

# Study on Software Cloning Concept

Amandeep

Department of Computer Science and Engineering  
Chandigarh University, Mohali, Punjab, India  
aamy.aman29@gmail.com

Akwinder Kaur

Department of Computer Science and Engineering  
Chandigarh University, Mohali, Punjab, India  
akwinderkaurkingra@gmail.com

## ABSTRACT

Reusing of the software by copy and pasting is a frequent activity in software development. The re-implementation of code is called software clones. It leads to the maintenance problems and bug propagation. Anecdotal evidence suggested that occurrence of similarities in code, suggesting that clones are as detrimental to model quality as they are to code quality. This paper presents the study on the software cloning, their reasons for the clone duplications, their pros and cons and their types of the software clones.

## Keywords

Software reuse, Software cloning, code duplication, structural clones, function clones, model based clones, UML (Unified Modeling Language)

## I. INTRODUCTION

An organized and systematically approach is adopted by software engineers regarding their work using some techniques and tools depending upon the resources available and problem to be solved. System engineering is different from software engineering. System engineering is concern with deployment, architectural design and integration where as software engineering is concern with development, quality and testing and control of the system.

Software engineering is the study and application of engineering to the design, development, and maintenance of software. Software engineering consists with forward engineering and reverse engineering. Forward engineering is the traditional process of moving from high-level abstractions and logical, implementation-independent designs to the physical implementation of a system whereas reverse engineering is the process of recreating a design by analyzing a final product. Reverse engineering is the process of discovering the technological principles of a device, object, or system through analysis of its structure, function, and operation. It is common in both hardware and software. Reuse of software and software maintainability is the important phase of the softwaredevelopment life cycle. Software reuse is basically consists of making use of any existing information, component or product when designing and implementing a new system or a product. Replication of

an entire software program does not count as a reuse. Reuse of assets is dependent upon both the similarities and differences between the applications in which the component is being used [1].

The software life cycle comprises of three steps: first we have to clearly define the requirements, implement these requirements; and then we have to maintain the software and evolve it according to user requirements. But from the development point of view maintenance is the most crucial activity in terms of cost and effort. Code clones and Model clones are considered one of the bad smells of software system and indicators of poor maintainability. Various studies show that the software system with code clones is difficult to maintain as compared to non-cloned code software system [1]. In software engineering, software cloning is the concept in which it is prevalent to reuse code fragments by simply copying and pasting the code with or without any modification. Due to copy and pasting software systems often contain sections of code that become almost mirror replica of one another, such a similar code is called code clones or cloning of code. It should be noted that code reuse is a standard practice in modern programming. Some of the major disadvantages of code cloning by using the concept of replication and copy-paste programming that it leads to code bloat that significantly increases the technical depth of software products and making software maintenance expensive and time consuming.

Maintenance of software is one of the most important phases of the software development life cycle. It consists of those changes that are made to a software system after it has been deployed to the client upon client acceptance.

During evolution and maintenance in software system are unavoidable but sometimes it becomes very risky [2]. An inconsistent change in the software without any awareness might cause software in the terrible state. Mostly frequent changes are in the artifacts (class, method, properties) has introduced. Clones of the code and design in software are the major challenges in these days.

“Code Cloning” — the intentional or un-intentional introduction of similar code segments which might be

occur in medium-sized to large software systems. From the survey, most software systems contain a significant amount of code cloning; typically 10–15% of the source code in large software systems is part of one or more code clones. Due to copy and pasting software systems often contain sections of code that become almost mirror replica of one another, such a similar code is called code clones or cloning of code. It should be noted that reuse of code is a standard practice in modern programming. Some of the major disadvantages of code cloning by using the concept of replication and copy-paste programming are that it leads to code bloat that significantly increases the technical depth of software products and making software maintenance expensive and time consuming [3]. The existing research shows that code cloning plays a momentous role in large software applications and management systems.

In cloning of the software, researchers presented evidence that code clones have positive and negative consequences for maintenance activities and thus, in general, code clones are neither good nor bad. It is not possible to eliminate certain clone classes from a software system to minimize their potential threats [4]. Consequently, the identification and management of software clones and the evaluation of their impact has become an essential part of software maintenance. Knowing the evolution of clones throughout a system's history is important for properly comprehending and managing its clones [5]. Software cloning is a controversial issue in the IT field. From the recent survey cloning in software become a prime concerned in the software engineering community [6, 7, 8].

Organization of the paper: Section 2 presents the reasons for the clone duplication, Section 3 presents the cloning in the UML models, Section 4 presents the types of the software clones, Section 5 presents the pros and cons of the software cloning and Section 6 presents the conclusion.

