LOW CODE & NO CODE SOFTWARE DEVELOPMENT OF THE FUTURE

Abhishek Gupta
School of Computer Applications, Babu Banarasi Das University Lucknow, India
Email: abhishekgupta86978@bbdu.ac.in

Abstract

In this article, we will examine and discuss methods that require low mathematics and those that do not involve mathematics in software development. How fair is it compared to today's software development process? What is low-code and no-code? We will also look at the difference between low numbers and zero numbers.

INTRODUCTION

How are software developed traditionally? Software is developed using the SDLC (Software Development Lifecycle) and there are a few models for SDLC. The agile SDLC model uses a combination of incremental and iterative processes, focuses on change and customer satisfaction, and quickly delivers the software products thecustomer wants. The agile process breaks the software product into small increments and checks the structure. This is how a project iteration is done and usually this iteration takes 1-3 weeks to develop.

Requirement collection

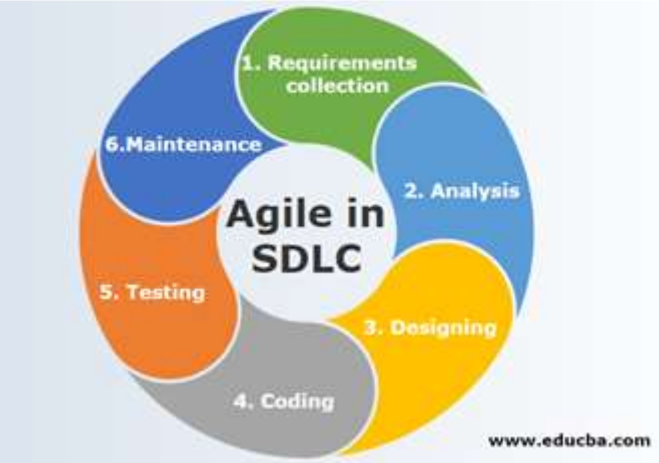
Analysis

Designing

Coding

Testing

Maintenance



How to use low-code, no- code software development? Before answering the question of how to create software, let's first look at what free code is. It is a low-code, no-code visual software development environment platform. This means that it is software where users can drag and drop elements such as buttons, containers, etc. and link them together without having to. or more Line runs less. It's definitely a quick way to create software or a website. We'll also take a look at how the IT industry is changing with the introduction of low-cost computing.

LITERATURE REVIEW

What do you need today as a developer? Software is the key to everything we do and since everything is now based on data we interact with different software and manage our data. The process of planning, building and testing the software we use every day. Developing, distributing and maintaining any software takes a lot of work. This can become a tedious task if not done correctly. Developers will be happy if development is a little easier and takes less time.

How did it all start for Low Code No Code? The term "permission" was coined by Forrester in 2014 for an environment that emphasizes development simplicity and ease of use. The platform is designed so that users and developers of all skill levels can create software without having to understand the coding aspects of development. It is directly proportional to the skills of the developers who create the software. This creates a conflict in software quality and there is no better solution to this problem. Bad software takes more time to test and develop, and this problem arises from the very beginning of software development. This paves the way for low-code operations without code egress and solves long-term operational issues in SDLC.

The evolution of low-code and no-code models

Employees suggest some improvements. Understand. But what's the difference Low-code no-code, low-code no-code may require little or no knowledge of tools or development environment Generally just drag and drop the job into place. A RAD tool is designed to work for use by the person who created it or by another person who participated in its creation.

LCNC applications are powerful enough to be used in many departments throughout the business. Where does a low-code, no-code platform fit into the big picture? SDLC is the software development life cycle. To be fair, LCNC can only reduce coding time in development; The time spent on planning, design and testing will remain the same. Depending on the job, not only the work of the developer will decrease and it may come to a point where there will not be much demand for developers, but on the other hand, software team construction will also need design knowledge to guide users. Cost of user interface development, Tesers are responsible for finding bugs in the system and designers are responsible for designing and creating the skeleton of the entire system, but these are all subtle and slow changes.

Why is low-code, no-code inevitable?

There is also the benefit of investing less in training the company's software development staff. This can reduce the cost of software development, and the consistency and quality of the software can be greatly improved. > Lots of coding, that's why it's called low code, which means less coding than usual. This just makes the developer's job easier because he doesn't have to sit down and code every little thing.

How to get started with low code and nocode?

