INTERNATIONAL UNIVERSITY OF EAST AFRICA
'Learning to Succeed'

FACULTY OF SCIENCE AND TECHNOLOGY

END OF SEMESTER EXAMINATIONS - DECEMBER 2025

PROGRAMME: BSSE
YEAR/SEM: YEAR 4/SEMESTER 1
COURSE CODE: SWE4103
NAME: SOFTWARE METRICS

DATE:
TIME:

INSTRUCTIONS TO CANDIDATES:

SECTION A

Attempt All questions in Section A

Question 1:

XYZ Technologies is a mid-sized software development company that specializes in building custom enterprise software solutions for various clients. Recently, the company has been experiencing challenges in tracking the quality and performance of their software products, as well as in estimating costs and effort accurately. The management team has decided to implement a structured approach to software metrics in order to improve the development process, enhance software quality, and optimize project estimation.

As part of this initiative, they want to measure both internal product attributes (such as size, structure, and complexity) and external product attributes (like software reliability, user satisfaction, and performance). They are also interested in tracking effort metrics and cost metrics to gain better control over project budgeting and scheduling.

Using the case study above, address the following questions:

a) Define software metrics and explain why XYZ Technologies needs to adopt software metrics in their development process. (10 marks)

b) Explain the importance of measuring internal product attributes such as size, complexity, and structure in software development. Identify two types of internal product metrics XYZ Technologies can use and discuss how these metrics could help in improving their development process. (10 marks)

c) Describe the role of external product metrics in software quality assurance. Discuss at least two external product metrics XYZ Technologies can implement to measure software reliability and user satisfaction. (10 marks)

d) Discuss the importance of effort and cost metrics in project management. Identify two types of cost and effort metrics XYZ Technologies can use to better estimate project costs and resource allocation. How can these metrics help in improving the accuracy of project timelines and budgets? (10 marks)

Question 2:

ABC Software Solutions is a software engineering firm that develops complex systems for large-scale enterprises. They have been facing issues with software development delays, poor quality in the final product, and difficulty in meeting client expectations. The company's leadership recognizes the importance of a structured measurement approach to improve their processes.

To address these issues, ABC Software Solutions has decided to implement a goal-based framework for software measurement. They plan to measure both process metrics (such as software development productivity and test coverage) and product metrics (like product quality and defect density). They also want to investigate how empirical investigations (e.g., retrospective analysis of past projects) can help improve their future processes. Using the case study above, address the following questions:

a) Explain the concept of measurement theory in software engineering and describe how it can be applied to ABC Software Solutions. (10 marks)

b) Define empirical investigation in software engineering and discuss how ABC Software Solutions can use retrospective analysis of previous projects to improve their current processes. (10 marks)

c) What is the importance of measuring software reliability for a company like ABC Software Solutions. Explain at least two software reliability metrics that could be used to assess the stability and performance of their products. How can these metrics help reduce the frequency of software failures? (10 marks)

d) Explain the role of software test metrics in the software development lifecycle. Identify two key test metrics ABC Software Solutions should track and discuss how measuring these metrics can help improve the testing process and overall software quality? (10 marks)

SECTION B

Attempt any 3 questions in section B

Question 1:

a. What do function points measure and what are the major drawbacks?

b. Give three examples of how you might use the function point measure in quality control or assurance (stating in each case why it might be preferable to some simpler alternative measure).

c. Consider the following specification of a simple system for collecting student course marks or grades. The course marks system enables lecturers to enter student marks for a predefined set of courses and students on those courses. Thus, marks can be updated, but the lecturers cannot change the basic course information, as the course lists are the responsibility of the system administrator. The system is menu driven, with the lecturer selecting from a choice of courses and then a choice of operations. Compute the number of function points in this system, stating carefully the assumptions you are making. The operations include

i. Enter coursework marks

ii. Enter exam marks

iii. Compute averages

iv. Produce letter grades

v. Display information (to screen or printer)

The information displayed is always a list of the students together with all the known marks, grades, and averages.

Question 2:

a) Explain the concept of complexity metrics in software engineering. Provide examples of two widely used complexity metrics. (5 marks)

b) How do quality metrics help in assessing the overall quality of software products? List and explain three key quality metrics

commonly used. (5 marks)

c) Discuss the importance of effort metrics in software project management. What role do they play in project estimation and resource allocation? (5 marks)

d) Critically evaluate the potential drawbacks of relying solely on quantitative software metrics for decision-making. What steps can be taken to ensure the effective use of metrics in software engineering? (5 marks)

Question 3:

a) Define software metric and explain why they are important in software engineering. (5 marks)

b) Differentiate between product metrics and process metrics, providing at least two examples of each. (5 marks)

c) Describe the role of size metrics in software development. Explain how metrics such as lines of code (LOC) can influence software quality and maintenance. (5 marks)

d) Identify and discuss any three challenges associated with the collection and interpretation of software metrics in a software development project. (5 marks)

Question 4:

a) Briefly explain Halstead's theory (5 marks)

b) Given the snippet of code below, answer the following questions (10 marks)

```
#include<stdio.h>
main()
{
int a;
scanf(”%d”, &a);
if( a>=10)
if(a<20) printf(“10<a<20 %d\n”, a);
else  printf(“a>=20 %d\n”, a);
else  printf(“a<=10 %d\n”, a);
}
```

i. Determine number of Operators ($\mu 1$)

ii. Determine number of Operands ($\mu 2$)

iii. Determine the program length in terms of the total number of occurrences of operators (N1) and operands (N2): N = N1 + N2

iv. Estimate program length

c) What are the main factors that affect Software metrics? (5 marks)

