

Microsoft

Microsoft® Technology Associate

STUDENT STUDY GUIDE

preparing for MTA certification

Microsoft
Technology Associate

EXAM 98-379
Software Testing Fundamentals

PREPARING FOR MTA CERTIFICATION

MICROSOFT TECHNOLOGY ASSOCIATE (MTA) STUDENT STUDY GUIDE FOR DEVELOPERS

98-379 Software Testing Fundamentals

AUTHORS

Tim McMichael Tim has been a high school computer science teacher for the past 13 years. He currently teaches computer science at Estrella Mountain Community College in Avondale, Arizona. Prior to teaching, Tim worked for several years as a database application developer. Tim is also the author of the *98-361: Software Development Fundamentals*, *92-362: Windows® Development Fundamentals* and *98-373 Mobile Development Fundamentals* Exam Review Kits in the MTA Certification Exam Review Kit series.

Patricia Phillips Patricia taught computer science for 20 years in Janesville, Wisconsin. She served on Microsoft's National K-12 Faculty Advisory Board and edited the High School Faculty Connection website for technology teachers. She has worked in a variety of roles related to K-12 curriculum development and pilot programs, including Expression Studio web design, XNA game development, and MTA Certification Exam Review Kits. She is currently the editor of the Computer Science Teachers Association newsletter, the *Voice*.

This content is only for use by or provision to students for their personal use.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

© 2013 Microsoft Corporation. All Rights Reserved. This content is provided "as-is" and Microsoft makes no warranties, express or implied.

CONTENTS

Introduction	v
Career Planning	vi
Exploring Job Roles	vii
Value of Certification	ix

98-379: SOFTWARE TESTING FUNDAMENTALS

OBJECTIVE 1

Describe Testing Fundamentals	3
--	----------

1.1 - 1.2 Describe Software Testing; Describe Software and Hardware Components	5
--	---

1.3 - 1.4 Describe Fundamentals of Programming; Describe Application Lifecycle Management	7
---	---

OBJECTIVE 2

Describe Testing Methodology	9
---	----------

2.1 Describe Testing Techniques	11
---------------------------------------	----

2.2 Describe Testing Levels	13
-----------------------------------	----

2.3 Describe Testing Types	15
----------------------------------	----

OBJECTIVE 3	Create Software Tests	17
	3.1 - 3.2 Describe User-Centric Testing; Describe Software Testability.....	19
	3.3 Create Test Plan Components.....	21
	3.4 Describe Feature Tests.....	23
	3.5 Define Appropriately Scoped Test Cases.....	25
OBJECTIVE 4	Manage Software Testing Projects	27
	4.1 Describe Testing Milestones.....	29
	4.2 Describe the Agile Process.....	31
	4.3 Work with Distributed Teams.....	33
	4.4 Define Test Reports.....	35
OBJECTIVE 5	Work with Bugs	37
	5.1 Detect Software Defects.....	39
	5.2 Log Bugs.....	41
	5.3 Manage Bugs.....	43
OBJECTIVE 6	Automate Software Testing	45
	6.1 Describe Test Automation.....	47
	6.2 Define Test Automation Strategies.....	49
	6.3 Write Automation Tests.....	51
	6.4 Manage Test Scripts.....	53

INTRODUCTION

MTA validates building-block technology concepts and helps students explore, discover and pursue successful careers in Information Technology (IT) in an exciting and rewarding way! As the first step in the Microsoft Technology Certification Series, this new, entry-level certification provides students with confidence, credibility, and differentiation.

Explore IT career options without committing a lot of time and resources MTA exams validate the core technology knowledge that is in demand today by businesses around the world. Whether you want to explore becoming a network administrator, software engineer, web developer, or database analyst, MTA gets you started on the right path.

Prepare to compete A little investment in IT can go a long way in today's job market. Becoming MTA certified helps you build a solid foundation to prepare for intermediate technology studies and for Microsoft Certified Technology Specialist (MCTS) certifications. It can also help you compete on college admissions and jumpstart your IT career planning!

Empower yourself As the first step toward becoming an MCTS, MTA shows your commitment to technology while connecting you with a community of more than five million Microsoft Certified Professionals. Learn from them and show them what you know by becoming MTA certified!

This MTA Student Study Guide serves as a study tool to help students prepare for their MTA certification exam. Students are challenged with real-life situations for each of the major topics covered in the exam. Although successful completion of the study guide exercises does not guarantee that you will pass your MTA exam, it is an excellent way to gauge your readiness to take the exam and build confidence that you know your stuff on exam day.

I wish you all the best as you prepare for a successful career in technology!

Victoria Pohto

Victoria Pohto
MTA Product Marketing Manager



START

CAREER PLANNING

Most IT solutions or infrastructure built on Microsoft technologies require proficiency with one or all of the following products, often referred to as “The Microsoft Stack.”

- Microsoft Windows® Server® as the data center or development platform
- Microsoft SQL Server® as the data and business intelligence (BI) platform
- Microsoft Visual Studio® as the suite of application life-cycle management tools

MTA is the starting point of Microsoft technology certifications, providing aspiring technologists with the fundamental knowledge essential to succeed with continued studies and a successful career with technology.

Preparing for and becoming MTA certified helps you explore a variety of career paths in technology without investing a lot of time and money in a specialized career path. When you find a path that is right for you, Microsoft learning products and certification can help you prepare and guide your longer-term career planning.

If you already know that you want to start building a career in technology, MTA preparation and certification is the recommended entry point. Becoming MTA certified shows that you have a firm working

knowledge of the fundamental IT concepts critical for success with intermediate learning and certifications such as Microsoft Certified Technology Specialist (MCTS). Moreover, Microsoft certifications demonstrate an individual’s commitment of self-investment and confidence to take his or her knowledge and skills to the next level with an industry-recognized credential.

MTA is the first step toward your IT career goal and can help differentiate you for an internship or to college admissions committees. As you prepare for your first job focusing on technology, be sure that you are equipped with an MCTS credential—the intermediate level certification that validates Microsoft product and technology skills.

A visual mapping of the MTA Certification paths can be found at <http://www.microsoft.com/learning/en-us/mta-certification.aspx> and will show you the MTA exams that are recommended prior to taking on some of Microsoft’s intermediate technology certification, MCTS. MTA is the first step to Microsoft® Technology Certification Series. MTA is a recommended but not required pre-requisite to MCTS exams. One certification is earned for each exam passed. Free Student Study Guides are available for download at www.certipoint.com/mta.



EXPLORING JOB ROLES

Choosing a career path is a big decision and it's not always easy, but you're not alone! Microsoft created a career site to help students understand the options and possibilities of pursuing a career in IT. The site also connects you with learning resources, student techie communities, and much more to help you prepare for a career in technology.

To chart your career with Microsoft technology, visit <http://www.microsoft.com/learning/en-us/default.aspx>.

DATABASE ADMINISTRATOR

As a database administrator, you are in charge of important databases that span multiple platforms and environments. You are a strong team player who thrives in a fast-paced environment. You build complex, highly scalable databases that meet business needs and security requirements. You are an expert in optimizing, maintaining, and troubleshooting databases, but also in designing archival, data distribution, and high-availability solutions.

SERVER ADMINISTRATOR

As a server administrator, you are in charge of implementing and managing some of the most important technology in your organization—the servers. You use extensive monitoring and profiling tools to manage the network and tune systems so they perform at optimal levels. You are an expert in Active Directory®, and you have an in-depth understanding of network protocols, and file and directory security.

COMPUTER SUPPORT TECHNICIAN

Consider starting your IT career by becoming a consumer support technician. You don't need any formal work experience, but a company might require that you know how to install, administer, and troubleshoot operating systems in a home network environment that has desktop computers, laptops, and printers. As a consumer support technician, you'll also handle network, virus, malicious software, and hardware support issues. You'll typically find this position in small to medium-sized organizations.