## 2. REASONS OF CLONE DUPLICATION

Software clones never be occurred by themselves. It might be created or generated intentionally or unintentionally by developer or tester or maintenance engineer. There are several factors that influence cloning in the software. Some of them are:

### 2.1 Development strategy

Cloning can be introduced during the reusability and programming approaches.

*i) Reuse Approach:* Reuse of logics, concepts, design, functionality and code. A simple reuse by copy and paste activity, which might be done by the developer to reduce their efforts and for reduce time constraints. As well the reuse of designing causes the cloning. Furthermore, reuse of functionality, logics and code,

forking are the main reasons behind the cloning. Basically forking is the concept which means to reuse of similar solution somewhere with the hope that will be diverged significant with the new software system.

*ii) Programming Approach:* From the programming approach, a similar copying and pasting of same data from existing software to new software. Sometimes to merge the similar systems by programming approach causes cloning.

### 2.2 Maintenance Benefits

Cloning in the software is done for the positive impacts. It might be beneficial for the maintenance of software and it speed up the maintenance.

*i) Avoiding:* From the cloning in the software, it avoids the risk as well unwanted design dependencies during the maintenance phase

*ii) Ensuring:* It ensures the robustness in the software and makes a better performance in the real time programs. It reflects towards the design decisions.

### 2.3 Overcoming underlying limitations

In this strategy, languages and programming dependent limitations are being overcome.

*i) Language Limitations:* Cloning is occurred due to lack of reuse mechanism. To overcome this abstraction is done but it creates complexity. As per abstraction is error-prone and requires significant efforts in making abstraction.

*ii) Programming Limitations:* Basically, the cloning done by the programmers are concerned with the time constrain to the developers which are assigned, lack of ownership, lack of logical knowledge or having a difficulty in understanding the large software systems, or the performance by LOC.

## 3. CLONING IN UML MODELS

UML modeling has become a key industry practice. Though modeling is a well established field but research in detecting clones in those models is still in infancy. We highlight some of the reasons for the presence of model clones:

*i) Model clones through copy/paste:* Ad hoc reuse through copy/paste creates model clones. Developers adopted this as the fast and immediate way of meeting the change.

*ii) Model clones through language limitations:* Sometimes the modeling language fail to factor out the common parts. Thus they essentially appear in the model.

iii) **Programmers Limitation and Time Constraints:** The modeling is sometimes done under severe time constraints. Limitations of programmer's skills and hard time constraints inhibit proper modeling.

iv) **Complexity of the System:** The difficulty in understanding large systems only promotes copying the existing functionality and logic.

v) **Lack of Restructuring:** Developers delay restructuring (refactoring, abstraction, etc.) of model due to time limits. Often, restructuring gets delayed indefinitely.

#### 4. TYPES OF THE SOFTWARE CLONES

There are several ways to define the software clones. Every researcher is leading to their own taxonomy. Mostly there are two types of clones existing in the development phase.

So, software development life cycle contains two types of clones such as:-

1. Code clones
2. Model clones

**Code Clone:** Code clones are the code elements in source file which are identical or similar to each other. The duplicated elements are called clones and the method in which copying or modifying is done is called cloning. Whereas, cloning of the codes are present in the implementation phase which is considered as the coding phase. The implementation phase is just after the designing phase.

**Model Clone:** Like code clones, model clones are the duplicity of the model fragments, its attributes, behavior etc. It is presented in the designing phase of the software development life cycle before the implementation phase and after the feasibility phase.

Some of the code clones or model based clones are classified into two categories:

##### 4.1 Textual Types of clones in software

a) **Type 1 (Exact Clones/ Exact Model Clones):** This type of clone identical code fragments except for variations in whitespace, layout and comments.

b) **Type 2 (Renamed/Parameterized Clones / Renamed/Parameterized Model Clones):** Program fragments which are structurally/syntactically similar except for changes in identifiers, literals, types, layout and comments.

c) **Type 3 (Near Miss Clones/ Near Miss Model Clones):** Program fragments that have been copied with

further modifications like statement insertions/deletions in addition to changes in identifiers, literals, types and layouts.

##### 4.2 Functionality based software cloning

d) **Type 4 (Semantic Clones / Semantic Model Clones):** Program fragments which are functionally similar without being textually similar.

**Structural clones:** These are patterns of interrelated classes emerging from design and analysis space at architecture level. Structural Clones [14] reflect design level similarities which help in maintenance.

**Function clones:** The clones which are limited to the granularity of a function/method or procedure. Several studies depicted the clone detection methods that found the clones at function level which can be extracted in a different procedure.

