

# ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

## College of Engineering

## Department of Software Engineering

## SCD Project

**BY**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Laelay Temesgen | 0746/13 |
| Kirubel Ateka | 0734/13 |
| Kirubel Dagnachew | 0735/13 |
| Natnael Endale | 1008/13 |
| Natnael Mulugeta | 1020/13 |
| Desalegn Sendek | 1553/13 |
| Mitiku Abebe | 0904/13 |

Submition Date: December, 2024

Submittes to Mr. Gzatie Desalegn

Table of Content

[Big Bang Model: 3](#_Toc185420889)

[Design: 3](#_Toc185420890)

[Steps in the Big Bang Model: 3](#_Toc185420891)

[When to use it and where not to: 4](#_Toc185420892)

[Features of the Big Bang Model: 4](#_Toc185420893)

[Pros of the Big Bang Model: 4](#_Toc185420894)

[Cons of the Big Bang Model: 4](#_Toc185420895)

[References 5](#_Toc185420896)

# Big Bang Model:

The **Big Bang Model** is a simple and straightforward approach to software development that requires minimal planning and focuses on immediate implementation. This model is named after the "Great Big Bang" theory, which led to the formation of the universe (galaxies, stars, planets…). Similarly, in this model, the product is gradually built as requirements emerge, often without a clear understanding of the final outcome. It is the simplest model in Software Development Life Cycle as it requires almost no planning. However, it requires lots of funds and coding and takes more time. This SDLC model combines time, effort, and resources to build a product. The product is gradually built as the requirements from the customer come, however, the end product might not meet the actual requirements.

In the Big Bang Model, development begins with the necessary inputs such as time, effort, and resources. In this model, developers do not follow any specific process. The process is highly flexible, allowing developers to start coding and integrating modules as soon as they receive requirements. This model is particularly suitable for small projects, experimental projects, or projects with undefined or changing requirements.

# Design:

The product requirements are understood and implemented as they arrive. The complete modules or at least the part of the modules are integrated and tested. All the modules are run separately and the defective ones are removed to find the cause.  It is a suitable model where requirements are poorly understood and the final release date is not given. In simple, it can be phased out in 3 points i.e.

1. Integrate each individual’s modules to give a unique integrated overview
2. Test each module separately to identify any errors or defects
3. If any error is found then separate that module and identify the cause of the error

# Steps in the Big Bang Model:

1. Requirement Gathering (Minimal): A basic understanding of the project requirements is gathered. This phase is often brief and may not be fully defined.
2. Design and Development**:** Developers start coding immediately, often without a detailed design phase. As requirements emerge, they are implemented directly into the code.
3. Testing and Debugging: Testing is typically done concurrently with development, and debugging efforts are focused on fixing issues as they arise.
4. Deployment: Once the software is deemed functional, it is deployed to the end-users.

# When to use it and where not to:

This SDLC model is suitable for small projects when few people are working on the project like an academic project or a practical project, the customer’s demands are not exact and keep changing and the release date is not confirmed, or if it is a side-project. As there is no proper planning in this model it is considered the worst SDLC model and is highly unsuitable for large projects. When to use Big Bang Model?

It is recommended to go for the Big Bang model only due to the following cases i.e.

1. Developing a project for learning purposes or experiment purposes.
2. No clarity on the requirements from the user side.
3. When newer requirements need to be implemented immediately.
4. Changing requirements based on the current developing product outcome.
5. No strict guidelines on product release or delivery date.

# Features of the Big Bang Model:

* Not require a well-documented requirement specification
* Provides a quick overview of the prototype
* Needs little effort and the idea of implementation
* Allows merging of newer technologies to see the changes and adaptability

# Pros of the Big Bang Model:

* There is no planning required for this.
* Suitable for small projects
* Very few resources are required.
* As there is no proper planning hence it does not require Management ial employees
* Easy to implement, and manage
* It develops the skills of the newcomers
* Very flexible for the developers working on it
* Simple Model

# Cons of the Big Bang Model:

* Not acceptable for a large project.
* Precarious model and uncertain
* Might be expensive if requirements are not clear
* Poor model for ongoing projects

# References

**[1] Tutorials Point (Website).** (n.d.). Big Bang Model. Retrieved from <https://www.tutorialspoint.com/sdlc/sdlc_bigbang_model.htm>

**[2] GeeksforGeeks (Website).** (n.d.). Overview of Big Bang Model. Retrieved from <https://www.geeksforgeeks.org/overview-of-big-bang-model/>

**[3] Codecademy Course.** (2023, May 16). Big Bang Model [Online course]. Retrieved from <https://www.codecademy.com/resources/docs/general/software-development-life-cycle/big-bang-model>

[4] JavaTpoint. (n.d.). Big Bang Model in Software Engineering. Retrieved from <https://www.javatpoint.com/software-engineering-big-bang-model>