EXPLORING JOB ROLES

WEB DEVELOPER

As a web developer, you are an expert in using the dynamic programming tools and languages that fuel the web. You might work independently or be part of a team that builds and integrates interactive web sites, applications, and services for both internal and public sites. Your role is to make it work, which means developing web applications and testing them on various browsers, enhancing and modifying them as necessary to ensure the best experience for the user. As a web developer, you might also architect websites, design data-driven applications, and find efficient client-server solutions. You must have an in-depth understanding of the software development life cycle and be able to communicate project status, issues, and resolutions.

WINDOWS DEVELOPER

As a Windows client developer, knowing how to optimize Windows code and track bugs is a given. But you also know how to use Microsoft Visual Studio® and the Microsoft .NET framework to design, develop, test, and deploy Windows-based applications that run on both corporate servers and desktop computers. Your key talents include understanding multiple Windows application models and n-tier applications, and knowing how to work with object-oriented programming,

algorithms, data structures, and multithreading. Windows developers have an in-depth understanding of software engineering principles, software life cycles, and security principles.

Additional Online Resources for New Developers:

<http://msdn.microsoft.com/beginner>

<http://msdn.microsoft.com/rampup>

IMAGINE CUP

The Imagine Cup is the world's premier student technology competition where students from around the world can learn new skills, make new friends, and change the world. Competitions include Software Design, Embedded Development, Game Design, Digital Media and Windows Phone 7. The brightest young minds harness the power of technology to take on the world's toughest problems.

www.imaginecup.com



VALUE OF CERTIFICATION

Technology plays a role in virtually everything we do. In the 20-plus years since Microsoft has been certifying people on its products and technologies, millions of people have gained the knowledge, expertise, and credentials to enhance their careers, optimize business solutions, and create innovation within just about every business and social sector imaginable. Today's Information Technology (IT) hiring managers are more often using professional credentials, such as Microsoft certification, to identify properly skilled IT candidates. Certification becomes a way to easily differentiate qualified candidates in a sea of resumes.

The job outlook for IT professionals, as reported in a study prepared by the U.S. Department of Labor's Bureau of Labor Statistics (BLS), is positive! The BLS indicates an increase that will be "faster than the average for all occupations through 2014" for Computer Support Specialists, Systems Engineers, Database Administrators, and Computer Software Engineers. One significant message resulting from this study is that information and communications

technology (ICT) skills are the entry ticket to the job market, regardless of the country, industry, or job function. Information Technology is clearly an area worth investing time, resources, and education in – and technology certification is a key part of the education process, validating product and technology expertise as a result of their learning experiences.

Microsoft IT Certifications provide objective validation of the ability to perform critical IT functions successfully for worldwide IT professionals, developers, and information workers. Microsoft certifications represent a rich and varied spectrum of knowledge, job roles, and responsibilities. Further, earning a specific certification provides objective validation of the candidate's ability to perform critical IT functions successfully. Embraced by industry professionals worldwide, Microsoft certification remains one of the most effective ways to help reach long-term career goals.



SHINE

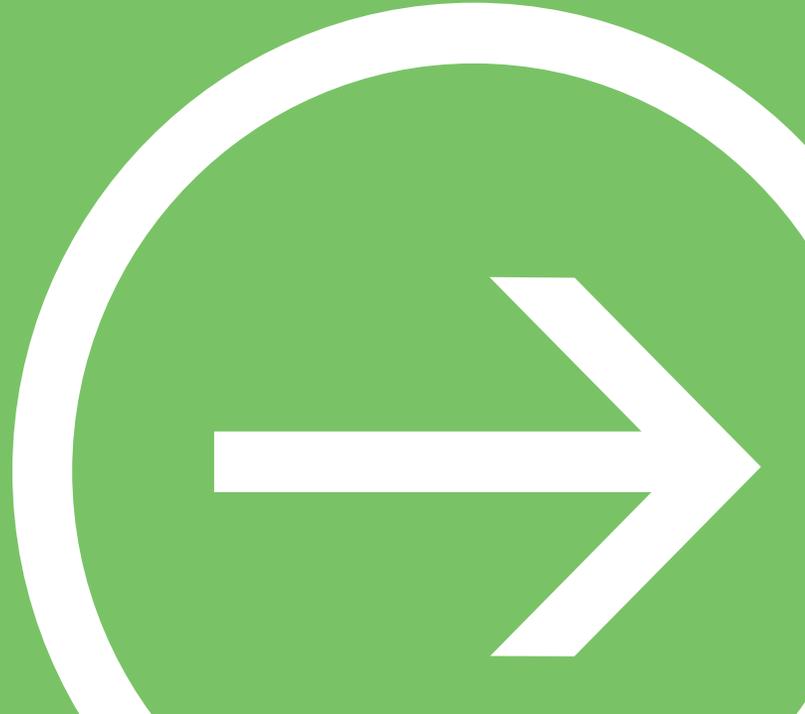


CAREER



COMPETE

MTA 98-379
SOFTWARE TESTING
FUNDAMENTALS



OBJECTIVE 1

Describe Testing Fundamentals

IN THIS CHAPTER

- 1.1 - 1.2 Describe Software Testing; Describe Software and Hardware Components
- 1.3 - 1.4 Describe Fundamentals of Programming; Describe Application Lifecycle Management



OBJECTIVE

describe testing fundamentals 1.1-1.2

DESCRIBE SOFTWARE TESTING; DESCRIBE SOFTWARE AND HARDWARE COMPONENTS

SCENARIO: Sanjay Patel is an intern at Contoso Pharmaceuticals and has approached his manager, Ann Beebe, with an idea for a mobile application. Sanjay wants to make an app that helps cancer patients manage their doctor appointments, treatments, and medications, as well as easily access medical information from respected websites and publications. He also wants the app to sync with a desktop application for patients to use while at home.

With Ann's help, Sanjay is going to present the idea to a project manager for consideration. Ann suggests that Sanjay think about how to integrate testing into the development process, and incorporate that philosophy into his presentation.

- 1. Which of the following is NOT a purpose of testing?**
 - a. To ensure that an application meets its design requirements
 - b. To identify bugs
 - c. To get feedback from end users about design ideas
- 2. Which of the following is a metric of software quality?**
 - a. Defect density
 - b. The number of lines of code
 - c. Display resolution
- 3. The desktop application will need to sync with the mobile app. Which type of software may be necessary for the application to communicate with a connected mobile device?**
 - a. A device driver
 - b. A dependency driver
 - c. A mobile driver

HINT 

A metric is a standard of measurement. For example, one metric of physical fitness is an individual's weight.

Answers

1. Which of the following is NOT a purpose of testing?
 - c. **To get feedback from end users about design ideas.** While this may be an important part of the design and development process, it is not a purpose of testing.
2. Which of the following is a metric of software quality?
 - c. **Defect density.** This refers to the number of confirmed defects or bugs detected in the project divided by the size of the project (i.e., the number of bugs per line of code).
3. The desktop application will need to sync with the mobile app. Which type of software may be necessary for the application to communicate with a connected mobile device?
 - a. **A device driver.** Device drivers permit a computer system to communicate with devices.

Essential details

- **Testing** is the process of examining an application to ensure it fulfills the requirements for which it was designed and meets quality expectations.
- **Testing** measures the quality of an application or project.
- In general, developers should assume that their project has bugs—even if they haven't been discovered yet!
- Most experts conclude that it is much easier (and cheaper) to identify and correct bugs during development rather than after deployment. Therefore, it is important to incorporate testing into the development process.
- **Software quality** can be measured with a variety of metrics, such as performance metrics and reliability metrics.
- Some testing is closely related to the specifics of the users' computer systems, such as the operating system and device drivers. It may also be necessary to test dependencies on network connections.

