INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

****

EVALUATING THE EFFECTIVENESS OF LOW-CODE PLATFORMS VS. TRADITIONAL CODING IN CAPSTONE SOFTWARE DEVELOPMENT PROJECTS

JOHANNA E. AVE
NU Las Piñas, Philippines
avejohanna@gmail.com

ABSTRACT

This paper looks at how well low-code development works compared to traditional high-code software development for the BSIS students capstone projects. With low-code platforms becoming more popular and important for developers, educators, and organizations, this study compared the two methods based on how fast they are, how easy they are to learn, the quality of the code, how satisfied users are and the ability of learners core programming skills. This comparison will provide justifications for institutions to adopt low-code software development for BSIS student capstone projects. Through the use of surveys and interviews, the researcher will investigate how low-code development will affect the efficiency, flexibility, and quality of software projects in comparison to traditional coding practices. The T-test will be used to evaluate the effectiveness of low-code development in the capstone project by comparing the key performance metrics of the two approaches based on development speed, code quality, user satisfaction, learning curve, and core programming skills.

Keywords: *low-code, high-code, capstone-project, programming-skills, development-speed, code-quality, user-satisfaction, learning-curves, core-programming-skills*

****

****

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

****

INTRODUCTION

Low-code development platforms provide significant traction in the software development landscape, where developers can build applications rapidly and experience minimal code, which lessens the complexity of doing long codes using visual interfaces and pre-built components. These platforms contribute to accelerating the development process and reducing the hassle of making complicated codes. However, despite the growing adoption of this approach, the effectiveness of these tools compared to traditional code-based development remains a subject of debate, particularly in the context of education settings such as the capstone project.

The research capstone project is one of the major requirements of BSIS students before they graduate from the said program. Their participation typically brings them experiences in software engineering and offers the opportunity to evaluate the practical application of the development tools. The projects require technical proficiency, creativity, and problem-solving skills, which are important components for the BSIS students to possess. The choice of development approach, even if it is low-code or traditional code-based development, can influence a student's learning experience and the quality of the final product. By this, it is crucial to justify if the student should adopt this low-code approach to their capstone project without compromising the effectiveness of their learning. This study seeks to evaluate the effectiveness of low-code development versus traditional code-based development in the context of capstone software development projects, based on its development speed, code quality, learning curves, satisfaction and core programming skills. This study aims to provide

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

answers on how these approaches compare in an academic environment by making a statistical analysis using the T-test approach to assess whether significant differences exist between two groups of BSIS students who use low-code platforms versus those who utilize traditional coding methods in their previous System Analysis and Design course.

Hypothesis:

Null Hypothesis (H_0): There is no significant difference in the responses of two groups of the respondents in terms of development speed, learning curves , code quality, user satisfaction, and core programming skills between students using low-code platforms vs. traditional high-coding methods for their SAD project.

Alternative Hypothesis (H_1): There is a significant difference in the responses of two groups of the respondents in terms of development speed, learning curves , code quality, user satisfaction, and core programming skills between students using low-code platforms vs. traditional high-coding methods for their SAD project.

The survey results will give useful information about how well the two development approaches compare in the academic software development project. These results will either support or reject the hypotheses.

METHODOLOGY

This outlines the method used in how the needed data of the study would be gathered and analyzed in order to meet the objectives and goals of the study.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Research Design

The researcher will employ a descriptive research approach, gathering and applying data analysis and research questions to the study. Descriptive studies, in which the researcher interacts with the participants, may involve surveys or interviews to collect the necessary information. It is used to determine participants' response to the effectiveness of low code compared to traditional high code based on speed development, code quality, learning curves, user satisfaction, and core programming skills. The purpose of descriptive research is to observe, describe, and document aspects of a situation as it naturally occurs and to collect data that will describe individuals, groups, or situations.

The researcher will use survey methods to evaluate the effectiveness of the low-code platform versus the traditional high coding approach in terms of development speed, learning curve, code quality, user satisfaction, and core programming skills for the success of software development project.

