

benchmark test questions for second grade math PDF

Benchmark test questions for second grade math are essential tools for educators and parents aiming to assess the foundational math skills of young learners. These tests serve as a valuable measure to determine students' understanding of key concepts, identify areas needing improvement, and inform instruction to ensure mastery of essential skills. In this comprehensive guide, we will explore the importance of benchmark testing for second graders, the types of questions typically included, tips for preparing students, and sample questions to help educators craft effective assessments.

The Importance of Benchmark Test Questions for Second Grade Math

Why Benchmark Testing Matters

Benchmark tests are periodic assessments administered throughout the school year to monitor student progress. For second graders, these assessments are particularly critical because they:

- Provide a clear picture of students' grasp of fundamental math concepts.
- Help identify students who may need additional support or intervention.
- Assist teachers in adjusting instruction to better meet students' needs.
- Build confidence as students see their progress over time.
- Prepare students for future standardized tests by familiarizing them with question formats.

Aligning with Curriculum Standards

Benchmark questions are designed to align with state and national curriculum standards for second grade mathematics. They typically cover key domains such as addition and subtraction, place value, measurement, geometry, and basic data analysis. Ensuring questions are aligned helps maintain consistency in assessment and provides meaningful data on student achievement.

Core Content Areas in Second Grade Math Benchmark Tests

Addition and Subtraction

Understanding basic addition and subtraction is foundational for second graders. Questions often focus on:

- Solving addition and subtraction problems within 100.
- Understanding the relationship between addition and subtraction.
- Using strategies such as counting, decomposing, and number lines.

Place Value

Students should demonstrate understanding of:

- Recognizing hundreds, tens, and ones.
- Reading and writing numbers up to 999.
- Comparing and ordering numbers.

Measurement and Data

Questions may assess:

- Understanding concepts of length, weight, and volume.
- Using standard and non-standard units for measurement.
- Collecting, organizing, and interpreting data using charts or graphs.

Geometry

Second graders should be familiar with basic geometric concepts such as:

- Identifying and describing 2D shapes (circles, squares, rectangles, triangles).
- Recognizing 3D shapes (cubes, spheres, cylinders).
- Understanding symmetry and spatial relationships.

Number Patterns and Basic Operations

Assessment questions might include:

- Completing number patterns.
- Understanding even and odd numbers.
- Introducing multiplication as repeated addition.

Sample Benchmark Test Questions for Second Grade Math

Addition and Subtraction

- What is $45 + 32$?
- Subtract 17 from 50. What is the answer?
- There are 24 apples in a basket. If you give away 9 apples, how many are left?

Place Value

- Write the number three hundred twenty-five in digits.
- Which number is greater: 248 or 284?
- What is the value of the digit 5 in the number 652?

Measurement and Data

- Measure the length of a pencil using a ruler. Is it 5 inches or 8 inches?
- Look at the bar graph. How many students like apples the most?
- Which container holds more water: a small cup or a large bowl?

Geometry

- Identify the shape that has four equal sides and four right angles.
- How many sides does a triangle have?
- Is a sphere a 2D or 3D shape?

Number Patterns and Basic Operations

- What is the next number in the pattern: 2, 4, 6, 8, ____?
- Are the numbers 11 and 13 even or odd?
- If you have 3 groups of 4 apples, how many apples do you have in total?

Tips for Preparing Second Grade Students for Benchmark Tests

Create a Practice Routine

Regular practice helps students become familiar with the types of questions they will encounter. Use sample questions, flashcards, and online resources to simulate test conditions.

Focus on Conceptual Understanding

Encourage students to understand the "why" behind math concepts rather than rote memorization. This deep understanding promotes better problem-solving skills.

Use Visual Aids and Manipulatives

Tools such as counters, base-ten blocks, and geometric shape models aid in grasping abstract concepts and making learning engaging.

Incorporate Real-Life Contexts

Apply math problems to everyday situations, like shopping, cooking, or playing, to make learning relevant and interesting.