FAST TRACK HELP

- **MSDN® Testing for Continuous Delivery with Visual Studio 2012**
<http://msdn.microsoft.com/en-us/library/jj159345.aspx>
- **MSDN Test Early and Often**
<http://msdn.microsoft.com/en-us/library/ee330950.aspx>
- **MSDN Testing Overview**
<http://msdn.microsoft.com/en-us/library/aa292191.aspx>



OBJECTIVE

describe testing fundamentals 1.3-1.4

DESCRIBE FUNDAMENTALS OF PROGRAMMING; DESCRIBE APPLICATION LIFECYCLE MANAGEMENT

SCENARIO: Jay Henningsen has been promoted to project manager at Litware, Inc. An avid basketball fan, Jay is going to create a web application for recreational basketball leagues around the world. League managers or commissioners will be able to create leagues, establish teams, and assign players to each team, with online signup of a league-wide draft.

Then, each league commissioner will be able to use the application to schedule games and keep statistics. The web application will automatically report scores, update league-wide standing, and maintain individual and team statistics. Players will be able to log on and check the updates, and commissioners can easily send reports to parents, spectators and local media organizations.

Litware, Inc., prefers to utilize an agile methodology when developing a software project.

- 1. Which of the following data types would be the best choice for tracking a player's scoring average per game?**
 - a. Integer
 - b. Double
 - c. String
- 2. Which of the following is likely to be a characteristic of Litware's agile philosophy?**
 - a. A lengthy, detailed design process before coding begins
 - b. Finishing all parts of a phase before moving on to the next
 - c. Multiple small releases that respond to user feedback
- 3. Which of the following is an example of a sequential development approach?**
 - a. Agile
 - b. Extreme Programming
 - c. Waterfall

HINT 

Extreme Programming is one example of an agile methodology.

Answers

1. Which of the following data types would be the best choice for tracking a player's scoring average per game?
b. Double
2. Which of the following is likely to be a characteristic of Litware's agile philosophy?
c. Multiple small releases that respond to user feedback.
3. Which of the following is an example of a sequential development approach?
c. Waterfall

Essential details

- Developers can employ a variety of philosophies, including **sequential** and **iterative models**.
- In **sequential development, (e.g. waterfall model)**, the development proceeds through a set of phases in a specific order.
- The **waterfall model** is a common approach to sequential development.
- In an **iterative approach**, the team cycles through a set of phases for a portion of the project, then moves on to another portion and repeats until the project is complete.
- The **agile model**, and its derivatives, is a common approach to iterative development.
- The **.NET languages (C#, Visual Basic, and F#)** create managed code that is executed by the common language runtime environment rather than directly by the operating system.
- **Native languages, (e.g. Visual C++)**, create code that is executed directly by the operating system, referred to as **unmanaged code**.

FAST TRACK HELP

- **MSDN® Data Types (C# Programming Guide)**
<http://msdn.microsoft.com/en-us/library/ms173104.aspx>
- **MSDN Compiled vs. Interpreted Applications**
<http://msdn.microsoft.com/en-us/library/aa240840.aspx>
- **MSDN Managed Execution Process**
<http://msdn.microsoft.com/en-us/library/k5532s8a.aspx>
- **MSDN Testing Methodologies**
<http://msdn.microsoft.com/en-us/library/ff649520.aspx>



OBJECTIVE 2

Describe Testing Methodology

IN THIS CHAPTER

- 2.1 Describe Testing Techniques
- 2.2 Describe Testing Levels
- 2.3 Describe Testing Types



OBJECTIVE

describe testing methodology 2.1

DESCRIBE TESTING TECHNIQUES

SCENARIO: Jay Henningsen is leading a team of developers working on a web application that will help people organize and manage recreational basketball leagues, and will allow players and fans from those leagues to access schedules, standings, statistics, and announcements.

Litware Inc., Jay's company, wants him to use an agile model with his team. Jay and his senior developers are hard at work creating a testing plan that will ensure quality software throughout the fast-moving development process.

Jay has created much of the testing plan, including detailed test cases for many of the typical tasks his users may want to complete within the application. Jay has hired Aidan Delaney, one of Litware's most experienced developers, to oversee testing and software quality assurance.

- 1. Aidan Delaney will test some of the features in the application by logging in to the web application and running it as if he was a regular user. What is this type of testing called?**
 - a. Automated testing
 - b. Beta testing
 - c. Manual testing
- 2. Aidan suggests that Jay utilize some black box testing. Which of the following people would be the best person to execute those tests?**
 - a. Jay (the project manager)
 - b. Any of the project's software developers
 - c. A friend of Jay who works in Litware's marketing department
- 3. Which of the following is a good scenario for automated testing?**
 - a. Load testing how the application handles thousands of simultaneous users.
 - b. Testing how basketball fans like the user interface.
 - c. A one-time test of a new scheduling feature.

HINT ↓

Black box testing focuses on the output or results of the application based on specific inputs.

Answers

1. Aidan Delaney will test some of the features in the application by logging in to the web application and running it as if he was a regular user. What is this type of testing called?
 - c. **Manual testing**
2. Aidan suggests that Jay utilize some black box testing. Which of the following people would be the best person to execute those tests?
 - c. **A friend of Jay's who works in Litware's marketing department.** In black box testing, the tester should be someone who doesn't know how the source code works.
3. Which of the following is a good scenario for automated testing?
 - a. **Load testing how the application handles thousands of simultaneous users.** Since automated tests may require significant time to create, one-time tests are often best handled manually.

Essential details

- **Automated testing** uses software (such as Microsoft Test Manger in Visual Studio) to execute tests automatically.
- **Manual testing** is a testing approach in which the tester plays the role of a user and interacts with the application to find defects.
- **Black box testing** is executed by someone without knowledge of the internal workings or source code. Black box testing focuses on the output or results to assess quality without an understanding of how the software works.
- **White box testing** is planned and executed by someone who examines and understands the source code. White box testers look at the source code and design tests to ensure that it works as intended.

FAST TRACK HELP

- **MSDN® Testing Process for Application Block**
<http://msdn.microsoft.com/en-us/library/ff649517.aspx>
- **MSDN Black Box and White Box Testing for Application Blocks**
<http://msdn.microsoft.com/en-us/library/ff649503.aspx>
- **MSDN Manual System Tests**
<http://msdn.microsoft.com/en-us/library/jj159334.aspx>
- **MSDN Test Early and Often**
<http://msdn.microsoft.com/en-us/library/vstudio/ee330950.aspx>



DESCRIBE TESTING LEVELS

SCENARIO: At Litware Inc., Jay Henningsen is managing a team of developers working on a web application that will organize recreational basketball leagues and maintain team and individual statistics as the season progresses.

A key feature of the application is automatically scheduling playoff games using “seeding,” or ranks, so that the top-ranked team plays the lowest-ranked team, the second-ranked team plays the team that finished second-to-last, etc. The team has broken the algorithm down into pieces, with each member building a method to solve one part of the problem.

Jay is writing a method that ranks the teams based on wins and losses. A different developer is building a method that iterates through a ranked list and creates games between the appropriate pairs of teams. And a third member is developing a method that looks at the times the court is available and schedules each game.

- 1. When should Jay create tests for his method?**
 - a. Before starting to build the method
 - b. When he has written enough code to be able to test something
 - c. After the method is completed
- 2. What is the name for the type of test Jay is creating for his method?**
 - a. Beta test
 - b. Integration test
 - c. Unit test
- 3. When Jay begins assembling pieces from different developers, what kind of testing will he do?**
 - a. Beta test
 - b. Integration test
 - c. Unit test

HINT 

Integration tests are also known as component tests.

