International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 12 | Dec 2022 www.irjet.net p-ISSN: 2395-0072

Clone of an organization

Miss Vidhi Gupta, Mr. Anand Tiwari, Mrs. Diksha Tiwari

[1] *Undergraduate student, Krishna Institute of Technology, Uttar Pradesh, India*
[2] *Asst. professors, Krishna Institute of Technology, Uttar Pradesh, India*
[3] *Asst. professors, Krishna Institute of Technology, Uttar Pradesh, India*

Abstract – *This work present the clone of any organization which helps to manipulate the designing or features of an organizations and testing process so that original site did not get disturb by any of them.*

This will helps people in testing of any site or will do many tasks by the help of clone.

Clone is a concept where we can make an exact copy of a thing and clone will do all the works same as original but if it get any interruption of or any damage then it will not able to do any work.

Key Words*: *Clone, database clone, domain clone, software testing, hacking, software reverse engineering, software development life cycle.

1. INTRODUCTION

Cloning is the process of generating a genetically identical copy of a cell or an organism. Cloning happens often in nature—for example, when a cell replicates itself asexually withoutany genetic alterationor recombination.

Prokaryotic organisms (organisms lacking a cell nucleus) such as bacteria create genetically identical duplicates of themselves using binary fission or budding. In eukaryotic organisms (organisms possessing a cell nucleus) such as humans, all the cells that undergo mitosis, such as skin cells and cells lining the gastrointestinal tract, are clones; the only exceptions are gametes (eggs and sperm), which undergo meiosis and genetic recombination. This is all base on the concept of medical science, when we come to know that in the field of technology and engineering, so the cloning is the process when the creation of replica of any software, technology, or an organization.

1.1 Different things we can built as a clone

Frontend design

We can build a website look like similar to any other website.

We can implement same design fonts etc. Frontend would be easy to maintain as a purpose of clone.

Backend design

Sometime it would be difficult to maintain a side as in working but if we have great knowledge of data structure and algorithm then it is easy to work as like same or original site.

Backend clone is easy to maintain with the high data configuration.

Database clone

There are a multiple of ways you could create a database clone. Standard Backup and restore is one method. Export/Import is another. Third party tools are also available for setting up Database Cloning.

Now we can create database clone also with the help of ORACLE, MySQL.

Domain

Domain is not possible to create a clone because domain will be taken by the servers which are not possible to manage 2 sites with the same domain.

1.2 Software testing

Software testing is a process where developer checks the requirement of the client that software is actually work or look like according to the client requirements. Software is defect free. The purpose of software testing is identifying errors, gaps or missing requirements in contrast to actual requirement. Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

1.2 Benefits of software testing

- **Cost-Effective:** It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

- **Security:** It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

- **Product quality:** It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.

- **Customer Satisfaction:** The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

2. Table of types of Software testing

Categories of testing	Types of testing
Functional testing	• Unit testing • Integration testing • Smoke • UAT(User Acceptance Testing) • Localization • Globalization • Interoperability
Non-Functional testing	• Performance • Endurance • Load • Volume • Scalability • Usability
maintenance	• Regression • maintenance

Testing strategies in software engineering

3. Software reverse engineering

Reverse engineering (also known as **backwards engineering** or **back engineering**) is a process or method through which one attempts to understand through deductive reasoning how a previously made device, process, system, or piece of software accomplishes a task with very little (if any) insight into exactly how it does so. It is essentially the process of opening up or dissecting a system to see how it works, in order to duplicate or enhance it. Depending on the system under consideration and the technologies employed, the knowledge gained during reverse engineering can help with repurposing obsolete objects, doing security analysis, or learning how something works.

Although the process is specific to the object on which it is being performed, all reverse engineering processes consist of three basic steps: Information extraction, Modeling, and Review. Information extraction refers to the practice of gathering all relevant information for performing the operation. Modeling refers to the practice of combining the gathered information into an abstract model, which can be used as a guide for designing the new object or system. Review refers to the testing of the model to ensure the validity of the chosen abstract. Reverse engineering is applicable in the fields of computer engineering, mechanical engineering, design, electronic engineering, software engineering, chemical engineering, and systems biology.

Software Reverse Engineering is a process of recovering the design, requirement specifications and functions of a product from an analysis of its code. It builds a program database and generates information from this.

The purpose of reverse engineering is to facilitate the maintenance work by improving the understandability of a system and to produce the necessary documents for a legacy system.

Reverse Engineering Goals:

- Cope with Complexity.
- Recover lost information.
- Detect side effects.
- Synthesize higher abstraction.
- Facilitate Reuse.

3. Software development life cycle

Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

There are different software development lifecycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models**."

