SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS511 SOFTWARE TESTING

III YEAR / V SEMESTER

Unit 5- Scope of automation

IT8076 SOFTWARE TESTING

Syllabus

UNIT I INTRODUCTION 9

Testing as an Engineering Activity – Testing as a Process – Testing Maturity Model- Testing axioms – Basic definitions – Software Testing Principles – The Tester‘s Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design –Defect Examples- Developer/Tester Support of Developing a Defect Repository.

UNIT II TEST CASE DESIGN STRATEGIES 9

Test case Design Strategies – Using Black Box Approach to Test Case Design – Boundary Value Analysis – Equivalence Class Partitioning – State based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing - Random Testing – Requirements based testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Additional White box testing approaches- Evaluating Test Adequacy Criteria..

IT8076 SOFTWARE TESTING

UNIT III LEVELS OF TESTING 9

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing –Compatibility testing – Testing the documentation – Website testing

UNIT IV TEST MANAGEMENT 9

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group- The Structure of Testing Group- .The Technical Training Program.

*

IT8076 SOFTWARE TESTING

UNIT V TEST AUTOMATION 9

Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.

IT8076 SOFTWARE TESTING

TEXT BOOKS:

1.Srinivasan Desikan and Gopalaswamy Ramesh, ―Software Testing – Principles and Practices‖, Pearson Education, 2006.

2. Ron Patton, ―Software Testing‖, Second Edition, Sams Publishing, Pearson Education, 2007. AU Library.com

REFERENCES:

1.Ilene Burnstein, ―Practical Software Testing‖, Springer International Edition, 2003.

2. Edward Kit,‖ Software Testing in the Real World – Improving the Process‖, Pearson Education, 1995.

3. Boris Beizer,‖ Software Testing Techniques‖ – 2nd Edition, Van Nostrand Reinhold, New York, 1990.

4. Aditya P. Mathur, ―Foundations of Software Testing _ Fundamental Algorithms and Techniques‖, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.

INTRODUCTION

- **Definition:** Automation testing refers to the process of using software tools to execute predefined test cases on software applications automatically, without manual intervention.
- **Purpose:** The primary goal is to improve testing efficiency, accuracy, and coverage by automating repetitive and critical test cases.
- **Context:** Automation testing is essential in Agile and DevOps environments where rapid development cycles require quick and reliable testing.

WHY AUTOMATION TESTING?

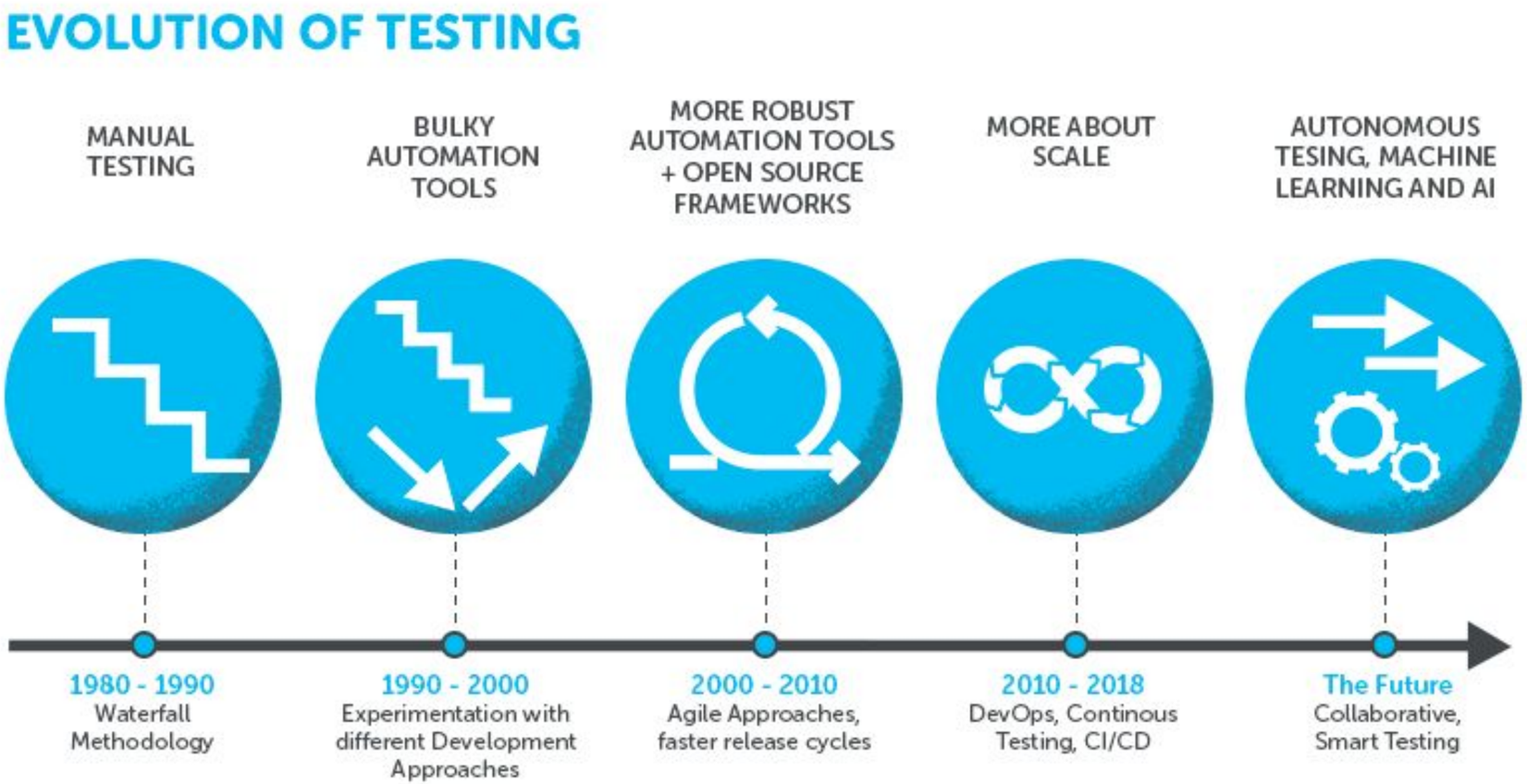
Efficiency: Automation allows the execution of thousands of test cases in a relatively short period, which is not feasible with manual testing.

