Institute of Information Technology

University of Dhaka

Topic: Goal Question Metrics(GQM)

Software Metrics (SE-611)

To Evaluate Increasing Student Comprehension and Usage of Software Engineering Best Practices in Project-Based Courses

Submitted by:
Group

Md Jihad Hossain - 1413
Md Mainuddin- 1472

Submitted to:

Dr. Emon Kumar Dey
Associate Professor
Institute of Information Technology
University of Dhaka

Table of Contents:

1. Introduction

GQM, an acronym for "Goal, Question, Metric," is a well-established goal-oriented approach for defining and using software metrics to improve and measure software quality.

GQM defines measurements across three dimensions:

1. Conceptual Level (Goal):
   A goal is established for a specific object, for various purposes, considering different models of quality, multiple perspectives, and a specific context or environment.

2. Operational Level (Question):
   A set of questions is formulated to define models related to the object of study. These questions help focus on the object to assess or achieve the specified goal.

3. Quantitative Level (Metric):
   A series of metrics, derived from the models, is associated with each question to provide measurable answers.

The process begins by defining the goal. From there, a relevant question set is developed to collect data from stakeholders or target groups. These questions must be linked to well-defined metrics to evaluate results quantitatively. Finally, a questionnaire is prepared to gather data from the defined scope, followed by metric analysis and result interpretation, leading to actionable outcomes.

2. Project Specification

2.1 Project Overview

The objective is to assess and enhance student understanding and application of software engineering best practices in project-based courses. This involves identifying their current level of comprehension, analyzing their practical usage of best practices, and providing actionable insights to improve their skills. Students from various educational backgrounds and levels will contribute by sharing their experiences and challenges with implementing these practices in their projects.

2.2 Motivation

Software engineering best practices, such as version control, code reviews, unit testing, and documentation, are essential for developing reliable, maintainable, and scalable software. However, these practices are often underutilized or misunderstood in academic project settings. With the increasing emphasis on project-based learning, it is crucial to equip students with the knowledge and habits to apply these practices effectively. By addressing gaps in understanding and usage, we can foster a generation of software developers who are better prepared for industry challenges.

2.3 Scope

This study targets students at the undergraduate levels from diverse curriculums at University Of Dhaka . Through surveys and practical assessments, we aim to learn about their familiarity with software engineering best practices, the extent of their application in project-based courses, and the obstacles they face. The collected data will guide the design of interventions and resources to improve student outcomes in software engineering education.

3. Goal Specification

3.1 GQM Framework

General Statement: Investigating Student Awareness and Application of Software Engineering Best Practices in Project-Based Courses: An analysis.

3.2 PPE Approach

- **Purpose:**
  To evaluate the understanding and usage of software engineering best practices among students in project-based courses, in order to determine the effectiveness of current teaching methods and identify areas for improvement.

- **Perspective**:
  To assess the level of comprehension, practical application, and perceived importance of software engineering best practices from the students' viewpoint, with a focus on identifying gaps and challenges in their learning process.

- **Environment:**
  In project-based courses, students often lack the exposure or training necessary to fully implement software engineering best practices, leading to challenges in producing maintainable, scalable, and reliable software. By exploring the experiences of students across different study levels and disciplines, we aim to understand how well these practices are being adopted and identify ways to enhance their usage.

Final Goal:
"To assess the comprehension and application of software engineering best practices among students in project-based courses, in order to identify challenges and propose effective strategies for improvement from the viewpoint of students."

3.3 Sub Goals

Our goal has 5 sub-goals:

A. Evaluate whether students know that software engineering best practices are essential for the success of project-based courses. This goal can be achieved by asking students questions about their awareness of best practices, such as whether they know the importance of version control, unit testing, code reviews, and documentation, and whether they understand how these practices contribute to producing reliable and maintainable software.

B. Determine how responsibly students apply software engineering best practices in their projects. This goal can be achieved by asking students questions about their usage of best practices, such as whether they follow version control, conduct code reviews, write unit tests, and document their code.

C. Understand students' perceptions of the value and effectiveness of software engineering best practices in project-based courses. This goal can be achieved by asking students open-ended questions about their experiences with best practices, such as whether they believe practices like unit testing and code reviews improve software quality, and whether they see the value of these practices in their academic projects. We can also ask about their level of study (undergraduate, graduate) and their major to understand if perceptions differ across disciplines.

D. Identify the factors that influence students' perceptions and adoption of software engineering best practices. This could be done by asking students about their experiences with best practices, such as the challenges they face in implementing them, their motivation for using them, and any external influences (e.g., peer pressure, course requirements).

E. Investigate the potential barriers or misconceptions preventing students from using software engineering best practices. This could be done by asking students about their concerns or frustrations related to best practices, such as whether they find them time-consuming, difficult to implement, or irrelevant to their specific projects.

3.4 Question and Metrics

Subgoal A: Evaluate whether students know that software engineering best practices are essential for the success of project-based courses

Q1: Are students aware of the importance of software engineering best practices for project success?

