1.A software failure happens when a program doesn't do what it's supposed to. This could be because of mistakes in the code, bad design, or something unexpected happening while it's being used.

2.The purpose of metrics validation is to make sure the measurements we use are accurate and actually represent what we want to measure. This ensures the results are reliable and useful for decision-making.

3.Process measures are metrics used to evaluate and improve the efficiency, quality, and effectiveness of a software development process. They track things like the time taken to complete tasks, defect rates during development, or the resources used.

4.

1.Jira – Tracks and manages requirements and issues.

2.IBM DOORS – Manages complex requirements for large projects.

3.ReQtest – Provides requirements tracking and testing features.

4.Confluence – Supports documentation and collaboration on requirements.

5.Helix RM – Helps manage and trace requirements efficiently.

5. 1.Requirements Analysis – Check if all needs are clear and well-defined.

2.System Design – Measure complexity and how easy it is to manage.

3.Detailed Design – Track how reusable and maintainable the design is.

4.Prototyping – Measure performance and user feedback.

5.Final Design – Check code quality and readiness for implementation.

6.Defect Removal Efficiency (DRE) shows how good the team is at finding and fixing bugs before the software is released.It is calculated as:

DRE = (Defects Found Before Release) / (Defects Found Before Release + Defects Found After Release).A higher DRE means the software has fewer bugs after it's released.

7.Test Effectiveness Ratio measures how well the testing process is at finding defects in software. It is calculated as: Test Effectiveness Ratio = (Defects Found by Testing) / (Total Defects Found).A higher ratio means the tests are good at finding defects before the software is released.

8.McCabe's Cyclomatic Complexity measures how complex a program is by counting how many different paths there are in the code.It’s calculated using the formula:  
Cyclomatic Complexity = E - N + 2P  
Where E is the number of connections, N is the number of decision points, and P is usually 1 for a single program.Higher values mean the code is more complex and harder to maintain.

9.Validation is the process of making sure the software works as intended and solves the right problem. It ensures that the product meets user needs and expectations.

10.Mean Time To Change (MTTC) measures the average time it takes to make a change or fix in the software.

It helps track how long it takes from when an issue is identified until the change is fully implemented and tested. Lower MTTC values mean quicker response times for making improvements or fixes.