Participants:

The participants of this study are BSIS students who used a non-code approach and traditional code-based development for their SAD (System Analysis and Design) and were required to complete a capstone project. The sample consists of 130 students divided into two groups.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Group 1 (Low-Code Group): 65 students who used the non-code approach to their System Analysis and Design course.

Group 2 (High-Code Group): 65 students who used the code-based approach to their System Analysis and Design course.

Data gathering:

The data collection process will involve multiple sources:

- **Survey** - an instrument used to distribute to BSIS students to gather responses and information prior to their experience with software development, their experience with low-code platforms or code-based development, and their expectations for the project.
- **Interview** - The researcher will conduct interviews to people with knowledge and experience in the study. Most interviews are non-formal.
- **Internet.** The World Wide Web is the most powerful tool in retrieving information from all corners of the planet. The researcher made use of this advantage in getting all the needed information, supporting facts and guidelines that needed in the study. Website addresses were listed on the appendices for reference purposes.

Project Performance Metrics

This metrics use to evaluate the effectiveness of low-code versus traditional coding in capstone software development project base on the following criteria.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

- **Development Speed.** It is a time taken to complete the students the software development project (measure in time or weeks)
- **Code Quality.** Assessment base on the rubrics in best practices in software development. (e.g., code readability, maintainability, and error-free functionality).
- **Learning Curve:** Representation of how quickly a BSIS students can learn something over time.
- **Developer Satisfaction:** A post-project survey will assess participant satisfaction with the development process using a Likert scale (1–5) on factors such as ease of use, frustration levels, and perceived effectiveness of the tools.

Core Programming Skills: Assess the programming ability to design and evaluate the proficiency level of strength and growth of programming areas.

Data Analysis:

The collected data will be analyzed using the following steps:

- **Descriptive Statistics**. Using the mean and standard deviation each group compute the key variables such as development time, code quality, and satisfaction scores.

$$\mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Standard Deviation Formula

- **T-test.** For hypotheses test, the independent sample will be use to compare the two groups based on performance indicator.

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{\sqrt{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)}}$$

T-test Formula

- **Weighted Mean** was used to determine the weight of the frequencies observed on the gathered data. The researcher will employ this statistical method. The following table is the corresponding interpretation of the gathered weighted means.

Weighted Mean Value Corresponding Interpretation

4.21-5.00 Strongly Agree(SA)

3.41-4.20 Agree(A)

2.61-3.40 Uncertain(U)

1.81-2.60 Disagree(D)

1.00-1.80 Strongly Disagree(SD)

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

****

RESULTS AND DISCUSSION

The information gathered was presented, analyzed and interpreted from the instruments used as conducted in this study. The survey questionnaires were given to the respondents in order to validate the Effectiveness of Low-code approach vs. Traditional code-based software development to their Capstone Project. These questionnaires were tabulated, computed using the T-test, documented to determine the feasibility and effectiveness of the research.

Table 1

Summary and Effectiveness of Low-Code vs. Traditional High-Code of BSIS Third Year Students

Software Development in terms of Speed Development

ITEM	LOW-CODE GROUP						HIGH-CODE GROUP						Critical Value	Computed- T	Remarks
	5	4	2	1	WEIGHTED MEAN	SD	5	4	2	1	WEIGHTED MEAN	SD			
Q1. Low-code platforms significantly speed up the development process compared to traditional high-code development.	19	34	6	6	**3.83**	**1.21**	18	35	7	5	**3.83**	**1.18**	**1.97867**	**0.0000**	**Accepted H0 / Failed to Accept H1**
Q2. The speed of high-code development is often	16	33	8	8	**3.63**	**1.32**	18	32	10	5	**3.74**	**1.24**	**1.97867**	**0.48968**	**Accepted H0 /**