Accuracy: Automated tests eliminate the risk of human error, ensuring consistent execution of tests.

Scalability: Automation supports large-scale testing across different platforms, browsers, and devices, ensuring comprehensive coverage.

Cost-Effectiveness: While initial setup might be expensive, automation reduces long-term costs by minimizing the need for extensive manual testing.

EVALUTION


EVOLUTION OF TESTING
MANUAL TESTING
BULKY AUTOMATION TOOLS
MORE ROBUST AUTOMATION TOOLS + OPEN SOURCE FRAMEWORKS
MORE ABOUT SCALE
AUTONOMOUS TESING, MACHINE LEARNING AND AI
1980 - 1990
Waterfall Methodology
1990 - 2000
Experimentation with different Development Approaches
2000 - 2010
Agile Approaches, faster release cycles
2010 - 2018
DevOps, Continous Testing, CI/CD
The Future
Collaborative, Smart Testing

BENEFITS OF AUTOMATION TESTING

Speed: Automation significantly reduces the time required for regression, functional, and performance testing, speeding up the overall development process.

Reusability: Once written, automated test scripts can be reused across different projects, versions, and environments, providing long-term value.

Continuous Testing: Automation is a key enabler of continuous testing within CI/CD pipelines, allowing for immediate feedback on code quality after each change.

Improved Test Coverage: Automation enables exhaustive testing, covering more test cases, including edge cases, that might be overlooked in manual testing.

KEY AREAS OF AUTOMATION TESTING

Regression Testing: Automated regression tests verify that new changes do not negatively impact existing functionalities, ensuring software stability over time.

Smoke and Sanity Testing: These quick, essential tests validate the critical functions of an application, ensuring it's ready for further testing.

Performance Testing: Automation tools like JMeter simulate multiple user interactions to test application performance under load, identifying bottlenecks and scalability issues.

API Testing: Automated API testing ensures that communication between different software components is reliable, consistent, and secure.

Cross-Browser and Cross-Platform Testing: Automation enables testing across various browsers and devices, ensuring consistent user experience.

