P P SAVANI UNIVERSITY

Tutorial NO. - 14
ON
SOFTWARE ENGINEERING(SSCS3010)

TITLE: To Understand Usage of Software Metrics.

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY (BSC-IT)

SUBMITTED TO: **SUBMITTED BY:**

Name: MS. HEMANGINI MEHTA(HGM)   **Name: RAJ MO FAHIM ZAKIR**

Designation: ASSOCIATE PROFESSOR   **Enrollment: 23SS02IT161**

P P Savani University   **BSCIT5B-Batch 2023-26**

Faculty Signature: _____

INSTITUTE OF COMPUTER SCIENCE AND APPLICATIONS
P P SAVANI UNIVERSITY
MANGROL, SURAT- 394125 (GUJARAT)

Student Name: RAJ MO FAHIM ZAKIR

Enrolment Number: 23SS02IT161

Subject Name: SOFTWARE ENGINEERING

Subject Code: SSCS3010

TUTORIAL-14	**Date:03/10/2025**

Aim: **To Understand Usage of Software Metrics.**

Software metrics are quantitative measures that assess different facets of software development. They offer a concrete way to score software quality, monitor progress, control complexity, and improve overall project performance.

Definition

Software Metrics are numerical indicators used to evaluate software products, processes, or projects, providing critical data for monitoring and controlling software development.

Why Software Metrics are Important

- Measure Quality:
  Metrics assess how maintainable, reliable, and efficient your resume builder is.
- Monitor Progress:
  Metrics help track the rate at which features are built and team productivity.
- Identify Risks:
  By measuring things like complexity or test coverage, you can catch potential problems early.
- Support Decision-Making:
  Metrics give managers hard data for task planning and resource allocation.
- Improve Processes:
  Lessons from metric trends in past updates help refine development workflows.

Types of Software Metrics

- Product Metrics:
  Measure characteristics of the resume builder itself, such as Lines of Code (LOC), complexity, or defect density.
- Process Metrics:
  Evaluate development activities, like time taken to build new modules or bug fix efficiency.
- Project Metrics:
  Track project health, such as cost spent, how much time or resources remain, and schedule adherence.

Examples Applied to Resume Builder

- Defect Density:
  If the resume builder has 6 bugs in 3000 LOC, defect density = 2 per 1000 LOC. Lower values indicate higher quality.
- Code Coverage:
  If you have automated test scripts covering 85% of your codebase, you ensure most features are tested.
- Cyclomatic Complexity:
  A module for "Preview Resume" with many conditions might have high complexity; lower values make code easier to maintain.
- Effort Estimation:
  Track hours spent on adding "Custom Themes"—helps future planning and resource management.

Usage in Resume Builder Development

- Quality is assured by keeping defect density low and code coverage high.
- Project managers estimate completion time and cost using metrics like team productivity and schedule variance.
- Development teams can focus on simplifying modules with high cyclomatic complexity, reducing future maintenance effort.
- Data from metrics help prioritize new features and optimize resource allocation, improving release velocity.

Conclusion

Software metrics are key to delivering a reliable, maintainable, and cost-effective resume builder system. Regular measurement and analysis drive better decisions at every project stage, underpinning your team's ability to deliver high-quality software.