Answers

1. When should Jay create tests for his method?
 - a. **Before starting to build the method.** Best practices suggest creating tests first, then using that to assist with implementation
2. What is the name for the types of test Jay is creating for his method?
 - c. **Unit Test**
3. When Jay begins assembling pieces from different developers, what kind of testing will he do?
 - b. **Integration Test**

Essential details

- **Unit tests** are automated tests that verify functionality at the component, class, method, or property level.
- The primary goal of **unit testing** is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect.
- Unit testing is completed before integrating units into components to test how they work together. These tests are called **integration tests**, or **component tests**.
- Thorough unit testing makes integration testing more efficient—if you find a bug during integration testing, it is likely to be a problem with the interface between the components, rather than a problem within one of the components.

FAST TRACK HELP

- **MSDN® Unit Testing**
<http://msdn.microsoft.com/en-us/library/aa292197.aspx>
- **MSDN Unit Testing: Testing the Inside**
<http://msdn.microsoft.com/en-us/library/jj159340.aspx>
- **MSDN Integration Testing**
<http://msdn.microsoft.com/en-us/library/aa292128.aspx>
- **MSDN Unit testing, component level testing and UI testing, what to use and when**
<http://blogs.msdn.com/b/raulperez/archive/2010/04/29/unit-testing-component-level-testing-and-ui-testing-what-to-use-and-when.aspx>



DESCRIBE TESTING TYPES

SCENARIO: As Litware Inc., continues to plan development of a web application to organize and manage recreational basketball leagues, team members are brainstorming a thorough plan for testing their work. Jay Henningsen, the project manager, wants to make sure that every aspect of the application is able to be tested efficiently, including the ability to re-test code throughout the full development cycle.

Jay has been working closely with Litware's IT department to establish the hardware the application will require when deployed, but Jay doesn't have a lot of background in server administration. He is very worried that the hardware and software will not scale well to thousands of users.

- 1. What type of testing will help determine how many users the system can handle before failing?**
 - a. Accessibility testing
 - b. Regression testing
 - c. Stress testing
 - d. Usability testing
- 2. Which of the following attempts to uncover new flaws in previously tested code?**
 - a. Accessibility testing
 - b. Regression testing
 - c. Stress testing
 - d. Usability testing
- 3. Which of the following best describes one purpose of localization testing?**
 - a. Validating performance on a local area network (LAN) before deployment
 - b. Testing very small segments or units of code before they are combined
 - c. Ensuring that the app will work as expected when translated to a different language

HINT 

Stress tests are sometimes referred to as load tests.

Answers

1. What type of testing will help determine how many users the system can handle before failing?
c. Stress testing
2. Which of the following attempts to uncover new flaws in previously tested code?
b. Regression testing. Regressions occur when new (or newly modified) code causes an unexpected behavior in code that has not been modified.
3. Which of the following best describes one purpose of localization testing?
c. Ensuring that the app will work as expected when translated to a different language. There are other localization issues, but UI problems in different languages are a big concern.

Essential details

- **Accessibility testing** validates an application's support for users with disabilities.
- **Localization testing** checks the quality of a product for a particular target culture/locale.
- **Performance testing** is intended to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload.
- **Regression testing** refers to the retesting of a modified program, rather than a test of only the modified units, to ensure that no errors have been introduced.
- **Security testing** is intended to validate an application's security services and identify potential security flaws.
- **Stress testing** involves testing of a system's functional limits, performed by subjecting the system to extreme conditions (peak volumes of data or a many simultaneous users).
- **Usability testing** evaluates a project by studying how users actually use the software.

FAST TRACK HELP

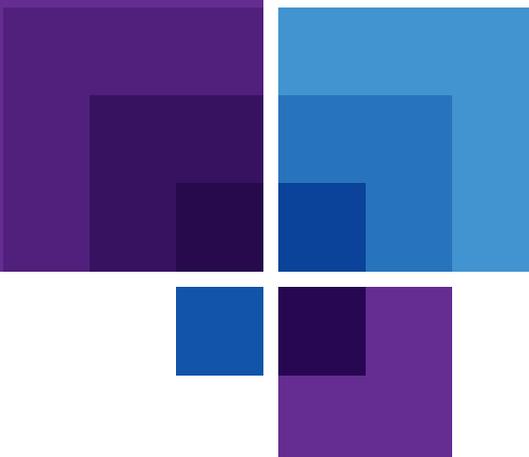
- **MSDN® Regression Testing**
http://blogs.msdn.com/b/narendra_parihars_blog/archive/2012/04/20/regression-testing.aspx
- **MSDN Testing Performance and Stress Using Visual Studio Web Performance and Load Tests**
<http://msdn.microsoft.com/library/dd293540.aspx>
- **MSDN Testing a User Interface**
<http://msdn.microsoft.com/en-us/library/windows/desktop/ff728827.aspx>



OBJECTIVE 3

Create Software Tests

IN THIS CHAPTER

- 3.1 - 3.2 Describe User-Centric Testing; Describe Software Testability
 - 3.3 Create Test Plan Components
 - 3.4 Describe Feature Tests
 - 3.5 Define Appropriately Scoped Test Cases
- 

OBJECTIVE

create software tests 3.1 - 3.2

DESCRIBE USER-CENTRIC TESTING; DESCRIBE SOFTWARE TESTABILITY

SCENARIO: Litware Inc. is developing a web application to manage recreational basketball leagues from anywhere around the world. Jay Henningsen is the project manager, and he will be implementing a test-driven development approach.

After discussing the project with people who manage leagues, as well as basketball players, Jay has identified three categories of application users: league commissioners, players, and fans. Each type of user will have access to different features and functionality. Commissioners can complete tasks related to running a league, such as creating a team and scheduling a game. A player can upload a photo and see contact information for teammates. Fans can view standings and results of games, among other activities.

- 1. When will Jay and his team create test cases for the project?**
 - a. Before beginning to develop source code
 - b. Before the project goes into beta testing
 - c. Before deploying the project for public use
- 2. What is the primary type of test the developers will create as they implement test-driven development?**
 - a. Alpha test
 - b. Unit test
 - c. User test
- 3. Jay will temporarily program the application to respond to a specific sequence of keystrokes that will automatically create a league filled with fictional teams and players so that he doesn't have to create them manually when he tests the application. What is this called?**
 - a. A cheat code
 - b. A testing hook
 - c. A unit test



Answers

1. When will Jay and his team create test cases for the project?
 - a. **Before beginning to develop source code.** Test-driven development requires test cases before developers begin to create code.
2. What is the primary type of test the developers will create as they implement test-driven development?
 - b. **Unit test**
3. Jay will temporarily program the application to respond to a specific sequence of keystrokes that will automatically create a league filled with fictional teams and players so that he doesn't have to create them manually when he tests the application. What is this called?
 - b. **A testing hook.** While this particular type of hook sounds similar to a cheat code in a video game, it is only there for testing.

Essential details

- Many projects begin with the development of **user-requirement documents** or models. These focus on how the software will work from the user's perspective.
- A **use case diagram** is a model that describes who uses the system, and what they use it for.
- A **user story** is a written description that communicates functionality that is of value to the end user of the product or system. It is often derived from a use case diagram, and is used by developers to create unit tests.
- **Test-driven development** is a programming model that uses unit tests to guide the design of software.
- A **testing hook** is a feature that allows internal functionality to be tested independent of the rest of the software. Hooks are often—but not always—removed before software is deployed or released.

FAST TRACK HELP

- **MSDN® Modeling User Requirements**
<http://msdn.microsoft.com/en-us/library/vstudio/dd409376.aspx>
- **MSDN User Story (Agile)**
<http://msdn.microsoft.com/en-us/library/dd380634.aspx>
- **MSDN Testing Methodologies**
<http://msdn.microsoft.com/en-us/library/ff649520.aspx>



CREATE TEST PLAN COMPONENTS

SCENARIO: Roger Harui and Marketa Pobudova met last semester in a computer science class at the local university. They have decided to create a multi-player Windows Phone game. Being avid chess players, they decided on a board game using a checkerboard layout but with different game pieces and movement rules.

