Teaching Knowledge Test Handbook

Teaching knowledge test is an essential component in the journey of aspiring educators, serving as a comprehensive assessment to evaluate their understanding of core teaching principles, subject matter expertise, and pedagogical strategies. This test not only ensures that future teachers possess the necessary skills to deliver quality education but also maintains high standards within the teaching profession. Whether you're preparing for a certification exam, a licensure process, or professional development, understanding the nuances of the teaching knowledge test is crucial for success. In this article, we will explore the purpose, structure, preparation strategies, and key topics of the teaching knowledge test to help educators confidently approach this vital assessment.

Understanding the Teaching Knowledge Test

What Is a Teaching Knowledge Test?

A teaching knowledge test is an evaluative exam designed to measure an individual's understanding of essential teaching concepts, educational theories, classroom management techniques, and subject-specific knowledge. It is often a requirement for teacher certification, licensing, or employment in various educational settings.

Purpose of the Teaching Knowledge Test

The primary objectives of the teaching knowledge test include:

- Verifying a candidate's grasp of fundamental pedagogical principles.
- Ensuring familiarity with curriculum standards and learning objectives.
- Assessing readiness to handle classroom challenges effectively.
- Upholding quality standards within the education system.

Who Needs to Take the Test?

This test typically targets:

- Aspiring teachers pursuing certification or licensure.
- Teachers seeking to renew or upgrade their credentials.
- Education students preparing for entering the teaching profession.
- Substitute teachers or teaching assistants requiring certification.

Structure and Format of the Teaching Knowledge Test

Common Formats

While formats may differ depending on the country or certification body, most teaching knowledge tests include:

- Multiple-choice questions (MCQs)
- Short answer questions
- Scenario-based questions
- Essay or written response sections

Typical Content Areas Covered

The test generally encompasses several core areas:

- Educational Foundations

- Learning theories (behaviorism, constructivism, etc.)
- Educational psychology

- Curriculum and Assessment

- Curriculum development
- Types of assessment and evaluation methods

- Instructional Strategies

- Lesson planning
- Differentiated instruction
- Use of educational technology

- Classroom Management

- Behavior management techniques
- Creating inclusive learning environments

- Subject-Specific Knowledge

- Pedagogical approaches for specific subjects (e.g., math, science, language arts)

- Legal and Ethical Responsibilities

- Student rights
- Professional conduct and ethics

Preparing for the Teaching Knowledge Test

Effective Study Strategies

Successful preparation involves a mix of systematic study and practical application:

- Create a Study Schedule: Allocate ample time for each content area.
- Use Official Study Guides: Most certification bodies publish exam frameworks and sample questions.
- Engage in Practice Tests: Simulate test conditions to build confidence and improve time management.
- Review Key Concepts: Focus on understanding rather than rote memorization.
- Join Study Groups: Collaborate with peers to clarify doubts and share insights.

Key Resources for Preparation

- Official curriculum and assessment frameworks
- Educational psychology textbooks
- Online courses and practice exams
- Teaching forums and professional networks

Common Challenges and How to Overcome Them

- Test Anxiety: Practice relaxation techniques and simulate exam environments.
- Content Gaps: Identify weak areas through practice tests and focus revision accordingly.
- Time Management: Develop pacing strategies during practice exams to ensure timely completion.

Key Topics to Focus On

Educational Theories and Psychology

Understanding how students learn is fundamental:

- Behaviorism (e.g., Pavlov, Skinner)
- Constructivism (e.g., Piaget, Vygotsky)
- Social Learning Theory
- Multiple Intelligences

Curriculum Development and Planning

Key points include:

- Setting learning objectives
- Designing lesson plans
- Incorporating diverse teaching methods
- Aligning assessments with learning goals

Assessment and Evaluation

Topics encompass:

- Formative vs. summative assessment
- Designing effective tests and rubrics
- Using assessment data to inform instruction

Classroom Management and Inclusive Education

Essential skills involve:

- Establishing classroom rules
- Addressing behavioral issues
- Differentiating instruction for diverse learners
- Promoting inclusive practices

Educational Technology

Understanding how to integrate technology:

- Digital tools for instruction
- Online assessment platforms
- Managing hybrid or remote learning environments

Tips for Success on the Teaching Knowledge Test

Day of the Exam

- Arrive early and prepared
- Read questions carefully
- Manage your time wisely
- Review your answers if time permits