- M1: Percentage of surveyed students who know the importance of best practices such as version control, unit testing, code reviews, and documentation.
- M2: Percentage of students who can accurately describe the role of these best practices in producing reliable software.
- M3: Measure from course feedback regarding the importance of best practices in academic projects.

Q2: How often do students apply software engineering best practices in their projects?

- M1: Percentage of students who use version control, unit testing, code reviews, and documentation in their projects.
- M2: Frequency of best practices usage (e.g., daily, weekly, monthly).

Subgoal B: Determine how responsibly students apply software engineering best practices

Q1: How frequently do students perform code reviews and write unit tests in their projects?

- M1: Percentage of students who conduct regular code reviews and write unit tests in their projects.
- M2: Frequency of code reviews and unit testing among students who engage in the practice.

Q2: Do students document their code and follow best practices like version control?

- M1: Percentage of students who use version control systems like Git and document their code.
- M2: Percentage of students who follow a standardized approach to documenting their code.

Subgoal C: Understand students' perceptions of the value and effectiveness of software engineering best practices

Q1: Do students believe that software engineering best practices improve the quality of their projects?

- M1: Percentage of students who believe that best practices such as unit testing, code reviews, and version control improve project quality.
- M2: Students' perceptions of the importance of best practices for creating reliable and maintainable software.

Q2: Do students from different academic levels or disciplines have different perceptions of software engineering best practices?

- M1: Comparison of perceptions regarding best practices across undergraduate students.
- M2: Comparison of perceptions across different departments or study programs.

Subgoal D: Identify the factors that influence students' perceptions and adoption of software engineering best practices

Q1: What motivates students to adopt software engineering best practices in their projects?

- M1: Percentage of students who adopt best practices due to course requirements, peer influence, or personal interest.
- M2: Reasons given by students for using or not using best practices in their projects.

Q2: What challenges do students face in applying software engineering best practices in their projects?

- M1: Percentage of students who report challenges like lack of time, complexity, or understanding of the practices.
- M2: Types of challenges students face when implementing best practices in their projects.

Subgoal E: Investigate the potential barriers or misconceptions preventing students from using software engineering best practices

Q1: What are the common misconceptions or barriers preventing students from using best practices?

- M1: Instances where students believe best practices are unnecessary or too

time-consuming.
- M2: Percentage of students who feel that best practices are not relevant to their specific projects.

Q2: How can students be encouraged to apply best practices more effectively?

- M1: Percentage of students who express a willingness to adopt best practices if provided with more guidance or resources.
- M2: Students' suggestions for improving the teaching or application of software engineering best practices in project-based courses.

4. Questionnaire Preparation and Data Collection

After defining the **Goal, Questions, and Metrics,** we prepared a survey questionnaire aimed at evaluating **student comprehension and usage of software engineering best practices in project-based courses.**

To analyze **"Increasing Student Comprehension and Usage of Software Engineering Best Practices in Project-Based Courses,"** we undertook a systematic investigation into students' understanding, application, and perceptions regarding software engineering best practices. The primary objective was to assess their familiarity, implementation methods, and the challenges faced in adopting these practices.

To facilitate this analysis, we carefully designed a survey questionnaire covering diverse aspects of software engineering best practices. Below is a breakdown of the key areas addressed:

General Awareness of Software Engineering Best Practices

We began by assessing respondents' awareness of key software engineering best practices. This foundational understanding provided critical context for analyzing their responses to more specific questions. Topics explored included awareness of principles like **modular design**, **version control**, **automated testing**, and **code reviews.**

Practical Application in Project-Based Courses

Respondents were asked how frequently and effectively they applied these practices in their projects. We also explored the types of projects (individual or group-based) in which they engaged, along with their adherence to best practices during different stages, such as planning, development, and testing.

Challenges and Barriers

We collected data on obstacles preventing students from effectively applying best practices. Questions targeted both internal barriers (e.g., lack of knowledge or confidence) and external barriers (e.g., insufficient resources or unclear guidance from instructors).

Perceived Benefits and Effectiveness

Questions in this section aimed to understand students' perceptions of the value and outcomes of using best practices, such as improved code quality, reduced errors, and enhanced team collaboration.

Demographic Information

Basic demographic data, such as academic level, field of study, and institution type, were collected to analyze trends across different student groups.

By gathering comprehensive data, the survey aimed to provide a nuanced understanding of how students perceive and utilize software engineering best practices, as well as the challenges and benefits they encounter in project-based courses.

Sample Question:

Q: How frequently do you conduct code reviews for your academic projects?

1. Always
2. Frequently
3. Occasionally
4. Rarely
5. Never

Full Questionnaire

Relating to Student Background:

1. Have you ever been formally introduced to software engineering best practices (e.g., code versioning, testing, documentation)?
2. What is your experience level in software development?

Relating to the application of software engineering best practices in project-based courses:

1. Which software engineering practices do you follow in your projects?
2. How often do you refer to guidelines or resources for best practices during project development?
3. What challenges do you face in applying software engineering best practices?
4. What tools or resources have you used to learn software engineering best practices?
5. Do you believe that following best practices improves project outcomes?
6. How do you ensure the quality of your code or project deliverables?
7. What suggestions do you have for improving student comprehension and application of software engineering best practices in project-based courses?