They are going to do most of the development themselves, but they foresee including a couple of their classmates to test the game and, if needed, assist with development. On the advice of a professor, they begin by setting up Team Foundation Server and creating a project for the game. Their professor also advises them to use an agile development model.

- 1. Which process will Team Foundation Server facilitate?**
 - a. Deploying their game on Windows Phone devices.
 - b. Application lifecycle management.
 - c. Porting their game to HTML5.
- 2. Roger and Marketa want 80% code coverage. What does this mean?**
 - a. 80% of each version will be complete before testing.
 - b. 80% of all code will be included in testing.
 - c. 80% of the code will be developed by them personally.
- 3. Which is most likely to guide their test plan creation?**
 - a. A plan for each sprint.
 - b. A plan for each game piece.
 - c. A plan for each developer.

HINT 

Their agile approach will incorporate a series of sprints.

Answers

1. Which process will Team Foundation Server facilitate?
b. Application lifecycle management.
2. Roger and Marketa want 80% code coverage. What does this mean?
b. 80% of all code will be included in testing.
3. Which is most likely to guide their test plan creation?
a. A plan for each sprint. An iteration of the agile cycle is a sprint, and each sprint should have its own thorough test plan.

Essential details

- Each development project goes through a complete **application lifecycle**, beginning with initial planning and progressing through deployment and maintenance.
- **Microsoft Visual Studio** provides a comprehensive suite of tools for Application Lifecycle Management (ALM). Team Foundation Server (TFS) is at the center of this functionality.
- **TFS** provides version control, a build system, and tools and metrics for managing and organizing projects.
- Other tools within Microsoft Visual Studio work with Team Foundation Server to integrate planning, testing, and development efforts.
- While many of these tools offer explicit support for agile approaches to development, they support a very wide variety of development models.
- An important consideration when developing a **testing strategy** is setting a goal for code coverage, which refers to the percentage of code that is covered by tests. 80% code coverage is a common goal; Microsoft Visual Studio makes it easy to track code coverage.

FAST TRACK HELP

- **MSDN® Application Lifecycle Management with Visual Studio and Team Foundation Server**
<http://msdn.microsoft.com/en-us/library/fda2bad5.aspx>
- **MSDN Test Early and Often**
<http://msdn.microsoft.com/en-us/library/ee330950.aspx>
- **MSDN Testing in the Software Lifecycle**
<http://msdn.microsoft.com/en-us/library/jj159342.aspx>



DESCRIBE FEATURE TESTS

SCENARIO: Litware, Inc. is developing a web application that will enable users to organize and manage recreational basketball leagues. Additionally, players in each league will be able to login and manage their own profile and easily access statistics and contact information for their teammates.

Jay Henningsen is the project manager and he has guided the development team through a couple of sprints or development iterations. The software meets Jay's testing objectives, but when he showed screenshots of the application to a friend who is involved in a basketball league but not a technical professional, he realized that non-technical users may have difficulties using the system.

- 1. Which of the following is an example of feature testing?**
 - a. Asking an end-user to try the application for an hour or two.
 - b. Running unit tests for the entire project when a new feature is added.
 - c. Running tests to make sure the functions for adding a new team work correctly.
- 2. If Jay asks his friend to try the application, what kind of testing is he conducting?**
 - a. Accessibility testing.
 - b. Usability testing.
 - c. White box testing.
- 3. What is a coded UI test?**
 - a. An automated test of the UI.
 - b. A test of the code that creates the UI.
 - c. A manual test of UI components.

HINT 

A software feature is a capability of the application—something the user can do with it.

Answers

1. Which of the following is an example of feature testing?
 - c. **Running tests to make sure the functions for adding a new team work correctly.** Option A describes usability testing; option B would be best described as regression testing.
2. If Jay asks his friend to try the application, what kind of testing is he conducting?
 - b. **Usability testing.**
3. What is a coded UI test?
 - a. **An automated test of the UI.**

Essential details

- **Feature testing** focuses on the functionality of individual features in the software, independent of other features. For example, testing to make sure that the feature enabling league commissioners to add a team to a league.
- There are several types of testing that assist with **feature testing**.
- **Exploratory testing** does not employ a set of tests defined in advance, and without a script of predetermined steps. Also called ad hoc testing, and is the type of testing developers often use themselves—try the program out and see if it works!
- **Usability tests** are studies of how end-users interact with the software. They are, by definition, a type of black box testing—the user doesn't understand the inner workings of the software.
- **User interface (UI) testing** focuses only on the user interface (UI) of the software. An automated UI test is called a **coded user interface test (CUIT)**.

FAST TRACK HELP

- **MSDN® Exploratory Software Testing**
<http://msdn.microsoft.com/en-us/library/jj620911.aspx>
- **MSDN System Testing with Coded UI Tests**
<http://msdn.microsoft.com/en-us/library/jj620891.aspx>
- **MSDN Testing Concepts and Phases**
<http://msdn.microsoft.com/en-us/library/ff798502.aspx>



OBJECTIVE

create software tests 3.5

DEFINE APPROPRIATELY SCOPED TEST CASES

SCENARIO: Roger Harui and Marketa Pobudova are university students working together to create a multi-player game for Windows Phone devices. The game was inspired by chess, using a similar board but with the students' own rules and game pieces.

As the pair begins the programming process, they are developing code targeting their first user story: "A player can move a game piece to a valid location on the board." Though each piece has its own rules for movement, just like the different pieces in a chess game, no game piece can move to a square outside of the board. The board has ten squares per side (unlike the 8-by-8 layout of a chessboard), numbered from 0-9 on each axis.

- 1. Which of the following is a boundary case for valid squares within a row on the board?**
 - a. 4
 - b. 10
 - c. 75
- 2. The first test case they write tests if the code works with a valid location in the middle of the board. What type of test case is this?**
 - a. Alpha test case
 - b. Boundary test case
 - c. Happy test case
- 3. The pair let a friend test the code; the only details they give this tester is "move a piece." Which of the following best describes this type of test?**
 - a. Boundary test
 - b. Exploratory test
 - c. User Interface test

HINT ↓

Boundary cases are also called edge cases.

Answers

1. Which of the following is a boundary case for valid squares within a row on the board?
 - a. 10. Since the last square in the row is 9, 10 is near the boundary. 75 is far outside of the expected range, and 4 is well within range.
2. The first test case they write tests if the code works with a valid location in the middle of the board. What type of test case is this?
 - a. **Happy test case.**
3. The pair let a friend test the code; the only details they give this tester is “move a piece.” Which of the following best describes this type of test?
 - a. **Exploratory test.** Exploratory test cases provide few, if any, details about the process.

Essential details

- When beginning a sprint or method, developers may use a “**happy case**” to get started. This is a test case that uses exactly the kind of data the code is expecting. Once that is done, it is important to test data that is NOT within the expected range.
- **Test cases** sometimes can’t cover all possible values, so it is important to design tests that cover many possible bugs.
- **Boundary testing** focuses on values at or near the extremes—the minimum or maximum values. In the case of this game, boundary tests would cover the edges of the board. Boundary cases are also called edge cases.
- **Manual test cases** often provide a great deal of detail, so that testers who may be unfamiliar with previous testing efforts can conduct tests consistently.
- **Manual tests** are usually broken into multiple steps, each with expected results. The test must pass all of these steps.

