

Software Dependencies

Rushiraj Nenuji¹ Harshit Shihora² Lay Shah³

^{1, 2, 3} Computer Science and Information Technology Department
^{1, 2, 3} Nirma University

Abstract--- Software developers many a times use different tools, so that we could find and come to know about dependencies of any particular module/system on the repository before we make changes to it. This is done in order to avoid adverse effects because of changes and hence to avoid failures related to it.

Keywords: Software Dependencies, Systems, Dependency errors, Software failures.

I. INTRODUCTION

As we work with geographically distributed software, this approach creates problems. When we work with modules and there is dependency between them we face problems like communication problem, work coordination, etc. The new study shows that when the teams work differently they can produce better results and so in order to work distributed manner they can produce better results but this creates dependency. It is found by researchers that while using the modularization technique i.e. modular design in software engineering as a traditional technique, many a times dependencies between different modules are their. But, these argument suggests that if we reduce the technical dependencies, it will in-turn make the develops team to work on the distributed modules independently. In short, these paper discusses more about the task dependencies rather than technical dependencies. These should be viewed because in future it can arise problems regarding the quality of the software and might not produce the desired results.

II. THE ACTUAL PROBLEM

For a large-scale software development project the tasks are divided among different workgroups .These workgroups individually work on particular subsystem and then integrate them in order to form a large system. In short workgroups are organized hierarchically and build components of subsystems.

Above these there is a graph which shows the inter-dependencies of different components in a system. If the number of dependencies is more then it's a great load on developers to maintain synchronization among different components. Software dependency is basically comprises of two kind of dependencies. First, Data Dependency, i.e. program A depends on the data of program B. Second, Call dependency, i.e. a component calls a function of another component.

If some changes are made in component A on which component B depends then there is a need for component B to undergo the changes in order to keep sync with A. But, these changes because of dependency cause errors in components which in-turn causes reduction in reliability of the software.

III. HOW TO IDENTIFY WORK DEPENDENCY.

Various standard operational routines or procedures have very limited applicability in such dynamic operations. For the field of Software Engineering the dependency related to technical codes, i.e. the code written by the user this problem should be addressed with more importance. If this problem is ignored then we have problem regarding more and more defects and this causes a big problem. So the developers/coders should keep in mind this issue and pay complete attention in solving it.

Projects of product development have two components social and technical, whose elements are needed to organized and executed in sequential manner to have a successful project. Research in this field is considering two factors: First, the dependencies among the tasks assigned to different developers teams. Second, the co-relation between the entities like tasks, resources, etc.

Projects of product development have two components social and technical, whose elements are needed to organized and executed in sequential manner to have a successful project. Research in this field is considering two factors: First, the dependencies among the tasks assigned to different developers teams. Second, the co-relation between the entities like tasks, resources, etc.

When the coordination analysis is taken out and drawn a pattern of coordination (in Figure 1 case A), if we have high levels of congruence. If the pattern of coordination Requirements and coordination analysis do not match, we have low levels of congruence. (in Figure 1 case B).

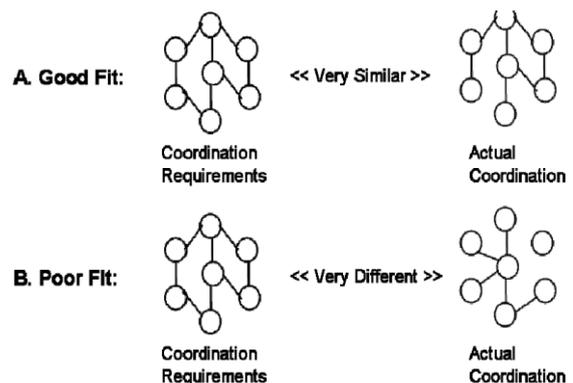


Figure 1: The Concept of Congruence

Fig. 1: The Concept of Congruence

For setting up the coordination of the workers and their activities we need to represent via two state relationships. First is represented by which people are working on which tasks. The second is relationships or dependencies among tasks. The Coordination Requirements matrix is determined as following:

$$CR = TA * TD * TA^T \text{ (Equation 1) [1].}$$

Here, the indications are as follows:

TA: Task Assignments,

TD: Task Dependencies, and

TAT: Transpose of the Task Assignment Matrix.