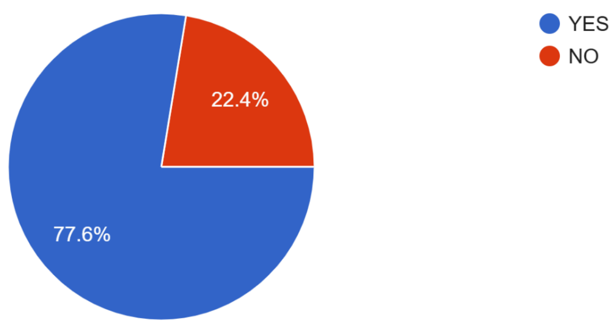
5. Data Visualization

In this section, the summary of the collected data is represented with pie charts

Q1.

Have you ever been formally introduced to software engineering best practices (e.g., code versioning, testing, documentation)?

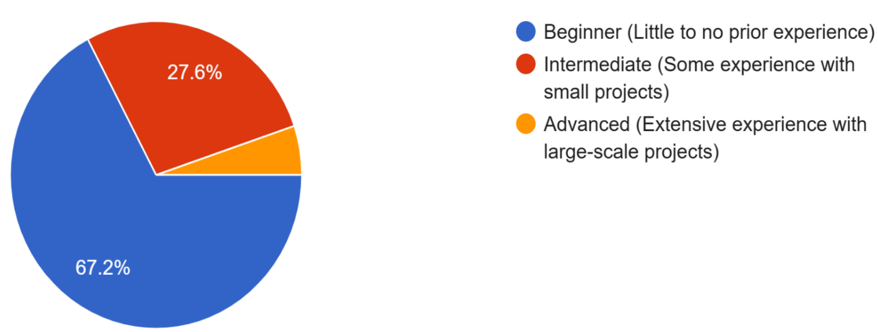
58 responses



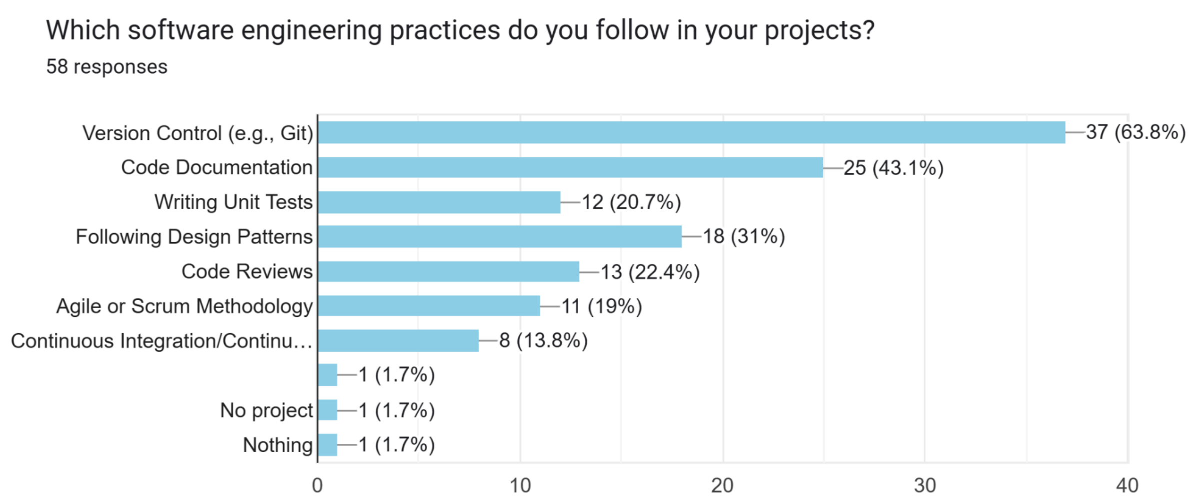
Q2.

What is your experience level in software development?