Post-Exam Strategies

- Analyze your performance to identify strengths and weaknesses
- Use feedback to guide future professional development
- Prepare for subsequent certification requirements if necessary

Conclusion

The teaching knowledge test is a vital step in establishing a competent and confident teaching workforce. By understanding its structure, key content areas, and effective preparation strategies, aspiring educators can approach the exam with confidence. Continuous learning and preparation not only help in passing the test but also lay a strong foundation for a successful teaching career. Remember, the key to success lies in thorough preparation, understanding core educational concepts, and developing practical classroom skills. With dedication and strategic study, you can excel in the teaching knowledge test and take a significant step toward fulfilling your passion for education.

Keywords for SEO Optimization:

- Teaching knowledge test
- Teacher certification exam
- Teaching licensure test
- Educational assessment
- Teacher preparation
- Classroom management strategies
- Teaching exam tips
- Pedagogical knowledge
- Curriculum development
- Educational theories

Teaching Knowledge Test: A Crucial Benchmark for Educators

Teaching knowledge test is increasingly recognized as a vital component in the journey toward becoming a qualified and effective educator. As education systems worldwide strive to improve teaching standards and student outcomes, the role of standardized assessments of teaching knowledge has gained prominence. These tests serve not only as gatekeepers for entry into the teaching profession but also as tools for ongoing professional development, ensuring that educators possess the core competencies necessary to foster learning and development in diverse classroom settings.

Understanding the Teaching Knowledge Test

What Is a Teaching Knowledge Test?

A teaching knowledge test is a standardized assessment designed to evaluate a candidate?s understanding of fundamental teaching principles, pedagogical strategies, curriculum content, and educational theories. Unlike subject-specific exams that focus solely on content mastery, teaching knowledge tests encompass a broader scope, evaluating the candidate?s grasp of effective teaching practices, classroom management, assessment techniques, and educational policies.

Typically administered by national or regional education authorities, teacher training institutions, or certification bodies, these tests aim to ensure that prospective teachers possess the essential knowledge base to support student learning effectively. They are often a prerequisite for admission into teacher training programs or for obtaining a professional teaching license.

Core Components of the Test

While the specific content may vary across countries and certification bodies, most teaching knowledge tests include several core areas:

- Educational Theories and Principles: Understanding of foundational theories such as behaviorism, constructivism, and social learning, and their implications for teaching practice.
- Curriculum and Content Knowledge: Familiarity with curriculum standards, subject content, and how to deliver it effectively.
- Instructional Strategies: Knowledge of various teaching methods, classroom activities, differentiation, and inclusive practices.
- Assessment and Evaluation: Ability to design, implement, and interpret assessments to monitor student progress.
- Classroom Management: Strategies for creating a positive learning environment, managing diverse student behaviors, and fostering engagement.
- Educational Policy and Ethics: Awareness of legal, ethical, and policy frameworks governing education.

The Significance of the Teaching Knowledge Test

Ensuring Quality and Consistency in Teaching

One of the primary reasons for implementing teaching knowledge tests is to uphold a consistent standard of teaching quality. By assessing core competencies before certification, education authorities aim to ensure that all teachers possess a minimum level of knowledge that can translate into effective classroom practice. This standardization helps reduce variability in teaching quality across schools and regions, leading to more equitable educational outcomes.

Gatekeeping and Professional Accountability

The teaching knowledge test functions as a gatekeeper, screening out candidates who may lack essential pedagogical knowledge. It encourages aspiring teachers to rigorously prepare and demonstrates their commitment and competence. Moreover, it fosters a sense of accountability, emphasizing that teaching is a professional discipline requiring specialized knowledge and skills.

Foundation for Professional Development

Beyond initial certification, teaching knowledge tests can serve as a baseline for ongoing professional development. Teachers can identify areas of weakness and seek targeted training or coursework, fostering a culture of continuous improvement. Some systems also require periodic re-assessment to maintain licensure, ensuring that educators stay current with evolving pedagogical practices and educational policies.