****

****

INSTABRIGHT e-GAZETTE

ISSN: 2704-3010

Volume VI, Issue IV

May 2025

Available online at https://www.instabrightgazette.com

****

hindered by the need for extensive debugging and testing compared with low-code.														
Q3. The use of low-code platforms accelerates the delivery of software solutions compared to using high-code development methods.	12	38	11	4	**3.66**	**1.15**	14	36	10	5	**3.66**	**1.20**	**1.97867**	**0.09701**	**Accepted H0 / Failed to Accept H1**
Q4. I believe low-code platforms can significantly reduce the time required to develop and deploy software compared to traditional coding methods.	15	35	10	5	**3.69**	**1.21**	20	33	7	5	**3.88**	**1.17**	**1.97867**	**0.91010**	**Accepted H0 / Failed to Accept H1**

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Table 2

Comparison Between the two groups response to the Effectiveness of Low-Code vs. High-Code in terms of Development Speed

SD- Standard Deviation note: n1=65 n2=65

df=130-2=128

LOW-CODE GROUP		HIGH-CODE GROUP		Critical Value	Computed T-Value	Remarks
Mean	**SD**	**Mean**	**SD**			
3.83	1.21	3.83	1.18	1.97867	0.0000	Accepted H0 / Failed to Accept H1
3.63	1.32	3.74	1.24	1.97867	0.48968	Accepted H0 / Failed to Accept H1
3.66	1.15	3.68	1.20	1.97867	0.09701	Accepted H0 / Failed to Accept H1
3.69	1.21	3.88	1.17	1.97867	0.91010	Accepted H0 /Failed to Accept H1

Based on the table results, the first group of respondents (third-year BSIS students who used low-code development on their SAD project) and the second group of respondents (third-year BSIS students who used high-code development) both agree with the effectiveness of low-code compared with high-code in terms of development speed.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

****

For the overall development speed (Q1), both low-code and high-code users gave a value of 3.83, indicating that they both agree that low-code development enhances overall speed. In terms of debugging and testing speed (Q2), despite the slight difference, the T-value of 0.48968 remained below the critical threshold, indicating no significant difference. Respondents still perceived low-code to be more effective in speeding up debugging and testing. For the delivery speed of the software solution (Q3), it shows no significant difference, and the perception of two respondents remained that low-code accelerates delivery. In terms of reduction in development and deployment time (Q4), although the high-code group's response was slightly higher than the low-code group's, both still agreed in reducing the overall time.

Table 3

Summary and Effectiveness of Low-Code vs. Traditional High-Code of BSIS Third Year Students Software Development in terms of Learning Curves

SD- Standard Deviation

note: n1=65
n2=65
df=130-2=128

ITEM	LOW-CODE GROUP						HIGH-CODE GROUP						Critical Value	Computed-T	Remarks
	5	4	2	1	WEIGHTED MEAN	SD	5	4	2	1	WEIGHTED MEAN	SD			
Q1. I feel more confident in using a low-code platform compared to	9	39	9	8	**3.49**	**1.25**	13	31	16	5	**3.48**	**1.19**	**1.97867**	**0.04447**	**Accepted H0 / Failed to**

****

****

INSTABRIGHT e-GAZETTE

ISSN: 2704-3010

Volume VI, Issue IV

May 2025

Available online at https://www.instabrightgazette.com

****

traditional coding for future development tasks.														
Q2. The low-code platform allowed me to quickly understand how to build applications co mpared with complicated high-code development.	15	36	9	5	**3.72**	**1.19**	16	29	15	5	**3.55**	**1.30**	**1.97867**	**0.77767**	**Accepted H0 / Failed to Accept H1**
Q3 I faced fewer challenges learning the low-code platform than learning traditional coding.	12	37	10	6	**3.60**	**1.22**	16	27	15	7	**3.46**	**1.47**	**1.97867**	**0.59085**	**Accepted H0 / Failed to Accept H1**
Q4. I found it easy to find support and resources for learning traditional coding compar ed with low-code.	10	36	14	5	**3.49**	**1.21**	16	27	16	6	**3.48**	**1.35**	**1.97867**	**0.04447**	**Accepted H0 / Failed to Accept H1**

