ISSN No. 0976-5697

Volume 4, No. 8, May-June 2013

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Assessing Opportunities Of software Reuse for Companies

Kenneth Otula Sigar
Computer Science Department Kabarak University Kenya
kennethkind@gmail.com

Abstract: Software development is now taking shape in most of the organizations. To gain competitive advantage software reuse has become best approach to make it viable. This paper explores and examines the opportunities that can be derived when software reuse as a practice is adopted by companies. It further explains how enterprises can gain competitive edge over others in the same field by adopting and committing to software reuse. In addition to these, this review aims to assess how increased productivity, improved software quality and reduction in all categories of cost can be realized where there are successful software reuse programs. While assessing the benefits of software reuse the paper also brings out facts why software engineering have been majored and centered on the original development of software and also explains why systematic software reuse is used to achieve better software in a more speedy method while incurring lower cost.

Keywords: Software Reuse, Software Development, competitive advantage

I. INTRODUCTION

Software reuse is the activity of developing software products from existing software rather than building software systems from scratch. Software reuse is an effective approach in business that stakeholders can use to catapult software development processes and to improve on the product quality. [1]. some codes within software that were originally designed for a different project and used in another project is referred as reuse. The software reuse is an efficient way in which programmers can probably save time and energy by reducing redundant work during software development which is made up of variety of distinct phases.

In as much as software reuse might be of great importance, it is regarded as one of the least well understood element in the software development life cycle.[2]

To enable companies save on cost of designing, writing and testing new codes, they must develop systems from already well designed and pre-tested components [1] and this has been even used and implemented in many latest programming paradigms like object oriented languages which includes C++,Java and many others. These languages accelerate code reuse through a number of its features like inheritance and extending of a class features to other classes in a program. This clearly saves time and helps programmers to avoid complications and hence reducing their frustrations during the development of projects.

During the usage of the already built code in a program, it's a must that some form of communication between the old code and where it is to be reused is put in place and most common ways of doing that is through a software call and use of a subroutine and this enable the developers to make a reference to the said code during reuse. Software reuse is very crucial in improving productivity if a clear and well cut attention is paid in the development for reuse. [3]

II. TYPES OF REUSE

According to [4] , there are major classification of Reuse which includes opportunistic reuse and phrased reuse. The first type is where the development team heavily depend on the existing components of the software to enable them create another new software with a different function while the second one is about when the development comes up with some common components of the software they intend to use in future during development of other software's.

Opportunistic reuse is further classified [4] into internal reuse and external reuse where the first one is used basically within organization since they might want to hold on vital matters of the code to themselves without leaking them outside [5] and the other one is where the parent developer can put a license on the code before its used by other parties and it comes at cost that has to be paid by these external users of the code.

In re-using the codes when developing a project there are several levels granularity that can be attained [6] and they include the following single line of code, function/procedure,modules,subsyetms,components and the entire program.

III. OPPORTUNITIES DERIVED FROM SOFTWARE REUSE

There are many benefits or opportunities that can be presented when software reuse is put to effective use and adopted systematically in the organization [4, 5] and they can include the following:

A. *Reduce errors /bugs:*

When a code is reused there are likely chances that it will have less errors or bugs if not any at all since it has been fully undergone all kinds of testing and hence bugs are completely denied a chance to reside within the components or code in the software [4,6].

B. *Effective use of software specialists:*

Many software programmers do the same work all through which is time consuming and therefore through software reuse, they can develop reusable codes that can extended to other activities that relate to the developed code [4, 7]

C. *Enhanced and faster development:*

By all standards, it's now clear that reusing of codes enhances and speeds up the rate of development of the software as a product and hence leads to it being delivered to the market as early as not expected and this brings with itself many benefits [4]. This is possible since production and validation and verification time is considerably reduced.

D. Standardization:

Reuse of codes can also bring forth the benefit of making sure that there is uniformity across the development platform for example user interface standards can be implemented as a set of standard reusable components. For example, if menus in a user interfaces are implemented using reusable components, all applications present the same menu formats to users. The use of standard user interfaces improves dependability as users are less likely to make mistakes when presented with a familiar interface as depicted by [4].

E. Reduced production cost:

When one compares the cost of developing a system from scratch and the one developed using software reused techniques, there will be considerable reduction in the production cost in that there are so many stages of development that will not be done afresh. Phases of development like validation, testing and many others will not be repeated since have been fully undergone rigorous process in the initial code development.

F. Increased Dependability:

There will Increased dependability since reused software has been tried and tested in working systems hence a newly created software will be bring forth status of dependability and this can bring many benefits further by even reduction of cost.

G. Improved software quality and competitive advantage:

In the process of software reusability there will be increased software quality since many errors are controlled and this opens a gap for putting the organization in the competitive edge.

IV. CONCLUSION

Software reuse is a technique that can be incorporated in the development of new software in organizations. If a systematic software reuse approach is adopted then there will be numeral benefits and opportunities that can be exploited. The greatest of all is that one of the major opportunities is reduction in cost in the development and improved software quality which places the organization above board hence competitive advantage. This paper concludes that effective adoption and effective use of software reuse can put the organization above its mainstream competitors and therefore it is a vital opportunity that they can tap, adopt and enjoy the explored benefits.

V. REFERENCES

[1]. Kevin D. Wentzel, “Software Reuse - Facts and Myths”, ICSE '94 Proceedings of the 16th International conference on software engineering, 1994, pp 267-268.

[2]. C. Gacek, editor,” Software Reuse: Methods, Techniques, and Tools”, 7th International Conference, ICSR-7, Austin, TX, USA, April 15-19, 2002, Proceedings , volume 2319 of Lecture Notes in Computer Science Springer, 2002.

[3]. M Ramachandran, “Software Reuse Guidelines”, School of Computing and Mathematical Sciences, Liverpool John Moores University, ACM, 1994.

[4]. Sarbjeet Singh, Sukhvinder Singh, Gurpreet Singh, “Reusability of the Software “, International Journal of Computer Applications (0975 –8887) Volume 7 –No.1 4 , October 2010 Software Engineering, Morality, Oxford University Press, vol SE -12 no. 1 1994. Gert B (1988)

[5]. Kenneth M. Anderson, “Software Methods and Tools”, CSCI 3308 - Fall Semester, 2004

[6]. Green R M, ”The Ethical Manager”, Macmillan Publishing, 1994

[7]. Gotterbam and Rogerson, 1998, “The Ethics of Software Project Management”, in Ethics and Information Technology, ed. G& an Collste, New Academic Publisher, 1998.