



RV College of
Engineering®

Orientation on AI Integrated Software Engineering

By

Prof. Narasimha Swamy S

Department of AIML

R V College of Engineering®, Bengaluru-59

Go, change the world®



Software Engineering

- Software Engineering is a disciplined, systematic approach to designing, developing, and maintaining software.
- It combines principles of computer science, engineering, and project management to build high-quality software systems.
- Key Elements of SE
- Example: Develop a web-based application for an online bookstore that allows users to browse, search, purchase books, and track their orders





Software Engineering

→ Relevance in Modern IT

- Growing Complexity: Modern IT systems are becoming highly complex and interconnected, requiring a structured approach for effective development and management.
- Dependence on Software: Industries such as finance, healthcare, and entertainment rely heavily on software to operate, making SE crucial for reliability and performance.

→ Software Engineering as a Backbone of Quality Software

- Ensuring Reliability: Software engineering principles help create reliable, bug-free applications, critical for user trust.
- Scalability and Efficiency: Software needs to grow and adapt as user bases and data increase; SE provides frameworks for scalable and efficient systems.
- Security and Compliance: Proper SE practices ensure that software is not only functional but also secure and compliant with regulatory standards.

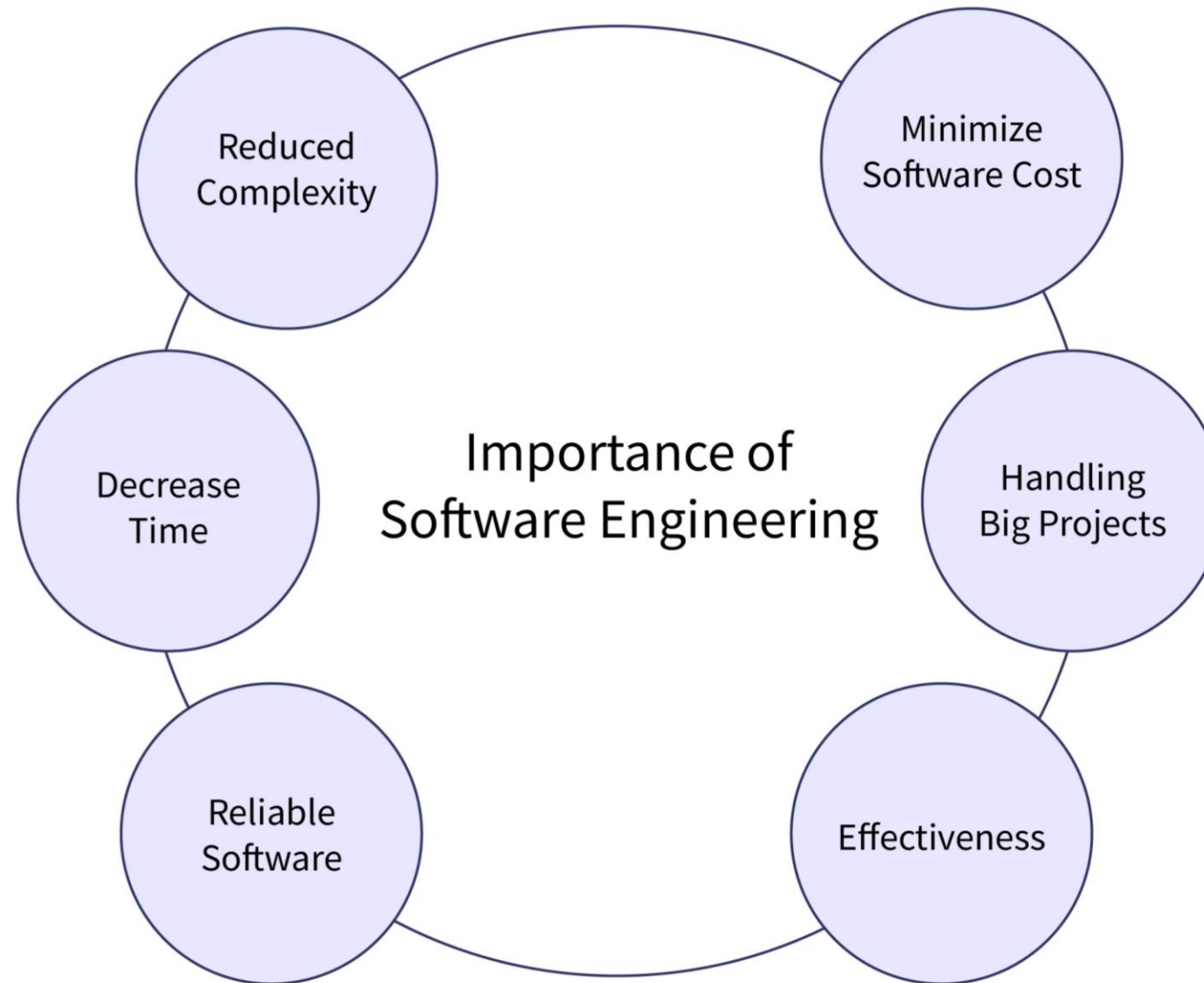


Goals of Software Engineering

- **Quality:** Ensuring that software meets user needs, is free of major bugs, and provides a high level of performance.
- **Efficiency:** Developing software in a way that optimizes resources, from development time to processing power.
- **Scalability:** Making sure that software can handle growth, whether in data size, user numbers, or complexity.
- **Maintainability:** Designing software so that it is easy to modify and improve over time, ensuring longevity and adaptability.



Benefits of Software Engineering





SE Job Roles

- Test Analyst
- Test engineer
- Automation Engineer
- Designing and Implementing Test Cases
- Software Tester
- Design Or Develop Automated Testing Tools
- Detecting Software Bugs And Errors
- Finalizing And Documenting Testing Procedures
- QA Analyst
- QA Tester
- Test Manager
- Create test plans
- Devops And Agile Methodology
- Documentation
- Engineering
- Quality engineering
- Quality Assurance Specialist
- Software Development Engineer In Test (SDET)



Syllabus

Unit-I	8Hrs.