Designing an Effective Teaching Knowledge Test

Test Format and Question Types

Effective tests employ a variety of question formats to comprehensively assess candidate knowledge. Common formats include:

- Multiple-Choice Questions (MCQs): Widely used for assessing factual knowledge and understanding of key concepts.
- Scenario-Based Questions: Present real or hypothetical classroom situations to evaluate problem-solving and pedagogical decision-making.
- Short Answer and Constructed Response: Require candidates to articulate their understanding or strategies in written form.
- Case Studies: In-depth analysis of specific teaching challenges, testing analytical and reflective skills.

Alignment with Teaching Standards

Test content should align with national or regional teaching standards and frameworks. This alignment ensures relevance and promotes coherence between assessment and actual teaching practice. For example, many countries adopt standards from organizations such as the National Board for Professional Teaching Standards (NBPTS) or similar entities, customizing them to local contexts.

Validity and Reliability

An effective teaching knowledge test must demonstrate high validity (accurately measuring what it intends to measure) and reliability (producing consistent results over time). This involves rigorous test development procedures, including pilot testing, item analysis, and expert review.

Incorporating Technology

Modern assessments often leverage technology to improve accessibility and efficiency. Computer-based testing allows for diverse question formats, immediate scoring, and broader reach. Adaptive testing can tailor difficulty levels based on candidate responses, providing a more precise assessment of knowledge.

Challenges and Criticisms of Teaching Knowledge Tests

Overemphasis on Memorization

Critics argue that multiple-choice and factual questions may encourage rote memorization rather than fostering deep understanding or pedagogical creativity. This concern highlights the need to balance knowledge assessment with practical skill evaluation.

Contextual Limitations

Standardized tests may not fully capture a candidate?s ability to adapt to real classroom challenges, cultural diversity, or individual student needs. Teaching is a complex, context-dependent profession, and assessments must evolve to reflect this complexity.

Potential for Test Anxiety

High-stakes testing can induce anxiety among candidates, which may impact performance and not accurately reflect their teaching potential. Support systems and test preparation resources are vital to mitigate these effects.

Equity and Accessibility

Ensuring fair access to testing facilities and accommodations for candidates with disabilities is essential. Additionally, socioeconomic factors can influence preparation opportunities, raising concerns about equity.

The Future of Teaching Knowledge Tests

Integrating Practical Assessments

Many educational systems are exploring the integration of practical assessments, such as teaching demonstrations or micro-teaching sessions, alongside written tests. Such approaches provide a more holistic view of a candidate?s readiness.

Use of Digital Platforms and Artificial Intelligence

Advances in technology promise more personalized and adaptive assessments. AI-driven platforms can analyze responses in real-time, providing nuanced insights into candidate competencies and supporting formative feedback.

Global Trends and Collaborations

International collaborations aim to develop standardized frameworks that facilitate teacher mobility and recognition across borders. Initiatives like the UNESCO International Institute for Capacity Building in Africa (IICBA) promote shared standards and best practices.

Conclusion

The **teaching knowledge test** stands as a cornerstone in the journey toward professionalizing teaching and ensuring quality education. While it faces challenges, its role in safeguarding standards, fostering accountability, and guiding professional development remains vital. As education evolves to meet the demands of diverse learners and technological advancements, so too must the assessments that underpin the teaching profession?balancing rigorous evaluation with a nuanced understanding of the multifaceted nature of teaching itself. Ultimately, a well-designed teaching knowledge test not only certifies competence but also inspires ongoing growth, innovation, and excellence in classrooms worldwide.

Microsoft Test Manager Tutorial

Microsoft Test Manager Tutorial: A Comprehensive Guide to Efficient Test Management

In the world of software development, ensuring the quality and reliability of your applications is paramount. Microsoft Test Manager (MTM) is a powerful tool designed to streamline the testing process, enabling teams to plan, execute, and track testing efforts with ease. Whether you're a beginner or looking to enhance your testing workflows, this **Microsoft Test Manager tutorial** will provide you with a step-by-step guide to harnessing MTM's full potential for your projects.

Understanding Microsoft Test Manager

Before diving into the tutorial, it's essential to grasp what Microsoft Test Manager is and how it integrates into the broader Visual Studio ecosystem.

What is Microsoft Test Manager?

Microsoft Test Manager is a test management tool integrated with Visual Studio Team Services (VSTS) and Azure DevOps. It provides an intuitive interface for managing test plans, creating and executing test cases, and tracking testing progress. MTM supports both manual and exploratory testing, making it versatile for various testing scenarios.