58 responses

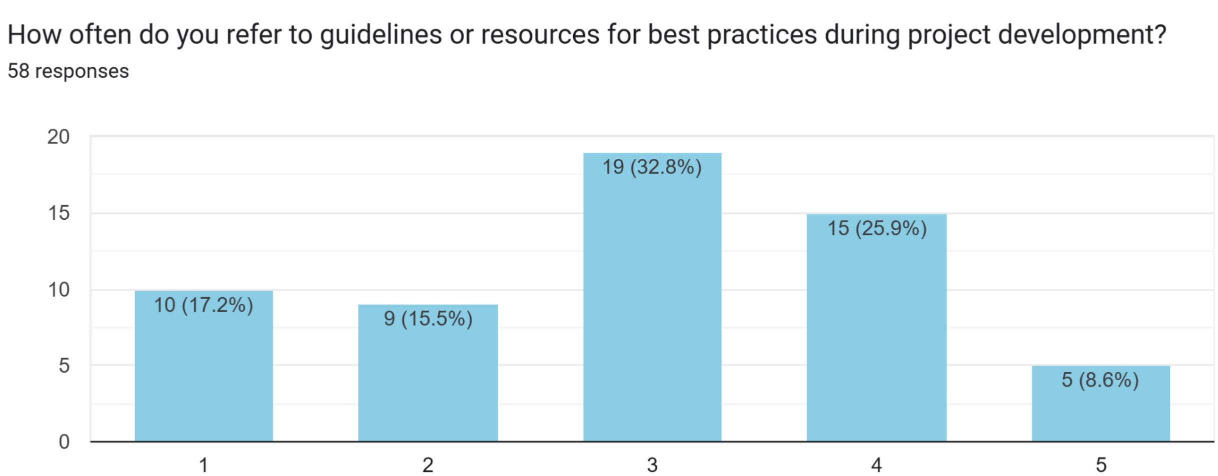


Q3.



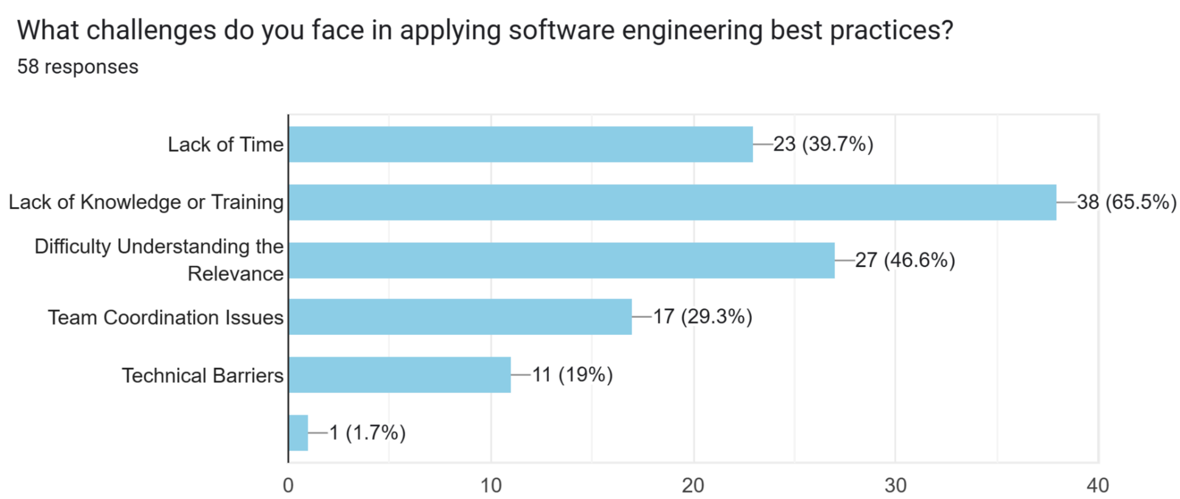
Which software engineering practices do you follow in your projects?
58 responses
Version Control (e.g., Git)
Code Documentation
Writing Unit Tests
Following Design Patterns
Code Reviews
Agile or Scrum Methodology
Continuous Integration/Continu…
No project
Nothing
37 (63.8%)
25 (43.1%)
12 (20.7%)
18 (31%)
13 (22.4%)
11 (19%)
8 (13.8%)
1 (1.7%)
1 (1.7%)
1 (1.7%)
0
10
20
30
40

Q4.



How often do you refer to guidelines or resources for best practices during project development?
58 responses
20
15
10
5
0
10 (17.2%)
9 (15.5%)
19 (32.8%)
15 (25.9%)
5 (8.6%)
1
2
3
4
5

Q5.

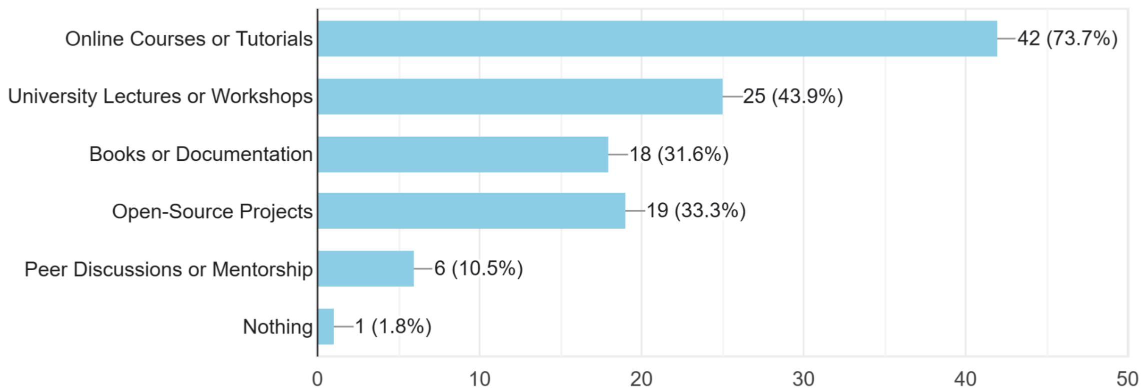


What challenges do you face in applying software engineering best practices?
58 responses
Lack of Time
Lack of Knowledge or Training
Difficulty Understanding the Relevance
Team Coordination Issues
Technical Barriers
23 (39.7%)
38 (65.5%)
27 (46.6%)
17 (29.3%)
11 (19%)
1 (1.7%)
0
10
20
30
40

Q6.

