

Brian trubshaw test pilot Manual

brian trubshaw test pilot

Brian Trubshaw was a pioneering figure in the field of aviation testing and a key contributor to the development of some of the most advanced aircraft of his era. His career as a test pilot was marked by remarkable achievements, technical expertise, and a dedication to pushing the boundaries of aerial performance. This article explores the life, career, and legacy of Brian Trubshaw as a test pilot, shedding light on his contributions to aerospace engineering and aviation history.

Early Life and Entry into Aviation

Background and Education

Brian Trubshaw was born in 1924 in England. From a young age, he exhibited a keen interest in aircraft and flying, nurtured by the burgeoning aeronautical industry in post-war Britain. He pursued engineering studies, which laid a solid foundation for his future work in aircraft development and testing.

Initial Experience in Aviation

Before becoming a test pilot, Trubshaw gained experience as a pilot through military service and civilian flying. His early flying days involved training in various aircraft, which helped him develop the skills necessary for the demanding role of a test pilot.

Transition to Test Pilot Career

Joining the Royal Aircraft Establishment

Trubshaw's entry into the world of aircraft testing was facilitated by his employment at the Royal Aircraft Establishment (RAE), a UK government research organization responsible for aircraft development and testing. His role involved conducting flight tests and providing critical data for aircraft improvements.

Move to Commercial Aircraft Development

Later, Trubshaw transitioned to working with aircraft manufacturers, where he participated in test programs for several pioneering aircraft. His expertise was increasingly recognized, leading to more prominent roles in high-profile projects.

Role in the Development of the Concorde

Introduction to the Concorde Project

One of the most significant chapters of Trubshaw's career was his involvement in the development and testing of the Concorde supersonic passenger aircraft. As a key test pilot, he played a vital role in validating the aircraft's capabilities and safety.

Test Flight Milestones

Trubshaw was instrumental in conducting the first transatlantic test flights of Concorde, which demonstrated the aircraft's ability to cruise at Mach 2.0. These flights marked historic milestones in aviation history and required exceptional skill and precision.

- First flight of Concorde 001 on March 2, 1969
- First supersonic flight in October 1973
- Transatlantic crossings confirming speed and range

Challenges Faced During Testing

Testing Concorde involved overcoming numerous technical and logistical challenges:

- Managing high-speed aerodynamic stability
- Ensuring passenger safety at supersonic speeds
- Addressing noise and environmental concerns
- Refining the aircraft's control systems for supersonic flight

Technical Skills and Qualities as a Test Pilot

Expertise in Aerodynamics and Aircraft Systems

Trubshaw's background in engineering and his flying experience equipped him with an in-depth understanding of aircraft systems, aerodynamics, and flight dynamics. This knowledge was crucial during complex test flights, especially at supersonic speeds.

Precision and Calm Under Pressure

Test pilots operate in high-stakes environments, often pushing aircraft beyond normal operational limits. Trubshaw's ability to remain calm and make split-second decisions was instrumental in the

success of his missions.

Collaboration with Engineers and Designers

Effective communication between pilots and engineers is vital during aircraft testing. Trubshaw worked closely with the design teams, providing feedback that led to iterative improvements and safer aircraft.

Legacy and Impact on Aviation

Pioneering Supersonic Travel

Brian Trubshaw's efforts contributed directly to the realization of commercial supersonic travel with Concorde. His test flights validated the aircraft's design and performance, paving the way for passenger service.

Advancements in Test Pilot Techniques

His career helped refine testing methodologies, including safety protocols, data collection, and aircraft handling procedures at high speeds.

Inspiration for Future Generations

As a trailblazer, Trubshaw inspired many aspiring pilots and engineers. His dedication exemplifies the qualities necessary for innovation in aerospace.

Later Life and Recognition

Retirement and Legacy Projects

After retiring from active test flying, Trubshaw remained involved in aviation through mentorship, consultancy, and public speaking. His insights continued to influence aircraft testing programs.