FAST TRACK HELP

- **MSDN® Manual System Tests**
<http://msdn.microsoft.com/library/jj159334.aspx>
- **MSDN Black Box and White Box Testing for Application Blocks**
<http://msdn.microsoft.com/en-us/library/ff649503.aspx>
- **MSDN Test Automation Code Review Guidelines**
<http://msdn.microsoft.com/en-us/library/ff519670.aspx>

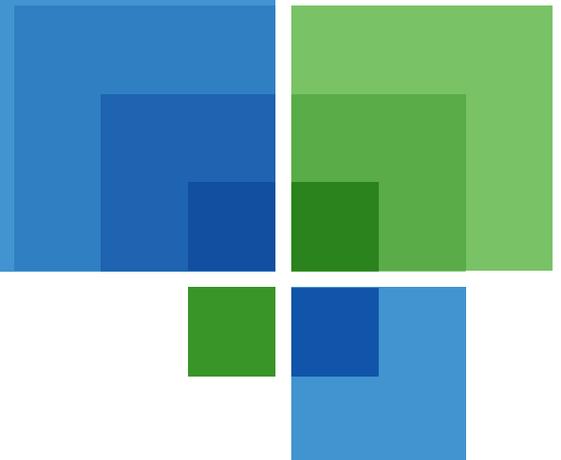


OBJECTIVE 4

Manage Software Testing Projects

IN THIS CHAPTER

- 4.1 Describe Testing Milestones
- 4.2 Describe the Agile Process
- 4.3 Work with Distributed Teams
- 4.4 Define Test Reports



DESCRIBE TESTING MILESTONES

SCENARIO: Contoso Pharmaceuticals is creating a mobile app designed to help cancer patients manage the complex and potentially overwhelming details of their treatment and care. It will track appointments, medications, clinical treatments, and contact information for all of a patient's healthcare providers. The app will also provide user-friendly access to a wide range of healthcare databases so the patient can access accurate and up-to-date information about their illness.

An intern named Sanjay Patel is working with Ann Beebe, a program manager, on some of the early planning and preparations. While Sanjay came up with the original idea and has a great deal of enthusiasm, he has never worked on a project of this scale, so he is relying on Ann's management expertise to translate that enthusiasm into a great software system.

- 1. Ann will use Team Foundation Server (TFS) to establish a Microsoft® SharePoint® portal with all of the documents related to the app. What is this feature of TFS called?**
 - a. Agile process
 - b. Milestone management process
 - c. Process guidance
- 2. Before the team begins writing test cases and code, Ann must approve a basic user requirements document. What is that approval called?**
 - a. A test plan
 - b. A Kanban approval
 - c. A sign off
- 3. Ann and Sanjay brainstorm a list of requirements the project must meet by the end of the first sprint. What is this list called?**
 - a. Entry criteria
 - b. Exit criteria
 - c. Sprint criteria



Answers

1. Ann will use TFS to establish a SharePoint portal with all of the documents related to the app. What is this feature of TFS called?
c. Process guidance
2. Before the team begins writing test cases and code, Ann must approve a basic user requirements document. What is that approval called?
c. A sign off.
3. Ann and Sanjay brainstorm a list of requirements the project must meet by the end of the first sprint. What is this list called?
b. Exit criteria

Essential details

- The term **process** refers to all of the activities that go into the development of a software application.
- **Process guidance** is the tool used in Microsoft Team Foundation Server to manage process.
- Often, process is broken into multiple **milestones**. Internal milestones focus on the team's activities; external milestones typical focus on the end user or customer.
- In addition to large milestones, teams generally work in phases. An agile team will work in sprints, while a waterfall team might focus on stages and versions.
- **Entry criteria** are the conditions that must be present before a phase can begin successfully.
- **Exit criteria** are the conditions that must be met before a particular milestone is complete.
- Depending on the management and development styles, phases and/or milestones may also require a **sign off**.

FAST TRACK HELP

- **MSDN® Process Guidance and Process Templates for Team Foundation Server**
<http://msdn.microsoft.com/en-us/library/hh533801.aspx>
- **MSDN Planning and Tracking Projects**
<http://msdn.microsoft.com/en-us/library/dd286619.aspx>
- **MSDN Work Items and Workflow (Agile)**
<http://msdn.microsoft.com/en-us/library/dd997897.aspx>



DESCRIBE THE AGILE PROCESS

SCENARIO: Contoso Pharmaceuticals is creating a mobile app designed to help cancer patients manage the complex and potentially overwhelming details of their treatment and care. It will track appointments, medications, clinical treatments, and contact information for all of a patient's healthcare providers. The app will also provide user-friendly access to a wide range of healthcare databases so the patient can easily access accurate and up-to-date information about their illness.

Sanjay Patel is working on the project under the guidance of Ann Beebe, an experienced program manager. Their team will function using agile principles, including scrum and extreme programming approaches.

- 1. Which of the following is characteristic of extreme programming?**
 - a. A project manager making all decisions.
 - b. Programmers working in pairs.
 - c. One major release at the end of the development process.
- 2. In scrum, what is the basic development phase called?**
 - a. A kanban
 - b. A scrumble
 - c. A sprint
- 3. Which of the following is NOT a characteristic of agile development?**
 - a. Customer involvement in the development process.
 - b. Short release cycles.
 - c. Strictly adhering to the initial project design.

HINT 

Agile development is iterative—the team repeats the phases of the development process until the project is complete.

Answers

1. Which of the following is characteristic of extreme programming?
 - b. **Programmers working in pairs**
2. In scrum, what is the basic development phase called?
 - c. **A sprint.** In extreme programming, a sprint is called an iteration.
3. Which of the following is NOT a characteristic of agile development?
 - c. **Strictly adhering to the initial project design.**

Essential details

- **Agile development** is a collection of values, principles, and practices that incorporates iterative development, test-driven development, and immediate feedback. It is a widely-used model for software development with several derivations:
 - **Extreme programming** is an agile programming or engineering methodology that emphasizes test-driven development and utilizes pairs of programmers.
 - **Scrum** is an agile framework or management approach that values responsiveness to customer requirements and focuses on time-based production cycles.
- Extreme programming and scrum are not mutually exclusive—they are commonly used together.
- Agile development focuses on brief, intense periods of development. In scrum, these are referred to as **sprints**. Extreme programming refers to them as **iterations**.
- Some agile teams utilize **kanban**, a scheduling model that enables just-in-time delivery by managing the workflow pipeline to avoid overloading the development team.

FAST TRACK HELP

- **MSDN® Agile Principles and Values**
<http://msdn.microsoft.com/en-us/library/dd997578.aspx>
- **MSDN Testing Methodologies**
<http://msdn.microsoft.com/en-us/library/ff649520.aspx>
- **MSDN Scrum**
<http://msdn.microsoft.com/en-us/library/dd997796%28v=VS.100%29.aspx>



The graphic consists of a blue rectangular box. On the left side, there is a white icon of a person sitting at a desk with a computer monitor. Below the icon, the word "TRAINING" is written in white capital letters. On the right side of the box, the text "Track Your Score" is written in white. Below this text, there is a white line followed by a slash and the number "3", indicating a score of 0 out of 3.

WORK WITH DISTRIBUTED TEAMS

SCENARIO: Contoso Pharmaceuticals is creating a mobile app designed to help cancer patients manage the complex and potentially overwhelming details of their treatment and care. It will track appointments, medications, clinical treatments, and contact information for all of a patient's healthcare providers. The app will also provide user-friendly access to a wide range of healthcare databases so the patient can easily access accurate and up-to-date information about their illness.

Ann Beebe will be managing the project, and she has an important decision to make before development begins. All of the developers in her regional office are occupied with an ongoing project and are not available for at least six months. However, Contoso does have developers available at two other regional offices on other continents. Ann has received permission to use those remotely-located developers if she wishes.