**Model based clones:** Nowadays graphical languages are replacing the code as core artifacts for system development. Unexpected overlaps and duplications in models [7] are termed as model based clones.

#### 5. PROS AND CONS OF SOFTWARE CLONING

The reasons behind the cloning has been described in section 2, as consequently cloning do occur in large systems as well in medium sized systems. Cloning has severe impacts on quality, usability and maintainability of software systems.

##### 5.1 Pros of Clones

Mostly the software developer introduces the clones by intentionally from the existing source code. Some of the points are discussed below:

- i) It is fast and immediate methods for addressing change in requirements.
- ii) Easy to understand the programming paradigms to encourage the use of template in programming.
- iii) It enhances the functionality of the programming language.
- iv) It promotes the code duplications for the efficiency considerations.

##### 5.2 Cons of Clones

There are mainly cons of the cloning in the source code which makes terrible in the software development life cycle. To oversee the issues required focus. Some of them discussed below:

**i) Increment in the maintenance cost:** [9, 10] from the study, cloning in the software increases the maintenance cost and increases the post maintenance efforts.

**ii) Increases the bug propagation:** Cloning concept also increases the bug propagation by pasting the code elements who having bugs in it to the new version software from the existing software. So it will increase the percentage of bugs in the software.

**iii) Bad Impact:** Cloning in the codes having bad impact in the software life cycle which discourages the use of refactoring, inheritance etc. It leads to bad design practice.

**iv) Impact on system understanding/modification/improvement:** It is quite common that the person who developed the original system is not the one who is maintaining it. Moreover the presence of duplicated code not only complicates the design but leads to decreased understanding thereby hampering improvements and modifications. In the long run, the software may become so complex that even minor changes are hard to make [9, 11].

## 6. CONCLUSION

Software cloning has a bad impact in the software development life cycle. Software cloning contains a number of redundant elements which causes inconsistency in the software systems. There is a healthy and plenty of work is required for model cloning. This paper presented the roadmap of the software cloning for better understandability.

## REFERENCES

- [1] P.A. González-Calero, "Applying Knowledge Modeling and Case-Based Reasoning to Software Reuse", In IEEE Proceedings – Software, Vol. 147, No. 5, pp. 169-177, 2000.
- [2] Ritu Garg and Rajesh Bhatia, "Clone Removal v/s Clone Avoidance", International Journal of Scientific Engineering and Technology, ISSN: 2277-1581, Volume No.3 Issue No.6, pp: 765-766, 2014.
- [3] L. Aversano, L. Cerulo, and M. D. Penta, "How clones are maintained: An empirical study", In Proceeding of the 11th European Conference on Software Maintenance and Reengineering (CSMR'07), pp. 81, 2007.
- [4] T. Bakota, R. Ferenc, and T. Gyim\_othy, "Clone Smells in Software Evolution", In Proceeding of the 23<sup>rd</sup> IEEE International Conference on Software Maintenance (ICSM'07), pp.24, 2000.
- [5] L. Barbour, F. Khomh, and Y. Zou, "Late Propagation in Software Clones", In Proceeding of the 27<sup>th</sup> IEEE International Conference on Software Maintenance (ICSM'11), pp. 273, 2004.
- [6] J. Mayrand, C. Leblanc, E.M. Merlo, "Experiment on the automatic detection of function clones in a software system using metrics", In Proceedings of the 12th International Conference on Software Maintenance (ICSM'96), 1996.
- [7] T. Mende, F. Beckwermert, R. Koschke, G. Meier, "Supporting the Grow-and-Prune model in software product lines evolution using clone detection", In Proceedings of the 12th European Conference on Software Maintenance and Reengineering, Szeged, Hungary, pp. 163–172, 2008.
- [8] R. Koschke, "Frontiers of software clone management", In Proceedings of Frontiers of Software Maintenance (FoSM'08), Beijing, China, pp. 119– 128, 2008.
- [9] Mohammed Abdul Bari and Dr. S Ahmed, "Code Cloning: The Analysis, Detection and removal", International Journal of Computer Applications, Vol 20, 2011.
- [10] Haon Anh Nguyen, Tung Thanh Nguyen, Pham, N.H., Al-Kofahi, "Clone Management for Evolving Software" Software Engineering, IEEE Transactions on, vol.38, no.5, pp. 1008-1026, 2012.
- [11] Pham, Nam, "Complete and accurate clone detection in graph-based models", Proceedings of the 31st International Conference on Software Engineering. IEEE Computer Society, 2009.