AUTOMATION IN AI-DRIVEN DEVELOPMENT

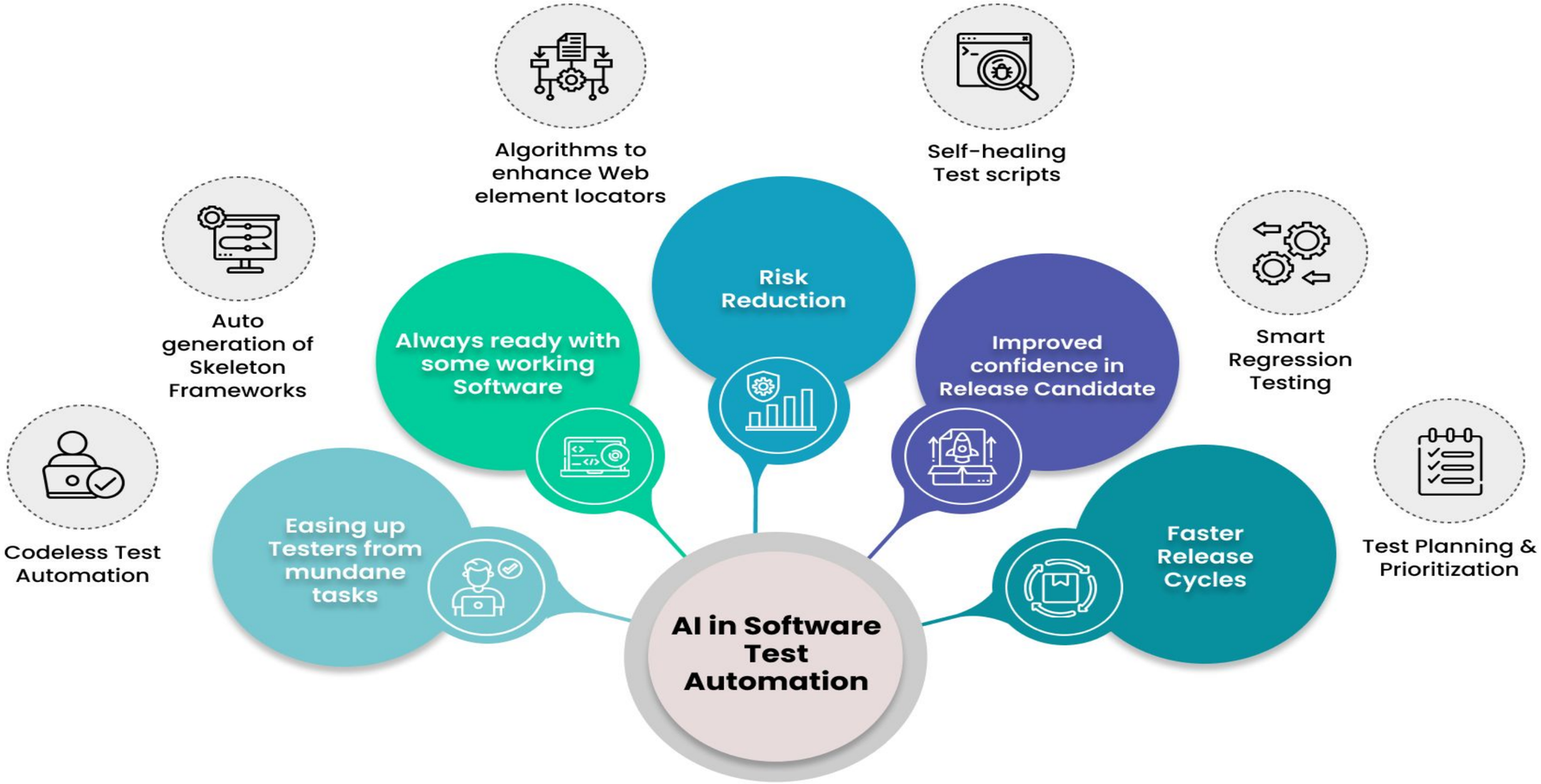
Introduction: AI is revolutionizing automation testing by making it smarter and more efficient.

AI-Powered Test Generation: AI tools can generate test cases based on code changes or user behavior, reducing the manual effort.

AI in Test Maintenance: AI can predict and fix flaky tests or adjust tests automatically when the application's UI changes.

Predictive Analysis: AI-driven tools analyze test results to predict potential failures in untested areas.

AI IN SOFTWARE TEST AUTOMATION



CHALLENGES IN AUTOMATION

Initial Setup Costs: Setting up an automation testing environment requires investment in tools, infrastructure, and skilled personnel, which can be expensive initially.

Maintenance: Automated test scripts need regular updates to reflect changes in the application, requiring ongoing effort.

Test Data Management: Managing test data effectively is challenging, especially when dealing with large and complex datasets.

Flaky Tests: Automated tests can sometimes produce inconsistent results (false positives/negatives) due to timing issues, environment configurations, or dependencies, requiring additional troubleshooting.

OVERCOMING CHALLENGES

Strategic Test Automation: Prioritize automation of test cases that are repetitive, high-risk, and provide the most value, such as regression tests.

Tool Selection: Carefully select tools that fit the project's needs, considering factors like ease of use, community support, and integration capabilities.

Regular Maintenance: Establish a routine for reviewing and updating test scripts to ensure they remain effective and aligned with the application's evolution.

Robust Test Data Management: Implement strategies for creating, managing, and using test data effectively, such as using mock data or data-driven testing approaches.

FUTURE TRENDS IN AUTOMATION

AI and Machine Learning: Emerging tools use AI to enhance test automation by generating test cases, analyzing results, and predicting areas of failure.

Test Automation in DevOps: As DevOps practices mature, test automation will become more tightly integrated with development and operations, enabling continuous delivery.