Build Test-Taking Confidence

Teach students strategies such as reading questions carefully, eliminating obviously wrong answers, and managing time effectively during the test.

Conclusion

Benchmark test questions for second grade math are a vital component of early education, providing a snapshot of student understanding and guiding instructional decisions. By focusing on key content areas—addition and subtraction, place value, measurement, geometry, and number patterns—educators can develop assessments that accurately gauge students' skills. Incorporating sample questions, practicing regularly, and fostering a supportive learning environment will help second graders succeed and build a strong foundation for future math learning. Whether used for formative assessment or standardized testing preparation, effective benchmark questions empower students and teachers alike to achieve academic growth and confidence in math.

Benchmark Test Questions for Second Grade Math: An In-Depth Analysis

In the realm of elementary education, particularly in second grade, math serves as a foundational pillar that shapes a student's future academic trajectory. As educators and curriculum developers strive to gauge student understanding, benchmark test questions for second grade math have become essential tools. These assessments not only measure mastery but also inform instructional strategies, identify gaps in knowledge, and prepare students for more advanced concepts. This article delves into the significance of benchmark test questions for second grade math, examining their structure, common content areas, developmental considerations, and best practices for crafting effective assessments.

The Importance of Benchmark Test Questions in Second Grade Math

Benchmark assessments are standardized or teacher-created tests administered periodically to evaluate students' progress relative to grade-level expectations. For second graders, these assessments serve several critical functions:

- Monitoring Learning Progress: They help teachers track individual and class-wide understanding of key concepts.
- Informing Instruction: Results guide targeted interventions and instructional adjustments.
- Preparing for Standardized Tests: They familiarize students with test formats and question types they will encounter in later grades.
- Building Student Confidence: Early success in assessments fosters motivation and a growth mindset.

Given these roles, the design of benchmark test questions must reflect developmental appropriateness, clarity, and alignment with curriculum standards.

Core Content Areas Covered by Second Grade Math Benchmark Tests

Second grade math benchmarks typically encompass several foundational domains aligned with Common Core State Standards (CCSS) and other state frameworks. These domains include:

1. Number Operations and Place Value

- Understanding the place value of digits within two- and three-digit numbers
- Comparing and ordering numbers
- Addition and subtraction within 100
- Recognizing and working with even and odd numbers

2. Addition and Subtraction Strategies

- Using mental math and strategies such as counting on or back
- Solving word problems involving addition and subtraction
- Understanding the inverse relationship between addition and subtraction

3. Measurement and Data

- Comparing lengths using non-standard and standard units
- Understanding concepts of time (hours and minutes)
- Collecting, organizing, and interpreting data (e.g., tally charts, bar graphs)

4. Geometry

- Recognizing and describing two-dimensional shapes (circles, triangles, rectangles, squares)
- Understanding basic attributes of shapes
- Partitioning shapes into equal parts (halves, thirds, quarters)

5. Fractions (Foundational Concepts)

- Recognizing simple fractions such as $\frac{1}{2}$ and $\frac{1}{3}$ as parts of a whole
- Identifying fractional parts of shapes

Design Principles for Benchmark Test Questions in Second Grade Math

Creating effective benchmark test questions requires careful consideration of developmental appropriateness, clarity, and instructional relevance. Several principles guide the development process:

1. Clarity and Simplicity

Questions should be straightforward, avoiding ambiguous language or complex sentence structures that might confuse young learners. Clear instructions and visual cues help ensure students understand what is being asked.

2. Alignment with Standards

Questions must directly reflect the specific skills and concepts outlined in grade-level standards to accurately measure mastery.

3. Use of Visuals and Contexts

Incorporating visual aids, such as pictures of shapes or number lines, and real-world contexts makes questions more engaging and accessible.

4. Varied Question Formats

A mix of multiple-choice, fill-in-the-blank, matching, and short-answer questions caters to different learning styles and assessment purposes.

5. Scaffolding and Differentiation

Including questions that range from simple recall to applied problem-solving allows assessment of various cognitive levels.