****

****

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Table 4

Comparison Between the two groups response to the Effectiveness of Low-Code vs. High-Code in terms of Learning Curves

SD- Standard Deviation **note: n1=65 n2=65**

df=130-2=128

LOW-CODE GROUP		HIGH CODE GROUP		Critical Value	Computed T-Value	Remarks
Mean	**SD**	**Mean**	**SD**			
3.49	1.25	3.48	1.19	1.97867	0.04447	Accepted H0 / Failed to Accept H1
3.72	1.19	3.55	1.30	1.97867	0.77767	Accepted H0 / Failed to Accept H1
3.60	1.22	3.46	1.47	1.97867	0.59085	Accepted H0 / Failed to Accept H1
3.49	1.21	3.48	1.35	1.97867	0.04447	Accepted H0 / Failed to Accept H1

The study compared the perceptions of low-code groups and high-code groups based on learning code. In terms of user confidence in future development (Q1), the t-value of 0.04447 is less than the critical value of 1.97867, indicating no significant difference, which

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

indicates that low-code supports user confidence for future development with a slight edge toward high-code. For the Quick Understanding in Building Applications (Q2), the t-value (0.77767) is less than the critical value, indicating that both respondents agree that low-code helps users quickly understand how to build applications. For the Ease of Learning (Q3), the t-value (0.59085) is less than the critical value, indicating no significant difference, which indicates that both respondents agree that low-code platforms are easier to learn than high-code platforms. In terms of Availability of Learning Resources (Q4), the t-value (0.04447) is less than the critical value, so again, no significant difference, which indicates that although both groups see little difference, the results slightly favor high-code in terms of availability of support and learning resources. The overall findings show that both groups agree on the effectiveness of low-code platforms in terms of improving user confidence, ease of learning, and quick understanding of application development, although Q4 indicates a slight preference for high-code regarding the availability of support and learning materials.

Table 5

Summary Effectiveness of Low-Code vs. Traditional High-Code of BSIS Third Year Students

Software Development in terms of Code Quality

SD- Standard Deviation

note: n1=65

n2=65

df=130-2=128

INSTABRIGHT e-GAZETTE

ISSN: 2704-3010

Volume VI, Issue IV

May 2025

Available online at https://www.instabrightgazette.com

****

ITEM	LOW-CODE GROUP						HIGH-CODE GROUP						Critical Value	Computed-T	Remarks
	5	4	2	1	WEIGHTED MEAN	SD	5	4	2	1	WEIGHTED MEAN	SD			
Q1. The code I wrote using traditional coding was clean and well-structured compared with low-code	13	30	16	6	**3.43**	**1.31**	10	31	19	5	**3.34**	**1.27**	**1.97867**	**0.39769**	**Accepted H0**
Q2. The code quality produced with the low-code platform was as good as the code I wrote using traditional coding.	10	37	14	4	**3.54**	**1.17**	9	37	14	5	**3.48**	**1.20**	**1.97867**	**0.28863**	**Accepted H0**
Q3 . The code quality from the low-code platform was sufficient for completing the project but lacked the refinement of traditional coding.	16	32	12	5	**3.65**	**1.26**	15	38	8	4	**3.80**	**1.12**	**1.97867**	**0.71736**	**Accepted H0**

****

****

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Q4. Traditional coding resulted in more efficient and optimized code than low-code development.	14	33	13	5	**3.58**	**1.25**	16	36	10	3	**3.80**	**1.12**	**1.97867**	**1.05680**	**Accepted H0**

Table 6

Comparison Between the two groups response to the Effectiveness of Low-Code vs. High-Code in terms of Code Quality

SD- Standard Deviation

note: n1=65 n2=65

df=130-2=128

LOW-CODE GROUP		**HIGH-CODE GROUP**		**Critical Value**	**Computed T-Value**	**Remarks**
Mean	**SD**	**Mean**	**SD**			
3.43	1.31	3.34	1.27	1.97867	0.39769	Accepted H0 / Failed to Accept H1
3.54	1.17	3.48	1.20	1.97867	0.28863	Accepted H0 / Failed to Accept H1
3.65	1.26	3.80	1.12	1.97867	0.71736	Accepted H0 / Failed to Accept H1

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

3.58	1.25	3.80	1.12	1.97867	1.05680	Accepted H0 / Failed to Accept H1

Based on the table results, the first group of respondents (third-year BSIS students who used low-code development on their SAD project) and the second group of respondents (third-year BSIS students who used high-code development) both agree with the effectiveness of high-code compared with low-code in terms of Code Quality.