Key Features of Microsoft Test Manager

- Test Planning: Create and organize test plans, test suites, and test cases.
- Test Execution: Execute manual tests and record results efficiently.

- Test Tracking: Monitor testing progress and generate detailed reports.
- Exploratory Testing: Conduct ad-hoc testing sessions with session-based testing capabilities.
- Integration: Seamlessly connects with Azure DevOps for continuous testing workflows.

Setting Up Microsoft Test Manager

Getting started with MTM involves installing and configuring the tool for your environment.

Prerequisites

- Visual Studio Enterprise or Professional edition (with test management features).
- Azure DevOps account or TFS (Team Foundation Server) setup.
- Appropriate permissions to access test management features.

Installing Microsoft Test Manager

- Download the Visual Studio Installer from the official Microsoft website.
- Choose the edition that includes testing tools (e.g., Visual Studio Enterprise).
- During installation, select the "Testing Tools" workload to install MTM.
- Complete the installation and launch Visual Studio.
- Connect to your Azure DevOps project or TFS server within Visual Studio.

Creating and Managing Test Plans

A well-structured test plan is the foundation of successful testing. MTM simplifies organizing your testing efforts through test plans and suites.

Creating a Test Plan

- Open Microsoft Test Manager from Visual Studio or the Azure DevOps portal.
- Navigate to the "Test Plans" section.
- Click "New Test Plan" and provide a descriptive name, description, and start/end dates.
- Associate the test plan with a specific build or release if needed.
- Save the test plan to begin adding test cases.

Organizing Test Suites

Test suites help categorize test cases based on features, modules, or testing strategies.

- **Static Test Suites:** Manually organize test cases into logical groups.
- **Requirement-Based Test Suites:** Link test cases to specific requirements or user stories.
- **Query-Based Test Suites:** Dynamically generate test cases based on queries.

Adding Test Cases

- Within your test suite, click "New Test Case" or import existing cases.
- Define the test case steps, expected results, and associated requirements.
- Assign priority, owner, and other metadata to facilitate management.
- Save the test case to include it in your suite.

Executing Tests with Microsoft Test Manager

Once your test cases are organized, the next step is executing tests and recording results.

Manual Test Execution

- Open the test plan and select the test suite to execute.
- Click "Run" on the test case you wish to execute.
- Follow the test steps as documented, marking each step as "Passed" or "Failed."
- If a test fails, record detailed bug information directly from the test runner.
- Complete the test run, and the results will be saved automatically.

Using Action Recording and Data Collection

MTM allows capturing detailed logs and screenshots during test execution, aiding in defect analysis and reproducibility.

Exploratory Testing

For ad-hoc testing, MTM offers session-based testing:

- Schedule a testing session with session details.
- Conduct testing while documenting observations and findings.
- Record bugs and issues discovered during the session.

Tracking and Reporting Testing Progress

Effective test management requires insight into testing status and quality metrics.

Monitoring Test Results

- Navigate to the "Test Results" section in MTM or Azure DevOps.
- Review individual test case outcomes and overall progress.
- Filter results by tester, status, or test plan for detailed analysis.

Generating Reports

MTM provides various built-in reports to visualize testing status:

- Test Results Summary
- Test Case Progress and Trends
- Bugs and Defects Reports

Integrating with Bug Tracking

During testing, bugs can be logged directly from MTM, linked to specific failed test cases, ensuring traceability and quick resolution.

Advanced Tips and Best Practices

Maximize your testing efficiency with these expert tips:

Automate Repetitive Tests

Integrate MTM with automated testing frameworks like MSTest, NUnit, or Selenium to run automated tests alongside manual efforts.

Maintain Test Cases Regularly

Keep your test cases updated with application changes to ensure relevance and accuracy.

Use Tags and Filters

Leverage tags and filters to quickly locate and execute specific test cases based on criteria like priority, feature, or owner.

Collaborate Effectively

Encourage team collaboration by sharing test plans and results, and conducting regular review meetings to discuss testing progress.

Conclusion

Microsoft Test Manager is an essential tool for teams aiming to implement structured and efficient testing workflows. From creating comprehensive test plans and organizing test suites to executing tests and tracking outcomes, MTM provides a centralized platform that enhances test visibility and quality assurance. By following this **Microsoft Test Manager tutorial**, you can streamline your testing process, improve defect detection, and deliver higher-quality software products. Remember to stay updated with the latest features and best practices to maximize your testing success with MTM.

