

Code Reuse & Reusability of the Software

Ms.Kirandeep Kaur¹, Ms.Rekha Rani², Ms.Jagdeep kaur³
¹(CSE, KCCEIT, Nawanshahrr /PTU, INDIA)

-----Abstract-----

Code Reuse is currently one of the most active and creative research areas in Computer Science. Code reuse is the use of existing software to build new software. Reuse and reusability are the two important factors in the software development. Reusability implies some explicit management of build, packaging, distribution, installation, configuration, deployment, and maintenance and upgrade issues. The paper identifies the scope, recent trends and future scope of software reuse.

Keywords: Software Reuse & Reusability, Types, Process.

Date Of Submission: 26 March 2013



Date Of Publication: 15 April 2013

I. INTRODUCTION

Code reuse is called Software reuse. Software reuse is the process of creating new software systems from existing software components. Reuse has an enormous impact on productivity. The following elements of software reused are software specifications, designs, tests cases, data, prototypes, plans, documentation, frameworks and templates. Reusability is the degree to which the artifacts can be reused. The ability to use all or the greater part of the same programming code or system design in another application. Reusability is the segment of source code that can be used to add new functionalities with slight or localizes code modifications when a change in implementation is required. Reusability is the extent to which a software component is able to be reused. Reusable modules and classes increase the prior testing reduce implementation time & use has eliminated bugs & no modification. Reusability is the degree to which the artifacts can be reused. Conformance to appropriate design and coding standards increases a component's reusability.

A set of candidate principles for software reuse are as follows:

- [1] Build software domain architecture as a framework for reuse activities.
- [2] Use a software development process that promotes and controls reuse.
- [3] Reuse more than just code.
- [4] Practice domain engineering.
- [5] Integrate reuse into project management, quality management and software engineering activity.
- [6] Organize the enterprise to facilitate partnering to achieve reuse across product boundaries.
- [7] Use automation to support reuse.
- [8] Couple modern reuses theory and technology with natural, traditional organizational reuse practices

II. TYPES OF SOFTWARE REUSE

The concerning motivation and driving factors, reuse can be:

- [1] **Opportunistic Reuse:** They can use in existing components or projects. Opportunistic reuse can be
- [2] categorized further:
 - Internal reuse: A team reuses its own components
 - External reuse: A team may choose to license a third-party component
- [3] **Planned Reuse:** They will be reusable in future projects.

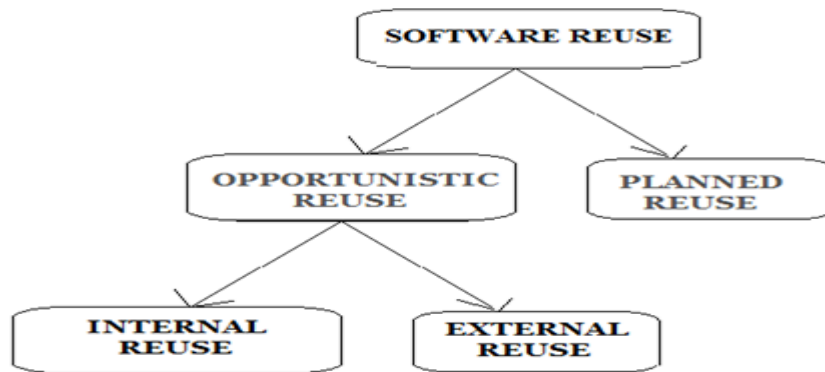


Figure 1: Types of Reuse

III. THE MAJOR STEPS IN A SOFTWARE REUSE PROCESS

The following some important steps are shown in Table 1.

Table 1: Software Reuse Process

S.NO.	SOFTWARE REUSE PROCESS
1	Specifying the object to be created
2	Searching the project, domain, and general databases for reuse candidates
3	Evaluating the candidates to determine which (if any) should be used
4	Modifying, if necessary, to fit specific needs
5	Integrating the reusable component(s)
6	Validating the system including the new component(s)
7	Feeding back the knowledge regarding the payoff of reuse

3.1 BENEFIT OF SOFTWARE REUSE

A reusable component may be code, but the bigger benefits of reuse come from a broader and higher-level view of what can be reused. Software reuse can cut software development time and costs. The major advantages for software reuse are as following:

- Increase software productivity.
- The systematic development of reusable components.
- Improve software system interoperability.
- Shorten software development time.
- Produce better quality software.
- Develop software with fewer people.
- Move personnel more easily from project to project.
- The systematic reuse of these components as building blocks to create new systems.
- Reduce maintenance costs & software development.
- Produce more standardized software.

IV. LIMITATIONS OF SOFTWARE REUSE

Limitations for software reuse are as follows:

Table 2: Limitations of Software Reuse

S.NO	LIMITATIONS
1	It increases the maintenance cost
2	Creating and maintaining a component library
3	Lack of tool support
4	Not-invented-here syndrome
5	Finding, understanding and adapting reusable components

V. CONCLUSION

In this Paper, we consider a few concepts, process and principles that have a positive impact on software reuse. We have reviewed the history of software Reuse & Reusability, the current trends and existing problems, and specific difficulties, which are helpful for increasing reuse and reusability.

REFERENCES

- [1] Sarbjeet Singh, Manjit Thapa, Sukhvinder Singh and Gurpreet Singh, "Software Engineering - Survey of Reusability Based on Software Component", International Journal of Computer Applications (0975 –8887), Volume 8, No.12, PP: 39-42, 2010.
- [2] Nabil Arman, "E-learning Materials Development: Applying and Implementing Software Reuse Principles and Granularity Levels in the Small", International Journal of u- and e- Service, Science and Technology, Vol. 3, No. 2, PP: 31-42, 2010
- [3] Sarbjeet Singh, Sukhvinder Singh, Gurpreet Singh, "Reusability of the Software", International Journal of Computer Applications (0975 – 8887), Volume 7, No.14, PP: 38-41, 2010.
- [4] Ashwin B. Tomar, V. M. Thakare, " A Study of Software Reuse and Models", National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2012) Proceedings published by International Journal of Computer Applications® (IJCA), PP: 14-17, 2012.
- [5] R. Kamalraj, Dr. A. Rajiv Kannan, P. Ranjani, "Stability-based Component Clustering for Designing Software Reuse Repository", International Journal of Computer Applications (0975 – 8887), Volume 27, No.3, PP: 33-36, 2011.
- [6] B.Jalender, N.Gowham, K.Praveen Kumar, K.Murahari, K.Sampath, "Technical Impediments To Software Reuse", B. Jalender et. al. / International Journal of Engineering Science and Technology, Vol. 2, No. 11, ISSN: 0975-5462, PP: 6136-6139, 2010.
- [7] T. Karthikeyan, J. Geetha, "A Study and Critical Survey on Service Reusability Metrics", I.J. Information Technology and Computer Science, Volume 5, PP: 25-31, 2012.
- [8] P.Niranjan, Dr. C.V.Guru Rao, "A Mock-up Tool For Software Component Reuse Repository", International Journal of Software Engineering & Applications (IJSEA), Volume 1, No.2, PP: 1-12, 2010.