What tools or resources have you used to learn software engineering best practices?

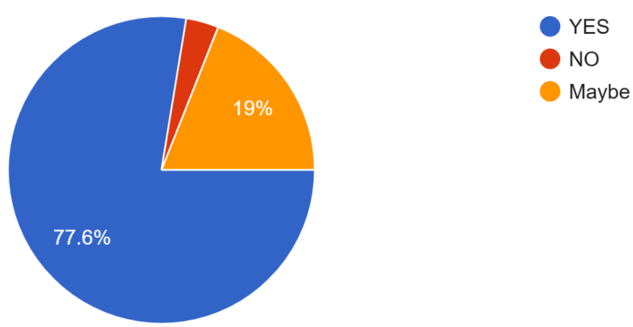
57 responses



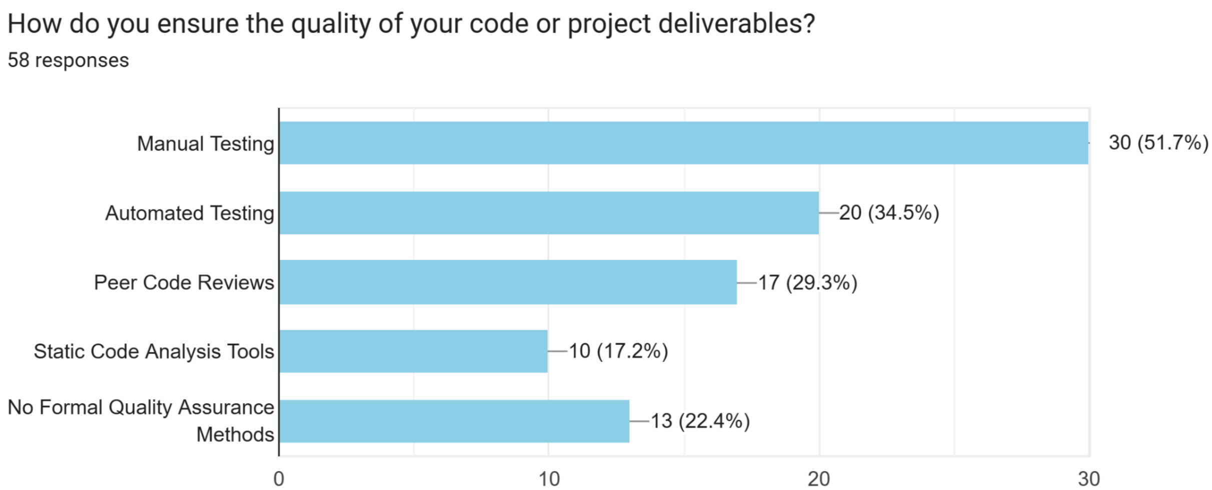
Q7.

Do you believe that following best practices improves project outcomes?

58 responses



Q8.



6.MetricsAnalysis

Q1. Have you ever been formally introduced to software engineering best practices (Yes/No)

`=chitest(E3:E4, F3:F4)`

	A	B	C	D	E	F
1	What is your experience level in software development?					
2					Observed	Expected
3	Yes	45			45	29
4	No	13			13	29
5				**P-value:**	0.00002647941857	

Hypotheses:

- **Null Hypothesis (H_0)**: The distribution of responses (Yes, No) matches the expected distribution.
- **Alternative Hypothesis(H_a)**: The distribution of responses (Yes, No) does not match the expected distribution.

Chi-Square Formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

- O_i = Observed value
- E_i = Expected value

Calculation:

$$\chi^2 = \frac{(45-29)^2}{29} + \frac{(13-29)^2}{29}$$

$$\chi^2 = \frac{(16)^2}{29} + \frac{(-16)^2}{29}$$

$$\chi^2 = \frac{256}{29} + \frac{256}{29}$$

$$\chi^2 = 8.83 + 8.83 = 17.66$$

Degrees of Freedom:

$$\text{df} = \text{Number of categories} - 1 = 2 - 1 = 1$$

P-Value:

Using a chi-square table or a statistical tool, for $\chi^2 = 17.66$ with $df = 1$, the p-value is:

$$P = 0.00002647941857$$

Significance Level (α):

$\alpha = 0.05$

Conclusion:

Since $P = 0.00002647941857 < 0.05$, we reject the null hypothesis (H_0). This indicates that the observed distribution of responses (Yes, No) is significantly different from the expected distribution.

Q2 . What is your experience level in software development?

`=chitest(E3:E4:E5, F3:F4:F5)`

A	B	C	D	E	F
What is your experience level in software development?					
				Observed	Expected
Advanced	3			3	19.33
Intermediate	16			16	19.33
Beginner	39			39	19.33
			P-value:	0.000000003414600369	

Hypotheses:

Null Hypothesis (H_0): The distribution of experience levels (Advanced, Intermediate, Beginner) matches the expected distribution.