But as discussed in the page above, proper planning, user interface design, testing and application development are equally important for creating an endeavor or software. Without all these, there is no point in learning "How to create low-code and no-code applications".

App pie

Air Table

AppSheet

JotForm table

Quixy

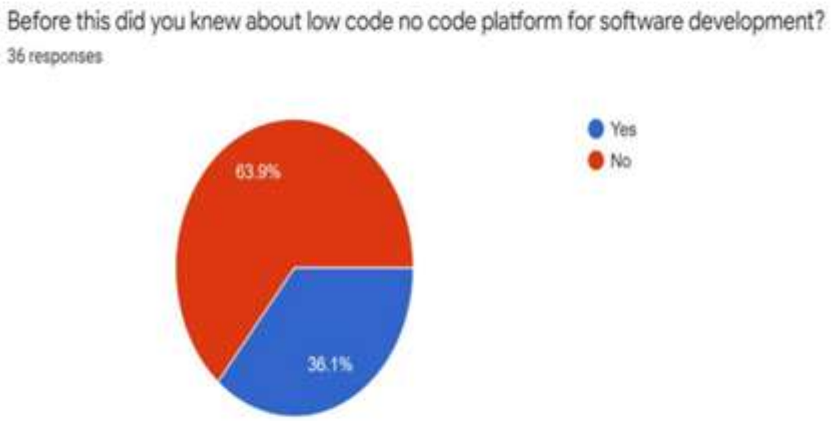
What are the disadvantages of low and low numbers and how can this be improved? Help reduce coding and improve time. It is too early to discover other shortcomings, but eventually it will mature enough to overcome some of the current problems. The approach is to integrate test models; Thus, once the application is created, it can be tested using the same software and provide information about the application's performance, data pipeline detailing, error detection and these applications need more time.

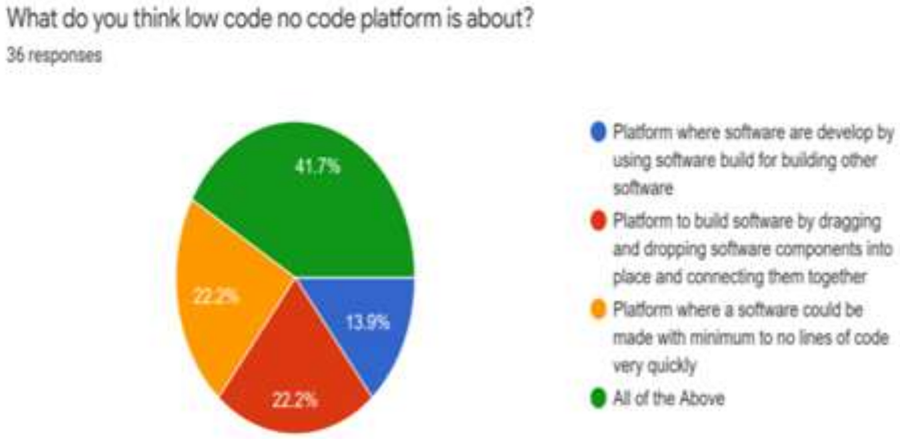
RESEARCH FINDINGS

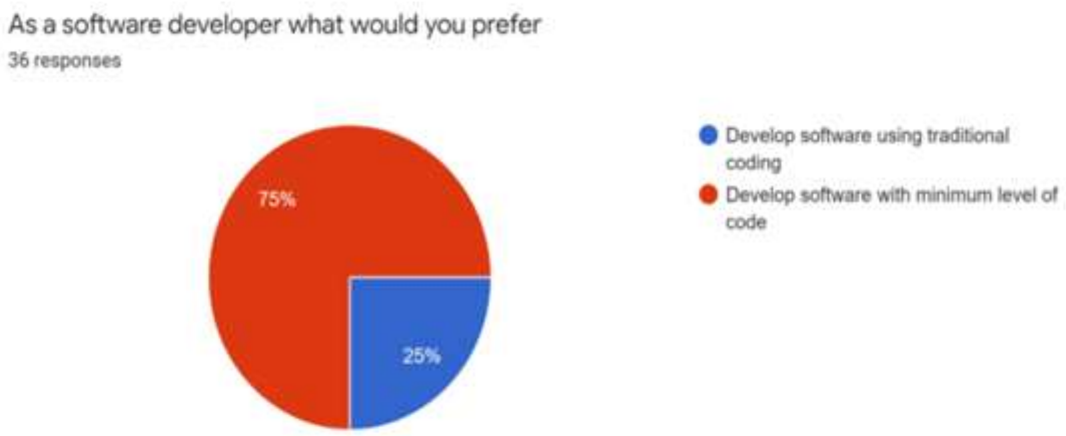
To understand the low-numbers, no-mathematics knowledge, we conducted a small survey of more than 35 people, mostly software developers, graduate students, and a few entrepreneurs. The results of the same survey are as follows: The figure above shows that low-code and no-code platforms are known by 63.9% of people. Even though it's a new platform, he understands that working with a lot of people means creators are taking photos of him ona regular basis. Although about 36.1% of people know about the platform, this is not surprising because the platform is still new. Although the platform is new to many people, the platform is performing well as the results are reliable and will be available to many people soon. Imagine a platform with software that could beused to build other software by dragging and dropping things into space and connecting them together with little to prevent coding.