Shift-Left Testing: Testing will continue to move earlier in the development cycle, with more emphasis on early defect detection and prevention.

Robotic Process Automation (RPA): Automation is expanding beyond testing into business processes, using RPA to automate repetitive tasks across different systems.

AUTOMATION TESTING BEST PRACTICES

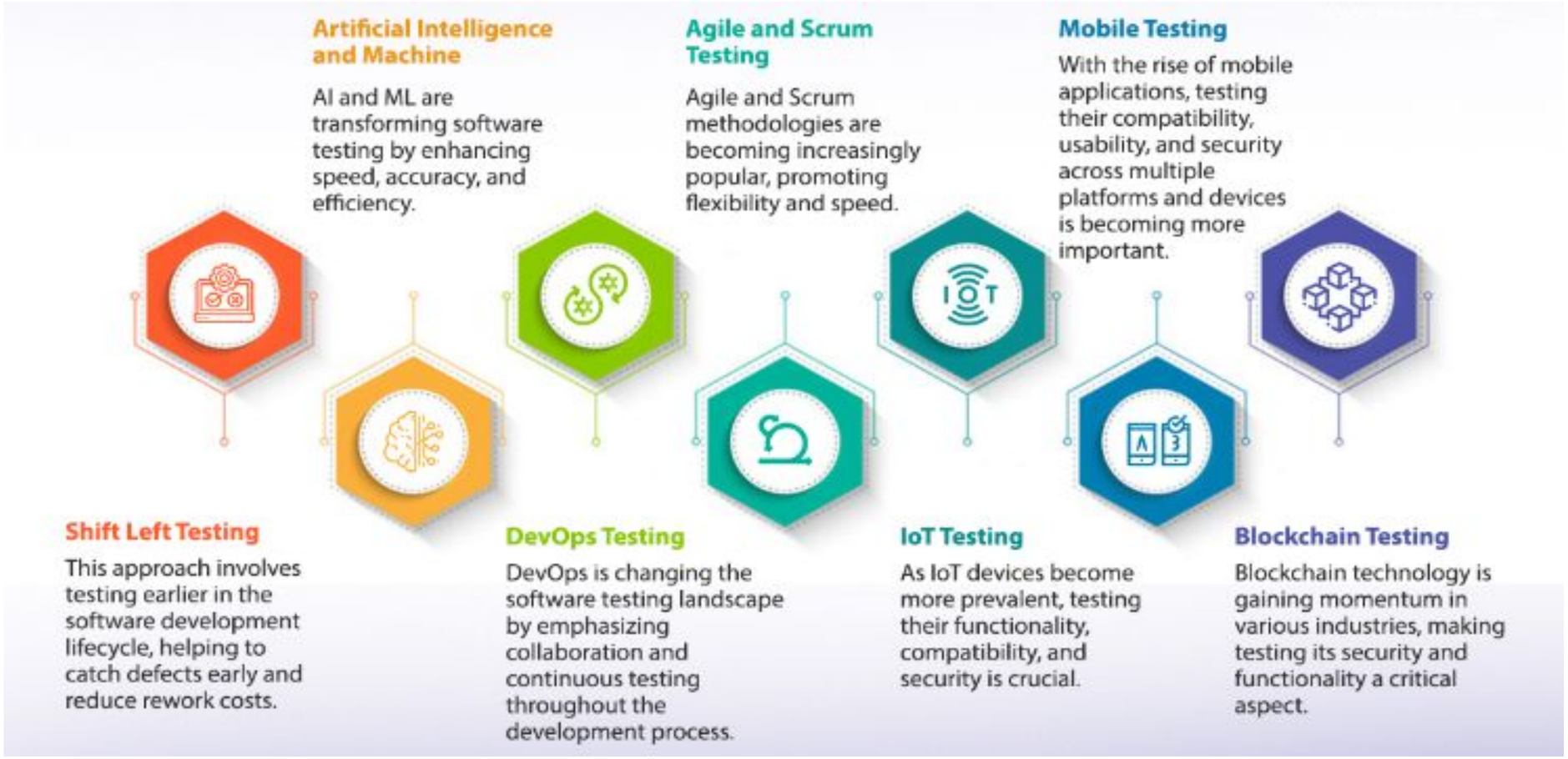
Set Clear Objectives: Define goals and scope for automation to ensure alignment with testing needs.

Design Modular Tests: Create reusable, modular test cases for maintainability and efficiency.

Maintain and Update Scripts: Regularly update test scripts and frameworks to keep them relevant and accurate.

Implement Data Management: Use data-driven testing and manage test data effectively for comprehensive coverage.

WHY SOFTWARE TESTING IS IMPORTANT ?



THANK YOU