2. Robert van Engelen, Subhajit Datta “*Project-entropy: A Metric to Understand Resource Allocation Dynamics across Software Projects*”