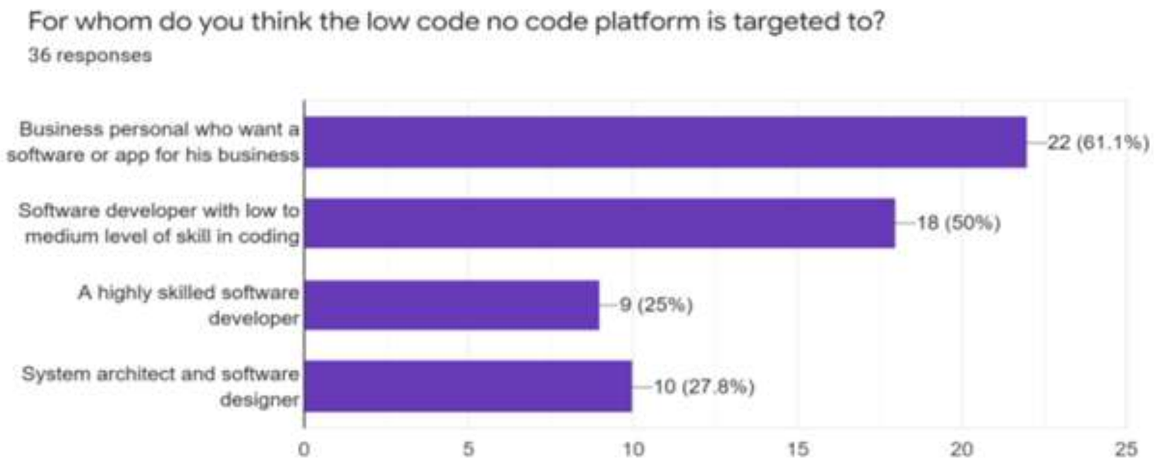
Approximately 58.3% of the other results are correct, but there is incomplete information on the platform. The above results show what the preferences of software developers are like today About 75% of software developers want to design and develop software with minimal coding. The platform is similar to codeless software development. This result shows what people think about the target audience of a low-code, no-code platform. Because the main purpose of the software is to solve business problems.

There is nothing wrong with that either. As a new developer, some of you may want to quickly develop software for your ideas.









A software developer allows you to use your ideas quickly and focus on other things rather than spending too much time on coding and development.

For engineers and designers, this may not be a problem because the work they do does not require a level of coding.

This chart shows us how many people think no-code platforms will be the future of software development. the platform needs more development before we start development.

No need to code anymore.

CONCLUSION

Let's see where all this ends up being, it's a low-code, no-code future. Here and now.

Now. This is already happening, but there is still some work to be done before we start development. used during development.

It has a very low learning curve, which means less learning time and more productivity.

Amount of coding required.

The platform is still in the early stages of development and will take some time to enter the development process.

ACKNOWLEDGEMENTS

I would like to thank all my professors, developer friends, and everyone who helped me with my research. Thank you to everyone who helped me with my research, I couldn't have done this alone without you.

REFERENCES

https://kissflow.com/low-code/history-of-low-code-development-platforms

https://kissflow.com/low-code/low-code-overview

https://www.infoq.com/articles/low-code-community-developer

https://searchsoftwarequality.techtarget.com/definition/low-code-no-code-development-platform

https://www.softwaretestinghelp.com/low-code-development-platforms

https://www.g2.com/categories/no-code-development-platforms