- Waterfall model
- RAD model
- Spiral model
- Hybrid model
- Prototype model
- V-model
- Incremental model
- Agile model
- Iterative model
- Big-bang model
- Prototype model

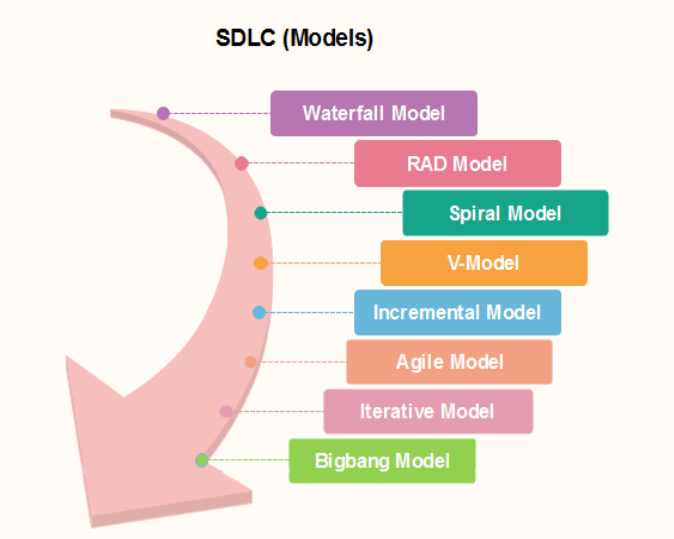

Figure -1: software development life cycle

1) Waterfall Model

The waterfall is a universally accepted SDLC model. In this method, the whole process of software development is divided into various phases.

The waterfall model is a continuous software development model in which development is seen as flowing steadily downwards (like a waterfall) through the steps of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Linear ordering of activities has some significant consequences. First, to identify the end of a phase and the beginning of the next, some certification techniques have to be employed at the end of each step. Some verification and validation usually do this mean that will ensure that the output of the stage is consistent with its input (which is the output of the previous step), and that the output of the stage is consistent with the overall requirements of the system.

2) RAD Model

RAD or Rapid Application Development process is an adoption of the waterfall model; it targets developing software in a short period. The RAD model is based on the concept that a better system can be developed in lesser time by using focus groups to gather system requirements.

- Business Modelling
- Data Modelling
- Process Modelling
- Application Generation
- Testing and Turnover

3) Spiral Model

The spiral model is a **risk-driven process model**. This SDLC model helps the group to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc. The spiral technique is a combination of rapid prototyping and concurrency in design and development activities.

Each cycle in the spiral begins with the identification of objectives for that cycle, the different alternatives that are possible for achieving the goals, and the constraints that exist. This is the first quadrant of the cycle (upper-left quadrant).

The next step in the cycle is to evaluate these different alternatives based on the objectives and constraints. The focus of evaluation in this step is based on the risk perception for the project.

The next step is to develop strategies that solve uncertainties and risks. This step may involve activities such as benchmarking, simulation, and prototyping.

4) V-Model

In this type of SDLC model testing and the development, the step is planned in parallel. So, there are verification phases on the side and the validation phase on the other side. V-Model joins by Coding phase.

5) Incremental Model

The incremental model is not a separate model. It is necessarily a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, each cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

6) Agile Model

Agile methodology is a practice which promotes continues interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.

Any agile software phase is characterized in a manner that addresses several key assumptions about the bulk of software projects:

1. It is difficult to think in advance which software requirements will persist and which will change. It is equally difficult to predict how user priorities will change as the project proceeds.

2. For many types of software, design and development are interleaved. That is, both activities should be performed in tandem so that design models are proven as they are created. It is difficult to think about how much design is necessary before construction is used to test the configuration.

3. Analysis, design, development, and testing are not as predictable (from a planning point of view) as we might like.

7) Iterative Model

It is a particular implementation of a software development life cycle that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. In short, iterative development is a way of breaking down the software development of a large application into smaller pieces.

8) Big bang model

Big bang model is focusing on all types of resources in software development and coding, with no or very little planning. The requirements are understood and implemented when they come.

This model works best for small projects with smaller size development team which are working together. It is also useful for academic software development projects. It is an ideal model where requirements are either unknown or final release date is not given.

9) Prototype Model

The prototyping model starts with the requirements gathering. The developer and the user meet and define the purpose of the software, identify the needs, etc.

A '**quick design**' is then created. This design focuses on those aspects of the software that will be visible to the user. It then leads to the development of a prototype. The customer then checks the prototype, and any modifications or changes that are needed are made to the prototype.

Looping takes place in this step, and better versions of the prototype are created. These are continuously shown to the user so that any new changes can be updated in the prototype. This process continues until the customer is satisfied with the system. Once a user is satisfied, the prototype is converted to the actual system with all considerations for quality and security.

3. CONCLUSIONS

Conclusion of this cloning is to be differentiating or create another ways of testing without interrupt the original site.

There are lot of data stored in an organizations some are running their own business through an organization, so if hackers or programmers want to learn and complete their needs of testing and manipulating new features on any organization so they are free to implement their test on this clone and all the data will be recorded and this data will be secured. Their practices would not get waste and people can able to learn lot things.

REFERENCES

[1] https://www.ibm.com/in-en/topics/software-testing.

[2] https://www.javatpoint.com/software-engineering-sdlc-models

[3] https://www.geeksforgeeks.org/software-engineering-reverse-engineering/

BIOGRAPHIES

Miss Vidhi Gupta
Under Graduate student of B.Tech,
Computer science engineering
Of Krishna institute of technology,
Kanpur, Uttar Pradesh, India

Mrs. Diksha Tiwari
Asst. professor in Krishna Institute
Of technology, Kanpur, Uttar
Pradesh, India.

Mr. Anand Tiwari
Asst. professor in Krishna Institute
Of technology, Kanpur, Uttar
Pradesh, India.