Alternative Hypothesis(H_a): The distribution of experience levels (Advanced, Intermediate, Beginner) does not match the expected distribution.

Chi-Square Formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

- O_i = Observed value
- E_i = Expected value

Calculation:

$$\chi^2 = \frac{(3 - 19.33)^2}{19.33} + \frac{(16 - 19.33)^2}{19.33} + \frac{(39 - 19.33)^2}{19.33}$$

$$\chi^2 = \frac{(-16.33)^2}{19.33} + \frac{(-3.33)^2}{19.33} + \frac{(19.67)^2}{19.33}$$

$$\chi^2 = \frac{266.89}{19.33} + \frac{11.09}{19.33} + \frac{386.77}{19.33}$$

$$\chi^2 = 13.81 + 0.57 + 20.00 = 34.38$$

Degrees of Freedom:

$$\mathrm{df} = \text{Number of categories} - 1 = 3 - 1 = 2$$

P-Value:

Using a chi-square table or a statistical tool, for $\chi^2 = 34.38$ with $df = 2$, the p-value is:

$$P = 0.00000003414600369$$

Significance Level (α):

$\alpha = 0.05$

Conclusion:

Since $P = 0.00000003414600369 < 0.05$, we reject the null hypothesis (H_0). This means the observed distribution of experience levels is significantly different from the expected distribution.

Q3. Which software engineering practices do you follow in your projects?

E11 | f_x =CHITEST(E3:E10,F3:F10)

	A	B	C	D	E	F
1	Which software engineering practices do you follow in your projects?					
2					Observed	Expected
3	Version Control	30			30	11.625
4	Code Documentation	15			15	11.625
5	Writing Unit Tests	12			12	11.625
6	Following Design Pat	10			10	11.625
7	Code Reviews	8			8	11.625
8	Agile or Scrum Methc	7			7	11.625
9	CI/CD	6			6	11.625
10	No Project/Nothing	5			5	11.625
11				**P value:**	0.000001417030	

Hypotheses:

Null Hypothesis (H_0): The software engineering practices are uniformly distributed (all categories are equally likely).

Alternative Hypothesis (H_a): The software engineering practices are not uniformly distributed (some practices are followed more frequently).

Expected Frequencies:

$$E = \frac{\text{Total Observations}}{\text{Number of Categories}} = \frac{93}{8} = 11.625$$

The expected frequency for each category is 11.625.

Chi-Square Test Formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

- O_i: Observed frequency
- E_i: Expected frequency

Calculations:

Practice	O_i	E_i	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
Version Control (e.g., Git)	30	11.625	18.375	337.6406	29.0470
Code Documentation	15	11.625	3.375	11.3906	0.9801
Writing Unit Tests	12	11.625	0.375	0.1406	0.0121
Following Design Patterns	10	11.625	-1.625	2.6406	0.2272
Code Reviews	8	11.625	-3.625	13.1406	1.1301
Agile or Scrum Methodology	7	11.625	-4.625	21.3906	1.8405
Continuous Integration/Deployment	6	11.625	-5.625	31.6406	2.7219
No Project/Nothing	5	11.625	-6.625	43.8906	3.7770
Total					**39.7359**

$$\chi^2 = 39.7359$$

Degrees of Freedom:

df = Number of Categories - 1 = 8 - 1 = 7

p-Value:

Using a Chi-Square distribution table or calculator with:

- χ2 =39.7359
- df=7

The p-value is approximately **0.0000014**.

Conclusion:

- **Significance Level (α)**: 0.05
- **p-value (0.0000014)** < α(0.05)

We reject the null hypothesis that software engineering practices are uniformly distributed. The data shows significant variation in the frequency of different practices, indicating that some are followed more frequently than others.

Q4. How often do you refer to guidelines or resources for best practices during project development?

H8 fx =CHITEST(F3:F4,G3:G4)

	A	B	C	D	E	F	G
1	Q4. How often do you refer to guidelines or resources for best practices during project development?						
2						Observerd	Expected
3	1 *	10		rarely prefer (1*, 2*, 3*)		38	29
4	2*	9		Frequently prefers (4*, 5*)		20	29
5	3*	19					
6	4*	15					
7	5*	5					
8					P value:	0.01810245515	
9							
10							

Hypotheses:

Null Hypothesis (H_0): The preference for referring to guidelines or resources is used frequently.

Alternative Hypothesis(H_a): The preference for referring to guidelines or resources is rarely used.

Calculations:

Group	Observed (O)	Expected (E)	$O - E$	$(O - E)^2$	$\frac{(O-E)^2}{E}$
Rarely Prefer	38	29	9	81	2.7931
Frequently Prefer	20	29	-9	81	2.7931
Total	**58**	**58**			**5.5862**

$$\chi^2 = 5.5862$$

Degrees of Freedom:

$$\text{df} = \text{Number of Groups} - 1 = 2 - 1 = 1$$

---

p-Value:

Using a Chi-Square distribution table or calculator with:

- $\chi^2 = 5.5862$
- $\text{df} = 1$

The p-value is approximately **0.0181**.