For the criteria Well-Structured and Clean Code (Q1), the t-test value 0.39769 is less than the critical value 1.97867, which makes it not a significant difference. This indicates that the respondents still perceive high-code as more effective in producing clean and well-structured code. In terms of overall code quality (Q2), the t-value of 0.28863 is less than the 1.97867 critical value, which indicates that both recognize the importance of code quality, with a slight preference for high code. For the criteria of Code Refinement (Q3), the t-test value of 0.71736 is also less than the critical value, which indicates no significant difference; both respondents agree that high-code development provides better code refinement than low code. In the criteria of Efficiency and Optimization (Q4), the t-test value of 1.05680 indicates that there is no significant difference, which means that both respondents agree that high-code is more efficient and optimized, though not to a statistically significant extent.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Table 7

Summary and Effectiveness of Low-Code vs. Traditional High-Code of BSIS Third Year Students

Software Development in terms of User Satisfaction

SD- Standard Deviation

note: n1=65
n2=65
df=130-2=128

ITEM	LOW-CODE GROUP						HIGH-CODE GROUP						Critical Value	Computed- T	Remarks
	5	4	2	1	WEIGHTED MEAN	SD	5	4	2	1	WEIGHTED MEAN	SD			
Q1. My overall satisfaction with the low-code platform was higher than with traditional coding.	12	40	9	4	**3.72**	**1.11**	14	30	17	4	**3.51**	**1.26**	**1.97867**	**1.00826**	**Accepted H0**
Q2. I had a more satisfying user experience using low-code compared to the traditional coding.	5	37	18	5	**3.29**	**1.18**	19	22	20	4	**3.49**	**1.36**	**1.97867**	**0.89553**	**Accepted H0**
Q3. The low-code platform provided a more enjoyable development experience	8	43	10	4	**3.63**	**1.08**	14	37	9	5	**3.71**	**1.18**	**1.97867**	**0.40321**	**Accepted H0**

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

compared to traditional coding.														
Q4. I felt more productive and satisfied using the low-code platform than when using traditional coding.	9	45	6	5	**3.72**	**1.07**	16	32	11	6	**3.63**	**1.28**	**1.97867**	**0.43493**	**Accepted H0**

Table 8

Comparison Between the two groups response to the Effectiveness of Low-Code vs. High-Code in terms of User Satisfaction

SD- Standard Deviation

note: n1=65 n2=65

df=130-2=128

LOW-CODE GROUP		**HIGH-CODE GROUP**		**Critical Value**	**Computed T-Value**	**Remarks**
Mean	**SD**	**Mean**	**SD**			
3.72	**1.11**	**3.51**	**1.26**	**1.97867**	**1.00826**	**Accepted H0 / Failed to Accept H1**
3.29	**1.18**	**3.49**	**1.36**	**1.97867**	**0.89553**	**Accepted H0 / Failed to Accept H1**

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

3.63	**1.08**	**3.71**	**1.18**	**1.97867**	**0.40321**	**Accepted H0 / Failed to Accept H1**
3.72	**1.07**	**3.63**	**1.28**	**1.97867**	**0.43493**	**Accepted H0 / Failed to Accept H1**

Based on the table results, the first group of respondents (third-year BSIS students who used low-code development on their SAD project) and the second group of respondents (third-year BSIS students who used high-code development) both agree with the effectiveness of low-code compared with high-code in terms of user satisfaction.

