10 Software Metrics - exam questions

You should know the answers to these questions

Can you give three possible problems of metrics usage in software engineering? How does the measurement theory address them?

- **Problems**:
  - Preciseness: Difficulty in defining consistent units for metrics.
  - Representation Condition: Whether the metric truly reflects the attribute it measures (e.g., does "code size" indicate "code quality"?).
  - Scale Types: Misinterpretation of data due to incorrect scale usage (e.g., ordinal vs. ratio scales).
- **Measurement Theory** addresses these issues by defining valid scales, representation conditions, and clear domain-to-range mappings.

What's the distinction between a measure and a metric?

- A **measure** is a mapping of a real-world attribute to a symbol set with known mathematical relationships.
- A **metric** satisfies additional mathematical properties: m(x,x)=0m(x,x) = 0m(x,x)=0, m(x,y)=m(y,x)m(x,y) = m(y,x)m(x,y)=m(y,x), and m(x,z)≤m(x,y)+m(y,z)m(x,z) \leq m(x,y) + m(y,z)m(x,z)≤m(x,y)+m(y,z).

Can you give an example of a direct and an indirect measure?

- **Direct Measure**: Length of source code.
- **Indirect Measure**: Defect density = (Number of defects) / (Length of source code).

What kind of measurement scale would you need to say "A specification error is worse than a design error"? And what if we want to say "A specification error is twice as bad as a design error?"

- **Ordinal Scale** for "worse than" (ordering but no arithmetic operations).
- **Ratio Scale** for "twice as bad" (arithmetic operations permitted).

Explain the need for a calibration factor in Putnam's model.

- It adjusts effort and productivity based on project size, complexity, and team capabilities to improve accuracy in predictions.

Fill in the blanks in the following sentence. Explain briefly, based on the Putnam's model. If you want to finish earlier (= decrease scheduled time), you should ... the effort ... .

- Increase the effort **a lot**, according to Putnam's model.

Give three metrics for measuring size of a software product.

- Lines of Code (LOC), Function Points (FP), Use Case Points (UCP).

Discuss the main advantages and disadvantages of Function Points.

- **Advantages**: Measures functionality, language-independent, usable after design.
- **Disadvantages**: Requires expert judgment, not automatic, counterintuitive.

What does it mean for a coupling metric not to satisfy the representation condition?

- It means the metric does not reflect empirical relations accurately (e.g., a high cohesion class showing a high LCOM value).

Can you give 3 examples of impreciseness in Lines of Code measurements?

- Defining what counts as a line, code reuse or duplication, verbosity differences.

You should be able to complete the following tasks

Given a set of use cases (i.e. your project) calculate the use case points.

Given a set of user stories, perform a poker planning session.

Can you answer the following questions?

During which phases in a software project would you use metrics?

- Metrics are used during requirement specification, design, implementation, testing, and maintenance.

Why is it so important to have "good" product size metrics?

- Accurate size metrics are critical for reliable effort, cost estimation, and quality control.

Can you explain the two levels of calibration in COCOMO (i.e. C & S vs. M)? How can you derive actual values for these parameters?

- C and S are derived from regression analysis of historical projects; M is a finer calibration for specific attributes like quality and constraints.

Can you motivate why in software engineering, productivity depends on the scheduled time? Do you have an explanation for it?

- Productivity depends on time as reducing time increases effort disproportionately (time to the power of 4 in Putnam's model).

Can you explain the cone of uncertainty? And why is it so relevant to cost estimation in software projects?

- The cone of uncertainty shows decreasing estimation error over time. It highlights the need for early flexibility and later accuracy in estimation.

How can you decrease the uncertainty of a project bid using Putnam's model?

- By increasing the scheduled time or improving process productivity.

Why do we prefer measuring Internal Product Attributes instead of External Product Attributes during Quality Control? What is the main disadvantage of doing that?

- Internal attributes are measurable during development. Disadvantage: may not directly reflect external quality.

You are a project manager and you want to convince your project team to apply algorithmic cost modeling. How would you explain the technique?

- Algorithmic models use historical data and formulae to estimate costs systematically, reducing bias and improving consistency.

Where would you fit coupling/cohesion metrics in a hierarchical quality model like ISO 9126?

- Under maintainability in a hierarchical quality model.

Why are coupling/cohesion metrics important? Why then are they so rarely used?

- They measure modularity and maintainability but are complex and often misunderstood.

Do you believe that "defect density" says something about the correctness of a program? Motivate your answer?

- No, it reflects defect count relative to size, not functional correctness.