Conclusion:

- **Significance Level (α)**: 0.05
- **p-value (0.0181)** < $\alpha(0.05)$

We **reject the null hypothesis (H_0)**.

Q5. What challenges do you face in applying software engineering best practices?

A	B	C	D	E	F	G
Q5. What challenges do you face in applying software engineering best practices?						
					Observerd	Expected
lack of time	23		Knowledge and Training Issues		65	39
Lack of Knowledge or Training	38		Time and Resource Constraints		34	39
Difficulty Understanding the Relevance	27		Team and Organizational Issues		18	39
Team Coordination Issues	17					
Technical Barriers	11					
Other	1			P value:	0.000000438009	

Hypotheses:

Null Hypothesis (H_0): The number of students experiencing each type of challenge is equal, and there are no significant differences between the groups.

Alternative Hypothesis(H_a): There are significant differences in how students experience the challenges, i.e., one or more of the groups (Knowledge and Training Issues, Time and Resource Constraints, Team and Organizational Issues) may have a significantly higher or lower number of students facing those challenges.

Group	Observed (O)	Expected (E)	(O - E)	$(O - E)^2$	$(O - E)^2 / E$
Knowledge and Training Issues	65	39	26	676	17.3333
Time and Resource Constraints	34	39	-5	25	0.6410
Team and Organizational Issues	18	39	-21	441	11.3077

Chi-Square Value:

$$\chi^2 = 17.3333 + 0.6410 + 11.3077 = 29.2820$$

Degrees of Freedom:

$$df = \text{Number of groups} - 1 = 3 - 1 = 2$$

For **$\chi 2$=29.2820 and df=2, the p-value is approximately 0.000000438009**

As p-value< α(0.05)

We can reject the null hypothesis**.**

Q6.What tools or resources have you used to learn software engineering best practices?

fx =chitest(D3:D8,E3:E8)

A	B	C	D	E
What tools or resources have you used to learn software engineering best practices?				
			Observed	Expected
Online Courses or Tutorials	37		37	15.833
University Lectures or Worksh	26		26	15.833
Books or Documentation	13		13	15.833
Open-Source Projects	8		8	15.833
Peer Discussions or Mentorsh	6		6	15.833
Nothing	5		5	15.833
		P value:	0.0000000003825	

Hypotheses:

- **Null Hypothesis (H_0):** The tools or resources used to learn software engineering best practices are uniformly distributed (i.e., all resources are equally likely to be used).
- **Alternative Hypothesis (H_a):** The tools or resources used to learn software engineering best practices are not uniformly distributed (i.e., some resources are used more frequently than others).

Expected Frequencies:

- **Total Observations** = 37 + 26 + 13 + 8 + 6 + 5 = **95**
- **Number of Categories** = 6 (Online Courses, University Lectures, Books, Open-Source Projects, Peer Discussions, Nothing)
- **Expected Frequency** for each category = Total Observations / Number of Categories
  E=95/6 = 15.833

Chi-Square Test Formula:

The Chi-Square statistic is calculated using the following formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

- O_i = Observed frequency for category i
- E_i = Expected frequency for category i

Calculation:

Now, we calculate the Chi-Square value for each category:

Category	Observed (O_i)	Expected (E_i)	$\frac{(O_i - E_i)^2}{E_i}$
Online Courses or Tutorials	37	15.8333	$\frac{(37-15.8333)^2}{15.8333} = 28.2924$
University Lectures or Workshops	26	15.8333	$\frac{(26-15.8333)^2}{15.8333} = 6.5263$
Books or Documentation	13	15.8333	$\frac{(13-15.8333)^2}{15.8333} = 0.5069$
Open-Source Projects	8	15.8333	$\frac{(8-15.8333)^2}{15.8333} = 3.8690$
Peer Discussions or Mentorship	6	15.8333	$\frac{(6-15.8333)^2}{15.8333} = 6.1057$
Nothing	5	15.8333	$\frac{(5-15.8333)^2}{15.8333} = 7.4089$

Total Chi-Square Value = 28.2924 + 6.5263 + 0.5069 + 3.8690 + 6.1057 + 7.4089 = **52.708**

Degrees of Freedom (df):

Degrees of freedom is calculated as:

df = Number of Categories - 1 = 6 -1 = 5

P-Value:

Using a Chi-Square distribution table or calculator with a $\chi2$ = 52.708 and df = 5, the **p-value** is:

p-value = 0.0000000003825835204

Conclusion:

Since the p-value (0.0000000003825835204) is **less than 0.05**, **we reject the null hypothesis (H_0)**.

This means that the distribution of tools or resources used to learn software engineering best practices is **not uniform**, and some resources (like Online Courses and University Lectures) are used more frequently than others.

Q7.Do you believe that following best practices improves project outcomes?

fx =chitest(F3:F4,G3:G4)

A	B	C	D	E	F	G	H
Q7.Do you believe that following best practices improves project outcomes?							
					Observerd	Expected	
YES			Believe		45	19.33	
NO			Deny		2	19.33	
MAYBE			Not sure about		11	19.33	

Hypotheses:

Null Hypothesis (H_0): Students do not believe that these best practices improve outcomes.