Microsoft Test Manager tutorial: A Comprehensive Guide to Streamlining Your Testing Process

Microsoft Test Manager (MTM) is a powerful tool integrated within the Azure DevOps suite, designed to facilitate efficient test planning, management, and execution. Whether you are a QA professional, a developer, or a project manager, mastering MTM can significantly enhance your testing workflows, improve defect detection rates, and ensure higher quality software releases. This tutorial aims to provide an in-depth overview of Microsoft Test Manager, guiding you through its features, setup, best practices, and real-world applications.

Introduction to Microsoft Test Manager

Microsoft Test Manager is a dedicated testing management environment that helps teams plan, execute, and track testing activities seamlessly. Originally part of Visual Studio Test Professional, MTM is now integrated within Azure DevOps, offering a unified platform for Agile testing, exploratory testing, and manual test case management.

Key Features of Microsoft Test Manager

- Test Planning: Create and organize test plans, test suites, and test cases.
- Test Execution: Run manual tests, record results, and capture screenshots.
- Test Management: Track defects, manage test configurations, and generate reports.
- Integration: Seamlessly connect with Azure DevOps for continuous integration and automated testing workflows.
- Exploratory Testing: Record exploratory sessions with detailed notes and recordings.

Setting Up Microsoft Test Manager

Before diving into testing, proper setup of MTM is essential.

Prerequisites

- A valid Azure DevOps account with appropriate permissions.
- Installed Visual Studio Test Professional or access via Azure DevOps.
- Access to a test environment or lab machines.

Installation and Configuration

- Download and Install: Download Microsoft Test Manager from the Visual Studio downloads page or via Azure DevOps.
- Connect to Azure DevOps: Launch MTM and connect it to your Azure DevOps project by signing in with your credentials.
- Configure Test Environments: Set up test machines, configurations, and network environments to align with your testing needs.

Tips for Effective Setup

- Organize your test plans hierarchically for easier management.
- Maintain a clear mapping between test environments and real-world configurations.
- Regularly update test artifacts to ensure they reflect current application versions.

Creating and Managing Test Plans

Test planning is the foundation of effective testing.

Creating a Test Plan

- In MTM, navigate to the "Test Plans" section.
- Click "New Test Plan" and provide a descriptive name, start/end dates, and any relevant details.
- Assign ownership and stakeholders.

Organizing Test Suites

- Static Test Suites: Manually added test cases grouped logically.
- Requirement-based Suites: Linked to user stories or requirements for traceability.
- Query-based Suites: Dynamic grouping based on queries.

Best Practices

- Break down large test plans into smaller, manageable suites.
- Link test cases to user stories or bugs for better traceability.
- Regularly review and update test plans to reflect current project scope.

Designing and Managing Test Cases

Creating effective test cases is critical to uncovering defects early.

Writing Test Cases

- Clear, concise steps with expected results.
- Use descriptive titles and tags for easy filtering.
- Include preconditions and postconditions.

Reusing Test Cases

- Modularize test steps for reuse across different test cases.
- Attach relevant documentation, screenshots, or scripts.

Organizing Test Cases

- Group related test cases into shared test suites.
- Use labels and tags for categorization.

Tips for Effective Test Cases

- Prioritize test cases based on risk and impact.
- Keep test cases up-to-date with application changes.
- Incorporate exploratory testing notes within test cases or as separate sessions.

Test Execution and Result Recording

Executing tests efficiently and accurately recording results is vital.

Running Manual Tests

- Select a test case from your test suite.
- Follow each step, observing the actual behavior.
- Record outcomes: Passed, Failed, Blocked, or Not Applicable.
- Capture screenshots or record videos if needed.

Recording Defects

- Link defects directly to failed test cases.
- Provide detailed descriptions, steps to reproduce, and severity.
- Attach logs or screenshots to aid debugging.

Managing Test Runs

- Use test configurations to run multiple test cases under different environments.
- Schedule and assign test runs to team members.
- Monitor real-time progress and results.

Pros and Cons of Test Execution Features

Pros:

- Streamlined process for manual testing.
- Rich media capture for detailed documentation.
- Easy defect linkage and traceability.

Cons:

- Manual process can be time-consuming for large test suites.
- Limited automation capabilities within MTM itself.