Honors and Awards

Trubshaw received numerous accolades, including:

- Fellowship of the Royal Aeronautical Society
- Recognition from aerospace institutions for his contributions
- Legacy in aviation history as a pioneering test pilot

Conclusion

Brian Trubshaw's career as a test pilot embodies the spirit of innovation, bravery, and technical excellence. His critical role in the development of Concorde not only advanced aviation technology but also demonstrated the importance of meticulous testing and pilot expertise in achieving groundbreaking milestones. Today, Trubshaw remains a celebrated figure in the annals of aerospace history, inspiring future generations to continue pushing the frontiers of flight.

References

- Royal Aeronautical Society Archives
- Concorde Official History
- Interviews and Biographies of Brian Trubshaw
- Aviation Safety and Test Flight Manuals

Brian Trubshaw Test Pilot: A Pioneering Force in Aeronautical Innovation

In the annals of aviation history, few names resonate with the same pioneering spirit and technical mastery as Brian Trubshaw Test Pilot. Renowned for his instrumental role in testing some of the most groundbreaking aircraft of the 20th century, Trubshaw's career exemplifies the blend of daring exploration and meticulous engineering that has driven aviation forward. This article offers an in-depth exploration into his life, career, key test programs, and lasting legacy, providing a comprehensive review suitable for aviation enthusiasts, industry professionals, and scholars alike.

Early Life and Path to Aviation

Born in 1928 in Birkenhead, England, Brian Trubshaw's fascination with flight was evident from a young age. Growing up amidst the post-war technological renaissance, he was captivated by aircraft and the promise of human flight. His early education in engineering laid the groundwork for a career that would intertwine technical expertise with adventurous spirit.

After completing his education, Trubshaw joined the Royal Air Force (RAF) as a pilot, where he honed his flying skills and developed a keen interest in aircraft performance and aerodynamics. His early military service provided a foundation of discipline and hands-on experience, making him an ideal candidate for the emerging field of aircraft testing.

Transition to Civil Aviation and the Testing Realm

Following his military career, Trubshaw transitioned into civil aviation, joining the British aircraft manufacturer Vickers-Armstrongs (later part of BAC-British Aircraft Corporation). His technical

acumen and flying prowess quickly distinguished him, leading to his recruitment as a test pilot.

Joining the Testing Program

In the late 1950s, the aviation industry was entering a new era of supersonic flight and technological complexity. Trubshaw's role became pivotal in testing aircraft that would push the boundaries of speed, altitude, and aircraft design. His reputation for calmness under pressure and precise handling made him a trusted figure in the high-stakes world of experimental flight.

The Concorde Project: A Historic Milestone

Perhaps the most celebrated chapter in Brian Trubshaw's career was his involvement with the Concorde supersonic passenger jet. As the chief test pilot for Concorde's initial test flights, Trubshaw played a critical role in turning a complex engineering concept into a tangible, operational aircraft.

Leading the Testing Efforts

In 1969, Trubshaw made aviation history by piloting Concorde's first flight from BAC's Filton airfield in Bristol. His meticulous preparation and deep understanding of the aircraft's systems contributed to the successful maiden voyage, setting the stage for a series of rigorous test flights that would validate Concorde's design and capabilities.

His expertise was essential during critical phases such as:

- High-speed handling tests to assess supersonic stability
- Landing and takeoff performance evaluations at various weights
- Supersonic cruise testing to analyze aerodynamic behavior

Challenges Overcome

The testing phase was fraught with challenges, including:

- Managing the aircraft's unique delta wing configuration at high speeds
- Ensuring passenger safety amid unprecedented speeds and altitudes
- Addressing noise, fuel efficiency, and environmental concerns

Trubshaw's calm demeanor and technical insight helped navigate these complexities, ultimately contributing to the aircraft's certification and commercial deployment.

Other Notable Test Programs

While Concorde remains Trubshaw's most iconic project, his career encompassed a variety of other significant test programs:

Vickers Valiant Bomber

- Trubshaw participated in testing the Valiant, Britain's first operational V-bomber, focusing on high-altitude performance and stability.