Alternative Hypothesis(H_a): Students believe that these best practices improve outcomes.

Calculations

For each category:

$$\frac{(O-E)^2}{E}$$

Response Category	Observed (O)	Expected (E)	$O-E$	$(O-E)^2$	$\frac{(O-E)^2}{E}$
Believe	45	19.33	25.67	659.51	34.11
Deny	2	19.33	-17.33	300.30	15.54
Not Sure About	11	19.33	-8.33	69.38	3.59
Total	**58**	**58**			**53.24**

$$\chi^2 = 34.11 + 15.54 + 3.59 = 53.24$$

Degrees of Freedom

$$df = \text{Number of Categories} - 1 = 3 - 1 = 2$$

---

p-Value

Using a Chi-Square distribution table or calculator with:

- $\chi^2 = 53.24$
- $df = 2$

The p-value is approximately **0.0000001**.

Conclusion

- **Significance Level** (α): 0.05
- **p-value (0.0000001)** < $\alpha(0.05)$

We **reject the null hypothesis** (H_0).

Q8. How do you ensure the quality of your code or project deliverables?

A	B	C	D	E	F	G	H
Q8.How do you ensure the quality of your code or project deliverables?							
					Observerd	Expected	
Manual Testing	30		Manual Quality Assurance Methods		47	30	
Automated testing	20		Automated Quality Assurance Methods		30	30	
Peer code reviews	17		No Formal Quality Assurance		13	30	
Static code analysis	10						
No formal method	13						
				P value:			0.000065508324

Hypotheses:

Null Hypothesis (H_0): Students prefer automated testing.
Alternative Hypothesis(H_a): Students prefer manual testing.

Group	Observed (O)	Expected (E)	(O - E)	$(O - E)^2$	$(O - E)^2 / E$
Manual QA Methods (Group 1)	47	30	17	289	9.63
Automated QA Methods (Group 2)	30	30	0	0	0
No Formal QA Methods (Group 3)	13	30	-17	289	9.63
Total	**90**	**90**			**19.26**

Degrees of Freedom

$$df = \text{Number of Categories} - 1 = 3 - 1 = 2$$

p-Value

Using a Chi-Square distribution table or calculator with:

- $\chi^2 = 19.26$
- $df = 2$

The **p-value ≈ 0.00006**.

Conclusion

At a significance level of $\alpha = 0.05$:

- $p = 0.00006 < 0.05$, so we **reject the null hypothesis**.

7. Result of Analysis

Sl no.	Question	Hypothesis (H_0)	Result
1	Have you ever been formally introduced to software engineering best practices (Yes/No)	The distribution of responses (Yes, No) matches the expected distribution.	Most of the students have been formally introduced to software engineering best practices.
2	What is your experience level in software development?	The distribution of experience levels (Advanced, Intermediate, Beginner) matches the expected distribution.	Most of the students are from beginner level.
3	Which software engineering practices do you follow in your projects?	The software engineering practices are uniformly distributed (all categories are equally likely).	Most of the students Use these practices as their wish or their skill.
4	How often do you refer to guidelines or resources for best practices during project development?	The preference for referring to guidelines or resources is used frequently.	Most of the students rarely follow these guidelines in their project development.
5	What challenges do you face in applying software engineering best practices?	The number of students experiencing each type of challenge is equal, and there are no significant differences between the groups.	The students may face their challenges more or less as their skill, resource allocation, etc.

6	What tools or resources have you used to learn software engineering best practices?	The tools or resources used to learn software engineering best practices are uniformly distributed (i.e., all resources are equally likely to be used).	Students use these resources when they get these available or their interest.
7	Do you believe that following best practices improves project outcomes?	Students do not believe that these best practices improve outcomes.	Students believe that these practices may help boost their project.
8	How do you ensure the quality of your code or project deliverables?	Students prefer automated testing.	Students prefer manual testing.

8. CONCLUSION

In conclusion, this study offers valuable insights into how university students approach software engineering practices and their application during project development. It reveals that while many students possess software development experience, a significant portion is at the beginner level. This indicates the need for more support and resources to advance their skills. The study also highlights that students tend to follow software engineering practices based on their individual preferences or abilities, rather than adhering to formal guidelines, which are rarely consulted. Furthermore, challenges in applying best practices vary depending on skills and resource availability. Although students believe that following best practices could improve project outcomes, many still face obstacles in fully implementing these practices. The study also shows a preference for manual testing over automated testing to ensure project quality. These findings emphasize the importance of providing students with consistent access to resources and guidance, encouraging the adoption of best practices, and fostering awareness of the value these practices bring to project outcomes. It also suggests the need for further education in automated testing and more structured support for overcoming challenges in applying software engineering best practices.

9. References

Survey questionnaire links: https://forms.gle/TAtgbtWu2WJ3Nhqg8

Chi test: https://www.indeed.com/career-advice/career-development/how-to-do-chi-square-in-excel

Response