

pltw unit 3 test answers Handbook

pltw unit 3 test answers are a common concern among students enrolled in Project Lead The Way (PLTW) courses, particularly in Unit 3, which covers fundamental engineering principles and design processes. Understanding the content and preparing effectively for the tests can significantly enhance student success and comprehension of key concepts. This article provides a comprehensive overview of PLTW Unit 3, offers insights into typical test questions, and shares tips on how to study efficiently, ensuring students are well-equipped to excel.

Overview of PLTW Unit 3: Engineering Design & Development

PLTW Unit 3 typically focuses on the engineering design process, problem-solving strategies, and the development of workable solutions. It emphasizes creativity, critical thinking, and teamwork, essential skills for aspiring engineers.

Key Topics Covered in Unit 3

- Understanding the Engineering Design Process
- Identifying and Analyzing Problems
- Developing Design Solutions
- Creating and Communicating Design Ideas
- Prototyping and Testing
- Iterative Design Improvements

Typical Structure of the Unit 3 Test

The test usually assesses students' grasp of theoretical concepts, practical applications, and problem-solving skills. It may include various question formats, such as multiple-choice, short answer, and application-based problems.

Common Question Types

- **Multiple-Choice Questions:** Testing knowledge of definitions, steps in the design process, and basic concepts.
- **Short Answer Questions:** Requiring explanations of processes or reasoning behind design choices.
- **Application Problems:** Presenting real-world scenarios where students must develop or

analyze solutions.

Sample Topics and Possible Questions

While I cannot provide specific test answers, I can outline typical questions and the concepts they target, helping students prepare effectively.

Understanding the Engineering Design Process

Questions may ask students to sequence the steps correctly or describe each phase:

- Define the main stages: Ask, Develop, Prototype, Test, and Improve.
- Explain the importance of iteration in design.

Problem Identification and Analysis

Sample questions might include:

- How do engineers identify customer needs?
- What tools are used to analyze problems (e.g., brainstorming, root cause analysis)?

Design Development and Communication

Potential questions could involve:

- Creating sketches or technical drawings to communicate ideas.
- Using CAD software or other tools to develop design prototypes.

Prototyping and Testing

Sample questions may focus on:

- The purpose of prototyping in the engineering process.
- How testing can lead to iterative improvements.

Strategies for Finding or Preparing for PLTW Unit 3 Test Answers

Students often seek out test answers online or through study guides, but the most effective approach is thorough understanding and preparation.

Effective Study Tips

- **Review Class Notes and Assignments:** Ensure you understand all concepts discussed in class.
- **Study the Engineering Design Process:** Memorize the stages and be able to explain each step.
- **Use Practice Quizzes:** Many online platforms or textbooks offer practice questions that simulate test conditions.
- **Form Study Groups:** Collaborate with peers to discuss and clarify complex topics.
- **Utilize Visual Aids:** Flowcharts, diagrams, and sketches can help solidify understanding.
- **Ask Your Instructor:** Clarify doubts directly with your teacher or instructor for guidance.

Resources to Support Study and Practice

- PLTW Student Guide and Workbook
- Online educational platforms offering engineering modules
- YouTube tutorials on engineering design and prototyping
- Sample tests and quizzes available through educational websites

Ethical Considerations and Academic Integrity

While seeking answers can be tempting, it's crucial to uphold academic integrity. Cheating or using unauthorized answers can jeopardize your learning and violate school policies. Focus on understanding the material, which will benefit you beyond the test.

Strategies for Honest Preparation

- Use practice questions to reinforce understanding, not to memorize answers.
- Discuss challenging topics with teachers or classmates for clarification.
- Attend study sessions or tutoring if available.
- Apply concepts through hands-on projects or simulations.

Conclusion

Understanding **PLTW Unit 3 test answers** involves more than memorizing solutions; it requires a

solid grasp of engineering principles, the design process, and problem-solving strategies. Preparing thoroughly through review, practice, and collaboration ensures that students are well-equipped to succeed. Remember, the goal of PLTW courses is to develop critical thinking and creativity?skills that extend well beyond any single test. Approach your studies honestly and diligently, and you'll build a strong foundation for future engineering challenges.