Introduction: Professional Software Development, Software Engineering Ethics, Case studies. Software Processes: Models, Process activities, Coping with Change, Process improvement. The Rational Unified Process. Computer Aided Software Engineering. Agile Software Development: Introduction to agile methods, Agile development techniques, Agile project management and scaling agile methods.	
Unit – II	9Hrs.
Requirements Engineering and System Modeling: Software Requirements: Functional and Non-functional requirements. Requirements Elicitation, Specification, Validation and Change. System Modeling: Context models, Interaction models, Structural models, Behavioral models, Model driven architecture. Architectural Design: Design decisions, Architectural views, Architectural patterns and architectures.	
Unit –III	8Hrs.
Development and Testing: Design and implementation: Object oriented design using UML, Design patterns, Implementation issues, Open-source development. Software Testing: Development testing, Test-driven development, Release testing, User testing. Software Evolution: Evolution processes. Legacy system evolution, Software maintenance	
Unit –IV	10Hrs.
Machine Learning to Support Code Reviews in Continuous Integration Introduction, Code review in CI, Code analysis toolchain, Code extraction, Feature extraction, Model development, Making a recommendation, Visualization of the results, Full example. Using Artificial Intelligence for Auto-Generating Software for Cyber-Physical Applications Introduction, Model-Based Methods, Learning-Based Methods, Fault Trees, Model-Based Software Engineering, Running Example, AI-Based Framework for MBSE Task, AI-based MBSE Model Construction Methods, MBSE Trade-Off Framework, Empirical Modelling Cost Comparison	



Syllabus

Unit –V

Application of Machine Learning in Software Testing

Introduction, Applications of Machine Learning in software testing-Machine Learning for software fault prediction, Machine Learning for test oracles automation, Machine learning for test cases generation, Machine learning for test suite reduction, prioritization and evaluation, other tasks.

Creating Test Oracles Using Machine Learning Techniques

Introduction, Background on Test Oracles, Test Oracles Based on Machine Learning Techniques

Reference Books

- 1 **Ian Sommerville, Software Engineering, 10th Edition, Pearson Education, 2013, ISBN: 9788131762165.**
- 2 **Meir Kalech, Rui Abreu, Mark Last, Artificial Intelligence Methods for Software Engineering, World Scientific Publishing Co. Pte. Ltd, 1st Edition, 2021, ISBN 978-981-123-992-2, ISBN 978-981-123-993-9.**
- 3 Roger S Pressman, Software Engineering-A Practitioners Approach||, 7th Edition, Tata McGraw Hill, 2007, ISBN: 9780071267823
- 4 Rajib Mall, Fundamentals of Software Engineering, 3rd Edition, Prentice-hall Of India Pvt Ltd., 2012, ISBN: 9788120348981.

Thank You