- 1. Which phrase describes a team with members working in different locations?**
 - a. Co-located team
 - b. Distributed team
 - c. Scrum team
- 2. Which of the following disadvantages is most often associated with using remotely-located team members?**
 - a. Difficulty communicating effectively
 - b. Cost/budget overruns
 - c. Software incompatibilities
 - d. None of the above
- 3. Which tool will assist Ann in overcoming many of the difficulties of remotely-located team members?**
 - a. Microsoft® Team Foundation Server
 - b. A Microsoft SharePoint® portal
 - c. Skype™
 - d. All of the above

HINT 

Telecommuting and outsourcing can both lead to remotely-located development teams.

Answers

1. Which phrase describes a team with members working in different locations?
 - b. **Distributed team**
2. Which of the following disadvantages is most often associated with using remotely-located team members?
 - a. **Difficulty communicating effectively.** Team members often chat informally; these interactions are difficult when they can't meet face-to-face.
3. Which tool will assist Ann in overcoming many of the difficulties of remotely-located team members?
 - d. **All of the above**

Essential details

- A **co-located** team consists of members who work at the same location and time.
- A **distributed team** includes team members working at different locations and, possibly, in different time zones.
- Co-located teams are generally more effective due to enhanced communication, including informal chats that contribute to the project and help team members bond.
- Distributed teams can have several benefits:
 - They may be less expensive due to cheaper labor and facilities costs
 - They open opportunities for a deeper talent base.
 - It provides local expertise if the customer is not located nearby.
- Distributed teams rely on technology to solve many communications challenges; video conferencing, phone calls, and a SharePoint portal all help team members work together seamlessly.

FAST TRACK HELP

- **MSDN® Collaborating within a Team Using Team Project Resources**
[http://msdn.microsoft.com/en-us/library/ms242904\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ms242904(v=vs.100).aspx)
- **MSDN Distributed Scrum**
<http://msdn.microsoft.com/en-us/library/jj620910.aspx>
- **MSDN Distributed Agile Development (PDF)**
http://download.microsoft.com/download/4/4/a/44a2cebd-63fb-4379-898d-9cf24822c6cc/distributed_agile_development_at_microsoft_patterns_and_practices.pdf



DEFINE TEST REPORTS

SCENARIO: At Contoso Pharmaceuticals, Ann Beebe's team has finished the first sprint in the development of a mobile app to help cancer patients manage and track their care. Working with team members in three different locations, the group has implemented two significant user stories. Ann has gotten feedback from Joe Healy, an upper level manager at Contoso who is acting as the team's "product owner," or customer. He is pleased with the progress and has helped prioritize the user stories he'd like to see Ann's team develop next.

Ann calls a retrospective meeting to debrief their work and discuss the next sprint. She has prepared a burndown report from the previous sprint, as well as the backlog with Joe's priorities.

- 1. What is burndown?**
 - a. Completed versus remaining work over time.
 - b. Finished versus unfinished user stories.
 - c. Resolved versus unresolved bugs.
- 2. A team member asks how much work the team completed during the sprint. Which metric would provide the answer?**
 - a. Dashboard
 - b. Scrum
 - c. Velocity
- 3. Which of the following best describes the backlog?**
 - a. A list of unresolved bugs.
 - b. A list of unfinished tasks.
 - c. A list of completed user stories.

HINT 

Ann is using a scrum model. To her, a "sprint" is an iteration in the development process.

Answers

1. What is burndown?
 - a. **Completed versus remaining work over time.**
2. A team member asks how much work the team completed during the sprint. Which metric would provide the answer?
 - a. **Velocity**
3. Which of the following best describes the backlog?
 - a. **A list of unfinished tasks.**

Essential details

- **Microsoft® Team Foundation Server (TFS)** features two primary ways of viewing data related to application lifecycle management (ALM): dashboards and reports.
- A **dashboard** is a dynamic overview of the project using easy-to-read visuals and charts.
- A **report** is a visualization or textual presentation of data based on a specific search or criteria.
- Dashboards and reports may offer similar information, with reports geared toward exporting for printing, emailing, or otherwise viewing outside of TFS.
- Common terms found in ALM reports:
 - **Burndown** shows the trend of completed and remaining work over time.
 - **Burn rate** is a calculation of the completed and required rate of work based on the specified time period.
 - **Velocity** measures work accomplished per unit of time, for example, iteration.

FAST TRACK HELP

- **MSDN® Burndown and Burn Rate Report (Agile)**
<http://msdn.microsoft.com/en-us/library/dd380678.aspx>
- **MSDN Excel Reports (Agile)**
<http://msdn.microsoft.com/en-us/library/vstudio/dd997876.aspx>
- **MSDN My Dashboard (Agile)**
<http://msdn.microsoft.com/en-us/library/dd420561.aspx>



OBJECTIVE 5

Work with Bugs

IN THIS CHAPTER

- 5.1 Detect Software Defects
- 5.2 Log Bugs
- 5.3 Manage Bugs



OBJECTIVE

work with bugs 5.1

DETECT SOFTWARE DEFECTS

SCENARIO: Litware Inc. is developing a web application to organize and manage recreational basketball leagues, with functionality to add team, schedule games, and track statistics and results. Jay Henningsen, the project manager, has put together a comprehensive test plan.

Jay found an unused conference room and has set up six different computers. He wants to use these computers to help with testing—he figures that dividing the workload over many computers will make things quicker, and will allow him to quickly test the application in a variety of circumstances.

Jay will manage the testing process from his own workstation in his office.

- 1. Which of the following should Jay install and configure on his workstation?**
 - a. A test agent
 - b. A test client
 - c. A test controller
- 2. Which of the following should he install on each of the six testing machines?**
 - a. A test agent
 - b. A test client
 - c. A test controller
- 3. Which of the follow best describes how the test machines should be configured in relation to Team Foundation Server?**
 - a. A lab environment
 - b. A test suite
 - c. A virtual test



Answers

1. Which of the following should Jay install and configure on his workstation? c. **A test controller**
2. Which of the following should he install on each of the six testing machines? a. **A test agent**
3. Which of the follow best describes how the test machines should be configured in relation to Microsoft Team Foundation Server? a. **A lab environment**

Essential details

- A **test plan** is a set of test cases, their associated test configuration information, and the iteration when it is planned to run these tests. The test cases can be organized into a test suite hierarchy to use when running the test suites.
 - To manage a test plan, use Microsoft Test Manager instead of Microsoft Visual Studio®.
- **Automated testing** is a testing approach that uses software (such as Microsoft Test Manger in Visual Studio) to execute tests automatically.
- An **automation script** is a set of steps that a computer may run programmatically to test the functionality of the system.
- A **lab environment** refers to a collection of virtual and physical machines, which can be used to develop and test applications.
- A **test controller** is a background process that manages a set of machines with the test agent software installed. The test controller acts as a centralized server for the test effort.
- A **test agent** is a background process that receives, runs, and reports on tests and collects data on a single computer. The test agent communicates with test controller, usually located on another computer.

FAST TRACK HELP

- **MSDN® Installing and Configuring Test Agents and Test Controllers**
<http://msdn.microsoft.com/en-us/library/vstudio/dd648127.aspx>
- **MSDN Essential Guide for Running Automated Tests from a Test Plan**
<http://msdn.microsoft.com/en-us/library/ff472576.aspx>
- **MSDN Running Tests in Microsoft Test Manager**
<http://msdn.microsoft.com/en-us/library/dd286680.aspx>



OBJECTIVE

work with bugs 5.2

LOG BUGS

SCENARIO: Litware, Inc. is developing a web application that will allow users to organize and manage recreational basketball leagues. Additionally, players in each league will be able to log in and manage their own profile and easily access statistics and contact information for their teammates.