Integrating Automated Testing with Microsoft Test Manager

While MTM excels at manual testing, integration with automated testing frameworks enhances overall efficiency.

Integration with Visual Studio Test Automation

- Use Visual Studio Test tools to develop automated test scripts.
- Link automated tests to test cases in MTM.
- Run automated tests as part of continuous integration pipelines.

Benefits of Integration

- Reduces manual effort.
- Ensures consistency between manual and automated testing.
- Facilitates comprehensive test coverage.

Limitations

- Setup can be complex for beginners.
- Requires scripting knowledge for automation.

Reporting and Tracking

Effective reporting provides insights into testing progress and quality metrics.

Built-in Reports

- Test Run Summary
- Test Results Trend
- Defect Status and Age
- Requirements Traceability Matrix

Custom Reports

- Use Azure DevOps Analytics or Power BI for tailored dashboards.

- Export data for external analysis.

Best Practices

- Regularly review reports to identify bottlenecks.
- Use metrics to prioritize testing efforts.
- Maintain up-to-date dashboards for stakeholder visibility.

Best Practices for Using Microsoft Test Manager

- Plan Early: Incorporate testing early in the development lifecycle.
- Maintain Test Artifacts: Keep test cases and plans updated with application changes.
- Leverage Linkages: Connect requirements, test cases, defects, and build versions.
- Automate Where Possible: Use automation to handle repetitive and regression testing.
- Collaborate: Engage team members through shared test plans and results.
- Review Regularly: Conduct test reviews and retrospectives for continuous improvement.

Common Challenges and How to Overcome Them

Challenge: Managing Large Test Suites

Solution: Break down into smaller, manageable suites; use query-based suites for dynamic grouping.

Challenge: Keeping Test Cases Up-to-date

Solution: Establish a review cycle aligned with development sprints; assign ownership.

Challenge: Integration Complexity

Solution: Invest in automation and continuous integration pipelines; use documentation and training.

Challenge: Limited Automation within MTM

Solution: Use external automation tools integrated with Azure DevOps, such as Selenium or Coded UI.

Conclusion

Microsoft Test Manager is a robust tool that, when used effectively, can significantly boost your testing efficiency, coverage, and traceability. Its comprehensive features for test planning, execution, defect management, and reporting make it indispensable for teams adopting Agile and DevOps practices. While there are challenges, especially around automation and managing large test repositories, with proper setup, training, and best practices, MTM can become a cornerstone of your software quality assurance process.

By following this tutorial, you should now have a clear understanding of how to leverage Microsoft Test Manager to streamline your testing operations, improve collaboration, and ultimately deliver higher quality software to your users. Remember, continuous learning and adaptation are key to maximizing the benefits of MTM in your projects.

QuestionAnswer What are the main features of Microsoft Test Manager (MTM)? Microsoft Test Manager (MTM) provides features such as test planning, manual test execution, test case management, bug tracking integration, and collaboration tools to streamline the testing process within Visual Studio and Azure DevOps. How do I create and organize test plans in Microsoft Test Manager? In MTM, you can create test plans by navigating to the 'Test Plans' hub, clicking 'New Test Plan,' and defining its name and details. You can then add test suites and test cases to organize tests based on requirements, features, or test types for better management. Can I automate tests using Microsoft Test Manager, and how is it integrated with other tools? While MTM primarily supports manual testing, it integrates seamlessly with Visual Studio and Azure DevOps, enabling automation through test scripts and frameworks like MSTest, Selenium, or CodedUI. Automated tests can be linked to test cases for comprehensive test management. What are the best practices for using Microsoft Test Manager effectively? Best practices include maintaining detailed and reusable test cases, organizing tests into logical suites, utilizing shared steps for common actions, integrating with bug tracking, and regularly updating test plans based on project changes to ensure thorough testing coverage. Is Microsoft Test Manager suitable for Agile development environments? Yes, MTM is well-suited for Agile teams as it supports iterative testing, easy integration with Azure DevOps Agile tools, and flexible test planning, allowing teams to adapt testing activities to sprint cycles and continuous delivery workflows.

Related keywords: Microsoft Test Manager, TFS testing, test planning, test case management, automated testing, test execution, bug tracking, test lab management, test reporting, test automation tools

Read the full article online: Microsoft Test Manager Tutorial