For the Effectiveness of User Satisfaction (Q1), the t-test value of 1.008 is less than the critical value, making it no significant difference, which indicates that both respondents perceive that low-code is at least as effective as high-code in delivering high user satisfaction. In terms of the criteria of Satisfying User Experience (Q2), the t-test of 0.896 is also less than the critical value of 1.979, which makes no significant value. This indicates that both respondents are leaning toward low-code outperforming high-code in user experience satisfaction. For the Enjoyable Learning Experience, the t-value of 0.403 is less than the critical t-value, making it no significant difference and indicating that low-code is seen as equally enjoyable as high-code for the development experience. Lastly, for the criteria of Productivity & Satisfaction (Q4), the t-test value of 0.435 makes no significant difference since the t-value is less than the critical value of 1.979. This means that both agree that there is no measurable

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

downside to using low-code in terms of productivity and satisfaction. The overall findings indicate that across all criteria—effectiveness, experience, enjoyment, and productivity—students believe low-code development is at least as good as, and often slightly better than, high-code development when it comes to user satisfaction.

Table 9

Effectiveness of Low-Code vs. Traditional High-Code of BSIS Third Year Students Software Development in terms of Core Programming Skills

ITEM	LOW-CODE GROUP		HIGH-CODE GROUP	
	WEIGHTED MEAN	SD	WEIGHTED MEAN	SD
Q1. High-code development helped me significantly strengthen my programming skills compared with low-code platforms.	**3.75**	**1.09**	**3.57**	**1.31**
Q2. I gained a deeper understanding of algorithms and logic using a high-code platform compared with a low-code platform.	**3.51**	**1.15**	**3.54**	**1.36**
Q3. I improved my ability to work with APIs and external systems during the project.	**3.85**	**1.02**	**3.74**	**1.20**
Q4. High-Code development enhanced	**3.78**	**1.02**	**3.63**	**1.28**

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

my knowledge of databases and data management compare with low-code.				

Table 10

Comparison Between the two groups response to the Effectiveness of Low-Code vs. High-Code in terms of Core Programming Skills

SD- Standard Deviation

note: n1=65
n2=65
df=130-2=128

LOW-CODE GROUP		HIGH-CODE GROUP		Critical Value	Computed T-Value	Remarks
Mean	**SD**	**Mean**	**SD**			
3.75	1.09	3.57	1.31	1.97867	0.86118	Accepted H0 / Failed to Accept H1
3.51	1.15	3.54	1.36	1.97867	0.13580	Accepted H0 / Failed to Accept H1
3.85	1.02	3.74	1.20	1.97867	0.56310	Accepted H0 / Failed to Accept H1
3.78	1.02	3.63	1.28	1.97867	0.73889	Accepted H0 / Failed to Accept H1

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Based on the table results, the first group of respondents (third-year BSIS students who used low-code development on their SAD project) and the second group of respondents (third-year BSIS students who used high-code development) both agree with the effectiveness of high-code compared with low-code in terms of core programming skills.

Q1/Core Programming Skills gathered a weighted mean of 3.75 for the BSIS students who used low-code development, while the second respondents, composed of third-year BSIS students who used high-code, got a mean value of **3.57**. The computed t-value equivalent to 0.86118 is less than the critical value equivalent to 1.97867. The result indicates that there is no significant difference between the responses of the two groups regarding the effectiveness of high-code in terms of strengthen the user development skills compared to low code. This suggests that respondents believe high-code development is more effective in strengthening the user development skills compared to low-code development.

The Q2/Core Programming skills stated that the first group of respondents who used low-code development had a weighted mean of **3.51,** while the second group of respondents, composed of third-year BSIS students who used high-code, got a mean value of **3.54**. The computed t-value equivalent to 0.13580 is less than the critical value equivalent to 1.97867. This finding indicates that there is no significant difference between the responses of the two groups regarding the effectiveness of high-code development in terms of deeper understanding of algorithms and logic compared to low-code development. This result reflects

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

the respondent's belief that high-code development outperforms low-code development in terms of deeper understanding algorithm and logic.