British Aerospace Hawk

- He contributed to the early flight testing of the Hawk jet trainer, helping refine maneuverability and handling characteristics crucial for pilot training.

Experimental Research Aircraft

- Trubshaw was involved in testing various experimental aircraft designed to explore new aerodynamics and propulsion technologies.

Test Pilot Philosophy and Techniques

Brian Trubshaw's approach to aircraft testing combined rigorous scientific methodology with an innate sense of adventure. His philosophy centered around:

- Preparation: Thorough analysis of aircraft systems and potential failure modes
- Safety: Prioritizing safety while maintaining a willingness to push aircraft to their limits
- Data Collection: Precise instrumentation and observation to inform engineering decisions
- Communication: Clear, concise communication with engineers and flight teams to adapt to evolving conditions

His techniques often involved incremental testing, gradually increasing aircraft performance parameters to ensure stability and safety at each stage.

Recognition and Legacy

Trubshaw's contributions earned him numerous accolades, including recognition from the Royal

Aeronautical Society and industry awards for innovation and safety. His work not only advanced aircraft design but also set new standards for test pilot professionalism.

His legacy endures through:

- Contributions to the success of Concorde, which remains an icon of engineering achievement
- Mentorship of subsequent generations of test pilots, emphasizing safety, precision, and innovation
- Influence on aviation safety standards, derived from his meticulous testing philosophy

Impact on Aviation and Modern Test Piloting

Brian Trubshaw's career exemplifies the critical role of test pilots in transforming theoretical aircraft concepts into operational realities. His work demonstrated that successful aircraft development hinges not only on engineering but also on the skill, judgment, and courage of those who fly the prototypes.

Modern test pilots continue to build on his legacy, employing advanced simulation, telemetry, and safety protocols, but the core principles of thorough preparation, safety consciousness, and adaptive problem-solving remain rooted in the pioneering spirit exemplified by Trubshaw.

Conclusion

The story of Brian Trubshaw Test Pilot is a testament to the daring and meticulous craftsmanship that has propelled aviation to new heights. From pioneering supersonic flight with Concorde to shaping modern aircraft testing standards, his influence is indelible. As aviation technology continues to evolve, the foundational work of pioneers like Trubshaw remains a guiding light, inspiring future generations to explore the boundaries of human flight with courage, curiosity, and precision.

His legacy reminds us that behind every groundbreaking aircraft is a pilot willing to venture into the unknown, driven by a passion for innovation and an unwavering commitment to safety. In the history of aviation, Brian Trubshaw's name stands as a symbol of exploration, excellence, and the relentless pursuit of progress.

QuestionAnswer Who is Brian Trubshaw and what is he known for in the field of aviation? Brian Trubshaw was a renowned British test pilot best known for being the first person to fly the Concorde supersonic passenger aircraft in 1969, playing a pivotal role in its development. What were some of Brian Trubshaw's significant contributions to aviation testing? Brian Trubshaw's significant contributions include leading the test flights for the Concorde, helping to validate its design and

performance, and advancing supersonic flight technology. How did Brian Trubshaw influence modern aerospace testing practices? Through his extensive experience with supersonic aircraft, Brian Trubshaw helped establish rigorous testing protocols and safety standards that continue to influence aerospace testing today. What challenges did Brian Trubshaw face during his test pilot career, especially with Concorde? Trubshaw faced numerous challenges including managing the complexities of supersonic flight, ensuring safety during test flights, and overcoming technical uncertainties associated with pioneering aircraft like Concorde. What is Brian Trubshaw's legacy in the aerospace community? Brian Trubshaw's legacy lies in his pioneering role in supersonic flight testing, his contributions to the successful certification of Concorde, and inspiring future generations of test pilots and aerospace engineers.

Related keywords: Brian Trubshaw, test pilot, Concorde, aviation pioneer, supersonic aircraft, aircraft testing, aerospace engineering, British Airways, flight testing, aviation history

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](#)