Sample Benchmark Test Questions for Second Grade Math

Below are representative questions across different content areas, illustrating best practices in assessment design.

Number and Operations

- Multiple Choice:

What is $47 + 26$?

- a) 63
- b) 73
- c) 83
- d) 93

- Fill in the Blank:

Write the number that comes after 58.

Word Problems

- Problem:

Sarah has 15 apples. She gives 7 to her friend. How many apples does she have left?

Answer: _____

Measurement and Data

- Question:

Which object is longer: a pencil or a crayon?

(Include images for visual support)

- Data Interpretation:

Look at the bar graph below showing the number of pets students have. How many students have cats?

(Provide a simple bar graph with labeled data)

Geometry

- Shape Identification:

Circle the shape that has four equal sides.

(Images of shapes: square, triangle, circle, rectangle)

- Partitioning Shapes:

Color in half of the circle.

(Shape diagram provided)

Fractions

- Recognizing Fractions:

Which shape shows $\frac{1}{2}$ of the pizza?

(Image of a circle divided into halves)

Developmental Considerations and Common Challenges

Second graders are at a developmental stage where attention spans are still maturing, and abstract reasoning skills are developing. Benchmark questions must accommodate these factors to accurately assess understanding without causing undue frustration.

Challenges include:

- Literal Interpretation: Young students may interpret questions literally, so instructions should be explicit.
- Limited Vocabulary: Use of age-appropriate language is essential to avoid misinterpretation.
- Visual Dependence: Many second graders benefit from visual cues, so assessments should incorporate relevant images and diagrams.
- Cognitive Load: Questions should balance complexity with clarity, avoiding overly multi-step problems that can overwhelm learners.

Strategies to mitigate challenges:

- Incorporate familiar contexts and objects.
- Use consistent question formats.
- Provide examples or practice items as needed.
- Include oral or manipulatives-based assessments alongside written questions for comprehensive evaluation.

Best Practices for Educators and Test Developers

To ensure benchmark assessments serve their intended purpose effectively, educators and developers should consider the following best practices:

- Align with Curriculum: Ensure questions reflect the content and skills emphasized during instruction.
- Prioritize Clarity: Use straightforward language and visuals to minimize misunderstanding.
- Balance Difficulty: Include a range of question difficulties to gauge different levels of mastery.
- Incorporate Real-Life Contexts: Use everyday scenarios to make questions relevant and engaging.
- Use Multiple Formats: Combine multiple-choice, open-ended, and matching questions to assess

various cognitive skills.

- Provide Clear Instructions: Explicit directions help students understand what is expected.
- Pilot Test Questions: Before full deployment, test questions with a small group to identify ambiguities or difficulties.

Conclusion: The Significance of Thoughtfully Designed Benchmark Questions

Benchmark test questions for second grade math are more than mere assessment tools; they are vital instruments that shape early mathematical understanding and confidence. Well-crafted questions that are developmentally appropriate, aligned with standards, and engaging can illuminate students' strengths and areas for growth. As education continues to evolve, so too must the assessments that measure its outcomes. By prioritizing clarity, relevance, and variety in question design, educators can foster an assessment environment that not only evaluates but also inspires young learners on their mathematical journey.

In summary, developing effective benchmark test questions for second grade math requires a nuanced understanding of developmental stages, curriculum standards, and best assessment practices. When thoughtfully designed, these questions serve as powerful tools to support student growth, inform instruction, and lay a strong foundation for future mathematical success.

QuestionAnswer What is $7 + 5$? 12 If you have 3 apples and buy 4 more, how many apples do you have in total? 7 apples Which number comes after 29? 30 How many sides does a triangle have? 3 sides What is 10 minus 4? 6

Related keywords: second grade math test, elementary math assessment, grade 2 math practice, math quiz for second graders, second grade math exercises, grade 2 math review, math skills assessment, second grade math worksheets, basic math questions for grade 2, elementary school math tests

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.

Question 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. **Answer** 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](#)