Q3/Core Programming skills results show a weighted mean of 3.85 for the first respondent (low-code development) and a weighted mean of 3.74 for the second respondent (high-code development). The computed t-value equivalent to 0.56310 is less than the critical value equivalent to 1.97867. The result indicates that respondents agree with the improved ability to work with APIs and external systems during the project of high-code development compared with low-code.

Q4/Core Programming skills gathered a weighted mean of 3.78 for the BSIS students who used low-code development, while the second respondents, composed of third-year BSIS students who used high-code, got a mean value of 3.63. The computed t-value equivalent to 0.73889 is less than the critical value equivalent to 1.97867. This finding indicates that there is no significant difference between the responses of the two groups regarding the effectiveness of high-code to enhance learner knowledge in databases and data management compared to low-code development.

Summary of Findings, Results and Conclusion

This study focuses on comparing the perception of third-year students who used low-code and traditional high-code development in their SAD (System Analysis and Design) course and were required to complete a capstone project. The comparison directly focuses on the

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

five aspects: development speed, learning curve, code quality, user satisfaction, and core programming skills. The t-test is used for statistical treatment and employed to determine the significant differences between two groups of respondents. With all questions, there were no significant differences found, as all computed t-values were less than the critical value of 1.97867, indicating they are statistically insignificant.

In development speed, low-code and high-code groups had similar perceptions across all four indicators. Q1-Q3 both groups agreed that low code is more effective in speeding up the development, debugging, testing, and accelerating the delivery, while Q4 has a slight preference toward high code in reducing creation and deploying time, although not significant.

In the Learning Curve, across all four questions, the low code was seen as favorable for quicker understanding, ease of learning, and user confidence. The Q4 slightly favored the high code regarding the ease of finding learning resources, but again, the difference was insignificant.

With code quality, all questions revealed a consistent perception that high-code development is more effective in producing well-structured, optimized, and high-quality code. The two groups of respondents agreed that high-code outperforms low-code in terms of code refinement and quality, despite no statistical significance.

In terms of user satisfaction, the response of two groups was in favor of low-code in providing a more satisfying, enjoyable development experience; however, one item, Q2, had a higher mean for high-code, indicating some students found high-code more satisfying, though the t-value shows no significant difference.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

Lastly, in core programming skills, all indicators are slightly in favor of high-code, especially in strengthening the development skills, understanding logic, and working with APIs and databases. The findings believe that high-code provides better grounding in technical skills.

Both low-code and high-code development methods were evaluated accordingly, and there is no significant statistical difference was found in any of the measured aspects. However, trends in responses reveal that in low-code it is more effective in terms of development speed, learning curve, and user satisfaction, while high-code is more effective in terms of code quality and core programming skills.

These results suggest that low-code tools may be more suited for rapid development and ease of use, where high-code approaches remain essential for mastering deeper programming competencies. The evidence suggests that both approaches are acceptable, but the choice should depend on the capstone project's nature. If the project focuses on the business system, prototypes, or applications emphasizing usability, speed, and client interaction, then low-code is the suggested approach, but if the capstone project focuses on technically complex systems requiring custom features, deep back-end integration, or algorithmic logic, then the capstone project should stick with high-code as the suggested approach.

INSTABRIGHT e-GAZETTE
ISSN: 2704-3010
Volume VI, Issue IV
May 2025
Available online at https://www.instabrightgazette.com

****

References:

Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021). Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective.

Yan, Z. (2021). The Impacts of Low/No-Code Development on Digital Transformation and Software Development.

Heuschkel, S. (2023). The Impact of No-Code on Digital Product Development.

Tornhill, A., & Borg, M. (2022). Code Red: The Business Impact of Code Quality – A Quantitative Study of 39 Proprietary Production Codebases.

"Traditional vs. low-code development: comparing needed effort and system complexity in the NexusBRaNT experiment" (IEEE CBI, June 2023)

"Metacognitive skills in low-code app development: Work-integrated learning in information systems development" (SAGE Journals, 2024)

****

****