The given frame work has many different ways to think of the requirements among workers depending on what type of data can be used to populate the TD matrix.

If for a particular set of Software dependency with various tasks, the proportion of coordinated activities that happen is congruence. The congruence relation can be stated as follows:

Congruence (CR, CA) = Diff (CR, CA) / |CR| (Equation 2)[1].

The above equation states that the value of Congruence is the floating/integer value between the binary coordinates i.e. [0,1].

IV. THE REASON OF SUCH SOFTWARE DEPENDENCIES.

What goes wrong in these Software related problems is just due to Bad Design. While writing code, people actually code and they do not care for the dependencies that are being carried out in their code by their inefficient and unsyntactical code.

This problem of BAD design is that for ease of coding the developers/coders directly inherit and then use codes that are multiprogramming dependent. The term suggests that the code is used in many such phases throughout the code written to develop the software.

Once the coder/developer makes changes to the code then the files linking to all those programs that have inherited or are linked with this particular code created **BAD ADDRESS** as this file has been changed/their contents are changed and so this creates an error while we run them.

In order to run them we need to change all the programs or files that are associated/linked to the original code but this is not a feasible solution so we introduce other frameworks in the following section.

V. FRAMEWORKS

Study and research of this framework with appropriate data sets leads to some result which are :

- 1) Task dependencies are formally in clustered form and changing a single component affects the cluster. [1]
- 2) Evolution of coordination requirement and its impact can be calculated from past result is possible with this framework. [1]
- 3) The methods of dependencies, syntactic and logical structure and how they affect the failure of a system can be determined. [1]
- 4) Evolution of coordination behavior is also possible. [1]

VI. APPLICATIONS

The tools for Collaboration and Communication are an important part of the software evolution process. The tools should be grouped on the basis of the role that they play in the evolution of the process. Once they are diversified, then such tools should have the feature to link the software with other in such a way that if any one of them changes then there is change in the remaining of them. There should be tools of various categories:

- 1) Tools for communication.
- 2) Tools for Document management and Software configuration management.
- 3) Tools for Task Management.

In general scenario such tools should not be used alone but in fact they should be used as a whole i.e. as a combination (properly integrating) of one or more tools. This integration of tools can be done in many ways

VII. CONCLUSIONS AND FUTURE WORK

The paper concludes that we can easily find out the faults that occur due to software dependency in the software and hence the pre release and the post release problems related to the software can be reduced and can even be completely removed.

Current work is being made such that data from the very different phases are made available to various departments and units about the dependency they have created if they have access rights so that they can manage the dependencies and can solve them.

ACKNOWLEDGMENT

We(the authors), sincerely thank Prof. Sharnil Pandya and Prof. Kruti Lavingya, Department of Computer Science and Information Technology, Nirma University for helping us throughout this paper.

We also thank anonymous readers for their valuable guidance.

REFERENCES

- [1] http://www.casos.cs.cmu.edu/publications/papers/Dissertation_final.pdf.
- [2] Marcelo Cataldo, Audris Mockus, Jeffrey A. ROBERTS, and James D. Herbsleb, "Software Dependencies, Work Dependencies, and Their Impact on Failures".
- [3] Nachiappan Nagappan, Thomas Ball; Microsoft Research Redmond,USA; "Using Software Dependencies and Churn Metrics to Predict Field Failures: An Empirical Case Study".
- [4] [http://en.wikipedia.org/wiki/Coupling_\(computer_programming\)](http://en.wikipedia.org/wiki/Coupling_(computer_programming)).
- [5] http://pic.dhe.ibm.com/infocenter/tivihelp/v28r1/index.jsp?topic=%2Fcom.ibm.tivoli.tpm.scenario.doc%2Fsoftware%2Fcsfm_reqcap.html.
- [6] <http://www.canonware.com/jemalloc/dependencies.html>.
- [7] <http://software.ac.uk/resources/guides/defending-your-code-against-dependency-problems>.
- [8] Stefan Jungmayr, FernUniversität Hagen, Praktische Informatik III, Universitätsstraße 1, D-58084 Hagen, Germany stefan.jungmayr@fernuni-hagen.de; "Testability Measurement and Software Dependencies".
- [9] Martin P. Robillard McGill University, Canada; "Topology Analysis of Software Dependencies".