PLTW Unit 3 Test Answers are a critical resource for students navigating the Principles of Engineering (POE) course, particularly as they prepare for assessments that evaluate their understanding of engineering principles, problem-solving skills, and technical knowledge. Mastering the content of Unit 3, which typically covers topics such as mechanism design, gears, levers, and mechanical advantage, is essential for success. This comprehensive guide aims to provide an in-depth analysis of common questions, concepts, and strategies to excel in the Unit 3 test, ensuring students are well-equipped to tackle challenging problems confidently.

Understanding the Importance of PLTW Unit 3 Test Answers

In the context of the PLTW (Project Lead The Way) curriculum, Unit 3 often focuses on the fundamentals of mechanical systems, including how different mechanisms work together to transfer motion and force. The PLTW Unit 3 test answers serve as a valuable tool for reviewing key concepts, practicing problem-solving techniques, and gauging readiness for the actual exam. Rather than just memorizing answers, students should aim to understand the reasoning behind each concept, which enhances long-term retention and application skills.

Core Concepts Covered in PLTW Unit 3

Before diving into specific answers, it's essential to grasp the foundational topics typically assessed in this unit:

Mechanical Advantage and Simple Machines

- Understanding how levers, pulleys, and gears provide mechanical advantage.
- Calculating mechanical advantage using formulas.
- Recognizing the types of simple machines and their applications.

Gears and Gear Trains

- Types of gears (spur, bevel, worm, etc.).
- Gear ratios and their effects on speed and torque.
- Designing gear trains for specific mechanical functions.

Linkages and Mechanisms

- Types of linkages (four-bar, slider-crank, etc.).
- How linkages convert motion and force.
- Analyzing linkage diagrams to determine motion paths.

Force and Motion Analysis

- Applying Newton's laws to mechanical systems.
- Calculating force, torque, and power in mechanisms.
- Understanding how different components interact to produce desired outputs.

Strategies for Approaching the PLTW Unit 3 Test

Success on the test is not solely about knowing the answers but understanding how to approach questions systematically:

- Review Key Vocabulary and Definitions

Familiarize yourself with terminology such as gear ratio, mechanical advantage, linkage, torque, and efficiency. Clear understanding of these terms is foundational.

- Practice Diagrams and Drawings

Many questions involve interpreting or creating diagrams of mechanisms. Practice sketching and analyzing diagrams to identify the type of mechanism and its function.

- Break Down Complex Problems

For multi-step problems, break them into smaller parts:

- Identify what is being asked.
- Determine what information is given.
- Decide which formulas or principles apply.
- Solve step-by-step, double-checking calculations.

- Use Real-World Contexts

Relate the questions to real-life mechanical systems or devices. This contextual understanding can help clarify abstract concepts.

Common Questions and Sample Answers in PLTW Unit 3

Below are typical questions students might encounter, along with detailed explanations and sample answers to clarify the reasoning process.

Question 1: Calculating Gear Ratios

If a gear train has a gear A with 20 teeth and gear B with 60 teeth, what is the gear ratio? How does this ratio affect the output speed?

Answer:

- Gear ratio = Number of teeth on driven gear / Number of teeth on driver gear = $60 / 20 = 3:1$.
- Effect on speed: The gear ratio of 3:1 means that for every 3 rotations of gear A (driver), gear B (driven) completes 1 rotation.
- Implication: The output speed decreases by a factor of 3, but torque increases proportionally.

Understanding this helps students see how gear ratios influence the trade-off between speed and torque in mechanical systems.

Question 2: Mechanical Advantage of a Lever

A lever is used to lift a 200 N weight. The effort applied at the other end of the lever is 50 N. What is the mechanical advantage of the lever?