Question 5:

a) Line of code is one of product metrics developed by William Barry Boehm. Suggest any two benefits and two demerits associated with this metric (4 marks)

b) One of the most reliable predictive software product metrics tools is the COCOMO, identify any FOUR parameters predicted by this tool. (4 marks)

c) After working on 63 software projects, Barry William Boehm, in 1970, categorized software systems into three types namely: Organic, Semi-detached and embedded. Differentiate among these types by illustrating the characteristics and two examples of each type. (6 marks)

d) Match the relevant COCOMO model to the following characteristic.

i. Used for quick and slightly rough calculations of Software Costs. (1 Mark)

ii. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations. takes Cost Drivers into account and (1 Mark)

iii. accounts for the influence of individual project phases. it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result. (2 mark)

e) Write COCOMO in full and define it. (2 Marks)

Question 6:

A project size of 150 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate:

a) the Effort

b) development time

c) average staff size

d) productivity of the project. (20 marks)

Question 7:

a) The management team at a small software development firm in Nairobi relies heavily on Lines of Code (LOC) as a primary measure of project size and programmer productivity. Discuss two significant limitations or weaknesses of using LOC as a size metric and explain why it can lead to inaccurate effort estimation. (06 Marks)

b) Define Function Point Analysis (FPA). Explain how FPA addresses the limitations of LOC by focusing on the functional user requirements. Briefly list the five primary components used when calculating Unadjusted Function Points. (08 Marks)

c) A startup is using the basic COCOMO formula for estimation: Effort = a * $Size^b$. Explain what the terms Effort and Size represent in this formula. If a project is estimated to be 10,000 Lines of Code, what kind of information does the constant “a” provide about the development environment? (06 Marks)

Question 8:

a) A newly developed mobile health application for rural clinics was released. During the first two months, 150 defects were reported by users. The application's size is measured at 5,000 Function Points. Calculate the Defect Density for this release and explain what this metric tells the Quality Assurance (QA) team about the software's quality. (Defect Density = Number of Defects /Size of Software) (06 Marks)

b) Define Defect Removal Efficiency (DRE). Why is DRE considered a critical metric for evaluating the effectiveness of the entire development and testing process (from requirements review to

system testing)? (06 Marks)

c) A critical financial transaction module deployed on a server in Johannesburg experienced failures after running for the following intervals (in hours): 50, 60, 40, 80. Calculate the Mean Time To Failure (MTTF). Why is increasing MTTF a key reliability objective for critical server-side applications? (08 Marks)

Question 9:

a) Distinguish between Process Metrics and Product Metrics in software engineering. Give one example of each and briefly explain how each type of metric supports the goal of continuous improvement within a software team. (06 Marks)

b) A development team in Cairo uses an Agile methodology. Define Lead Time (or Customer Lead Time) and Cycle Time in the context of their software development process. Which of the two metrics would a development manager focus on most closely to identify bottlenecks within the team's workflow? Justify your choice. (08 Marks)

c) In an Agile team, the Velocity metric is calculated for planning purposes. Define Velocity and explain why it is crucial for a team to calculate the average velocity over several past iterations (Sprints) rather than relying on the velocity of only the last one or two Sprints. (06 Marks)

Question 10:

a) The complexity of a payment processing function is being analyzed. Define Cyclomatic Complexity and explain what feature

of the code (e.g., control flow) this metric specifically measures. State how a high Cyclomatic Complexity score relates to the code's testability and maintainability. (08 Marks)

b) The Halstead Software Science Metrics measure complexity based on the count of unique operators and operands. Why are these metrics often referred to as "software science"? Briefly explain how counting the total number of operators and operands is used to calculate the program length. (06 Marks)

c) The Maintainability Index (MI) is an aggregated metric often calculated using Cyclomatic Complexity and LOC. If the MI score for a legacy system used by a power utility company is very low (e.g., below 20), what action does this suggest the software engineering team should take regarding the module, and why? (06 Marks)

Question 11:

a) Distinguish between a Direct Measurement and an Indirect Measurement in software engineering. Give one example of each (e.g., from the context of code quality) and explain why many important software attributes (like quality or reliability) must be measured indirectly. (08 Marks)

b) A public sector agency is launching a new software project and wants to ensure that all collected metrics are actually useful. Explain the Goal-Question-Metric (GQM) paradigm. Using an example, demonstrate how a specific Goal might be decomposed into a relevant Question and then tied to a measurable Metric. (06 Marks)

c) Suppose a team is collecting data on the time spent fixing bugs, but the developers are inconsistent in reporting their time. In the context of measurement theory, what is the term used to describe this problem with the data, and why does it undermine the validity of the resulting metric (e.g., Mean Time to Repair)? (06 Marks)

Question 12:

a) Identify and describe two different types of software metrics (e.g., product, process, or project) that a Project Manager could use to proactively identify and mitigate high-priority risks in a large software development project. Explain how the measurement provides an early warning signal for the risk. (08 Marks)

b) Explain the common management pitfall known as "Management by Metrics" (or goal displacement). Provide an example of how an over-emphasis on a single metric (like the number of bugs fixed per day) could negatively impact the overall quality and security of the final software product. (06 Marks)

c) An organization is looking to outsource the development of a non-critical module to a new third-party vendor. Explain how the concept of Benchmarking using industry standard metrics (e.g., Defect Density or average Schedule Variance) can be used to select the most reliable supplier. (06 Marks)