Answer:

- Mechanical advantage (MA) = Load force / Effort force = $200 \text{ N} / 50 \text{ N} = 4$.
- Interpretation: The lever multiplies input force by a factor of 4, making it easier to lift heavy loads with less effort.

This reinforces the concept that simple machines like levers provide mechanical advantage, reducing the effort needed to perform work.

Question 3: Identifying the Type of Mechanism

Given a diagram of a linkage that converts rotary motion into oscillating motion, identify the type of mechanism.

Answer:

- Likely a four-bar linkage or oscillating crank, depending on specific linkages.
- Explanation: Four-bar linkages are common in converting rotary motion into linear or oscillating motion, used in various mechanical devices like windshield wipers or pistons.

Students should analyze the linkage diagram to identify pivot points, link lengths, and the motion path.

Question 4: Calculating Power in a Mechanical System

A motor applies a force of 50 N to move a mechanism at a velocity of 2 m/s. What is the power output?

Answer:

- Power (P) = Force (F) $\times$ Velocity (v) = 50 N $\times$ 2 m/s = 100 Watts.
- Significance: Power measures how quickly work is done; understanding this helps evaluate system efficiency.

Tips for Using Test Answers Effectively

- Use answers as a learning tool: Review explanations to deepen understanding.
- Cross-reference with class notes and textbooks to clarify misconceptions.
- Practice similar problems to reinforce concepts and improve problem-solving speed.
- Seek feedback from teachers or peers to ensure comprehension.

Additional Resources for Mastery

- PLTW Engineering Curriculum Guides: Offer detailed explanations and practice problems.
- Online Tutorials and Videos: Visual aids can clarify complex mechanisms.
- Practice Tests: Simulate exam conditions to build confidence and identify weak areas.

- Study Groups: Collaborate to discuss challenging concepts and solutions.

Final Thoughts: Preparing Beyond Just Answers

While PLTW Unit 3 test answers are helpful for review, the ultimate goal is to develop a deep understanding of mechanical systems and engineering principles. Focus on mastering core concepts, practicing problem-solving, and applying knowledge to real-world scenarios. This approach not only prepares you for the test but also builds a solid foundation for future engineering endeavors.

Remember, success in engineering is about curiosity, critical thinking, and continuous learning. Use these answers as stepping stones to enhance your understanding and to become a more confident and capable problem solver.

Happy studying, and best of luck on your PLTW Unit 3 assessments!

Question What are the key topics covered in the PLTW Unit 3 test? The PLTW Unit 3 test typically covers topics such as mechanical systems, mechanisms, gears, levers, and structural analysis, focusing on understanding how different components work together in engineering designs. How can I effectively prepare for the PLTW Unit 3 test? To prepare effectively, review your class notes, complete all practice questions, understand key concepts like gear ratios and types of mechanisms, and utilize any available study guides or online resources provided by your instructor. Are there specific formulas I should memorize for the PLTW Unit 3 test? Yes, formulas related to gear ratios, mechanical advantage, and force calculations are essential. Make sure to understand how to apply these formulas in different scenarios presented in the test. Where can I find official PLTW Unit 3 test answers for practice? Official answers are typically provided by your instructor or through authorized course materials. It's best to use these resources for accurate practice, and avoid seeking unverified answer keys online. What types of questions are commonly on the PLTW Unit 3 test? Common questions include multiple-choice, short answer, and diagram-based problems that assess understanding of mechanisms, gear systems, structural analysis, and applying engineering principles. How can I improve my understanding of mechanical systems for the PLTW Unit 3 test? Improve understanding by conducting hands-on activities, drawing and analyzing diagrams, practicing with real-world examples, and discussing concepts with classmates or teachers to reinforce learning.

Related keywords: PLTW Unit 3 test, PLTW Unit 3 answers, PLTW unit 3 solutions, PLTW engineering test answers, Project Lead The Way Unit 3, PLTW Unit 3 quiz, PLTW Unit 3 practice questions, PLTW Unit 3 exam tips, PLTW engineering coursework, PLTW Unit 3 review

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](#)