Jay Henningsen, the project manager, has found a couple of bugs after executing the team's test plan. First, he found that if a user does not enter the correct password on three consecutive attempts, the website crashes without any kind of message to the user. Second, he has discovered that if any player in a league goes through an entire season without scoring any points, the page displaying the league's statistical leaders will have some formatting errors.

- 1. Jay decides that the password bug will be the next item the team will address. What number should he assign for that work item's *stack rank*?**
 - a. 1
 - b. 5
 - c. 10
- 2. Because the team rarely tests with players scoring zero points, the second bug was difficult to discover. Where should Jay document the process for observing the defect, so developers can find it?**
 - a. Bug severity
 - b. Repro steps
 - c. Test script
- 3. Which feature would be helpful for a developer struggling to reproduce a bug?**
 - a. IntelliSense
 - b. IntelliTrace
 - c. Process Guidance



Answers

1. Jay decides that the password bug will be the next item the team will address. What number should he assign for that work item's *stack rank*?
 - a. **1.** Lower numbers indicate a higher ranking in the stack order.
2. Because the team rarely tests with players scoring zero points, the second bug was difficult to discover. Where should Jay document the process for observing the defect, so developers can find it?
 - a. **Repro steps**
3. Which feature would be helpful for a developer struggling to reproduce a bug?
 - a. **IntelliTrace**

Essential details

- Logging a bug typically includes the following information:
- **Bug priority** is a subjective rating of the bug as it relates to the business. Values include:
 - 1 = Product cannot ship without the successful resolution of the bug, and it should be addressed as soon as possible.
 - 2 = Product cannot ship without the successful resolution of the bug, but it does not need to be addressed immediately.
 - 3 = Resolution of the bug is optional based on resources, time, and risk.
- **Bug severity** is a subjective rating of the impact of the bug on the project. A bug can range from Critical (severity = 1) to Low (severity = 4).
- **Bug stack rank** is a subjective rating of the bug compared to other bugs. A bug that is assigned a lower number should be fixed before a bug that is assigned higher number.
- **Repro steps** are the steps other testers or developers can execute to reproduce a bug.

FAST TRACK HELP

- **MSDN® Bug (Agile)**
[http://msdn.microsoft.com/en-us/library/dd380645\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/dd380645(v=vs.110).aspx)
- **MSDN Triage Workbook**
[http://msdn.microsoft.com/en-us/library/dd380707\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/dd380707(v=vs.100).aspx)
- **MSDN Submitting Bugs**
<http://msdn.microsoft.com/en-us/library/dd286746.aspx>



OBJECTIVE

work with bugs 5.3

MANAGE BUGS

SCENARIO: Dana Birkby is a software tester working on a point-of-sale (cash register) application for Fourth Coffee. The application has been in use for several years, but the company recently hired a new team to update the software and add some new features. Some of the team members were reassigned to begin an unrelated project.

Because of the disruption to the team structure, Adriana Giorgi, the new-appointed project manager, has some quality assurance concerns. She has hired Dana to review the current bugs, generate a new test plan and identify and undiscovered bugs in the current build, and establish a plan for ensuring software quality as the project wraps up.

- 1. Dana's first task is to look at the current bugs and prioritize which ones should be addressed first. What is this called?**
 - a. Bug summary report
 - b. Test strategy
 - c. Triage
- 2. Dana notices many bugs listed as resolved but not verified. If she verifies that the bugs are fixed, what state should they have?**
 - a. Active
 - b. Closed
 - c. Resolved
- 3. Which of the following is NOT a reason for Dana to mark a bug resolved?**
 - a. She can't reproduce the error.
 - b. She decides to postpone fixing it.
 - c. She realizes the bug is not fixed.

HINT 

Dana is working in an agile development model, so the state names reflect an agile philosophy.

Answers

1. Dana's first task is to look at the current bugs and prioritize which ones should be addressed first. What is this called?
c. Triage
2. Dana notices many bugs listed as resolved but not verified. If she verifies that the bugs are fixed, what state should she assign?
b. Closed
3. Which of the following is NOT a reason for Dana to mark a bug *resolved*?
c. She realizes the bug is not fixed. In this case, she should change the bug to *active*.

Essential details

- **Triage** is the process that is used to review newly reported or reopened bugs and assign a priority (or stack rank) and iteration for working on them.
- A bug is assigned a state to track its current progress.
- In agile, possible states include **active**, **resolved**, and **closed**.
- **Active** bugs are still being investigated or fixed.
- **Resolved** bugs have been fixed, but the fix has not been verified. Typically a different team member will verify a fix.
- **Closed** bugs have been fixed and verified.
- Closed bugs can be activated again if they reappear in later builds. This is called regression.
- Any time a state is changed, the team member should identify the reason for the change. Often, the reason is straightforward—such as when the bug is fixed. However, there are other reasons: the bug could not be reproduced, the fix is being postponed, etc.

FAST TRACK HELP

- **MSDN® Bug (Agile)**
<http://msdn.microsoft.com/en-us/library/dd380645.aspx>
- **MSDN Triage Workbook**
<http://msdn.microsoft.com/en-us/library/dd380707.aspx>
- **MSDN Bug Status Report**
<http://msdn.microsoft.com/en-us/library/dd380736.aspx>



OBJECTIVE 6

Automate Software Testing

IN THIS CHAPTER

- 6.1 Describe Test Automation
- 6.2 Define Test Automation Strategies
- 6.3 Write Automation Tests
- 6.4 Manage Test Scripts



DESCRIBE TEST AUTOMATION

SCENARIO: Tailspin Toys is developing a simple, robotic dog for small children. The dog will connect to a computer via a USB cable, and kids will be able to use a simple application to personalize the toy to their liking. Although the hardware is nearly complete, Tailspin has not yet created the software that owners will use to customize their toy dog.

Arlene Huff has been hired to oversee the testing portion of the development project. They are using a test-driven development model, and the team needs to be able to “test early and test often.” Arlene’s first challenge is to suggest efficient testing strategies. She plans to utilize a great deal of automation.

- 1. Which of the following is least likely to be automated from the outset of development?**
 - a. Integration tests
 - b. Unit tests
 - c. User Interface tests
- 2. Which of the following is NOT a benefit of automated tests?**
 - a. They do not need to be updated as the project evolves.
 - b. They can be run unattended.
 - c. They can produce code coverage data.
- 3. Arlene needs to hire another employee to assist with creating automated tests. In addition to testing experience, which skill is likely to be important?**
 - a. Programming
 - b. UI design experience
 - c. Server maintenance

HINT 

In agile development, unit tests are developed before methods are created, and they are nearly always automated.

Answers

1. Which of the following is least likely to be automated from the outset of development?
 - a. **User Interface tests.** UI tests can be automated, but this requires significantly more effort than automating unit and integration tests.
2. Which of the following is NOT a benefit of automated tests?
 - a. **They do not need to be updated as the project evolves.**
3. Arlene needs to hire another employee to assist with creating automated tests. In addition to testing experience, which skill is likely to be important?
 - a. **Programming**

Essential details

- An **automated test** can be thought of as a set of steps that a computer may run programmatically to test the functionality of the software.
- Automated testing complements, but does not replace, manual testing.
- Automated tests must be created with code, so developing them requires programming skills.
- Benefits of automated testing include easy repeatability, unattended execution, and the ability to gather code coverage data.
- Important disadvantages include the time and skill required to create automated tests and the need to maintain tests as code changes.
- **Unit tests, integration/component tests, and load tests** are almost always automated.
- **User interface tests** are sometimes automated. These are called **Coded UI Tests** (CUITs).

FAST TRACK HELP

- **MSDN® Test Early and Often**
<http://msdn.microsoft.com/en-us/library/ee330950.aspx>
- **MSDN Creating Automated Tests**
<http://msdn.microsoft.com/en-us/library/dd380755.aspx>
- **MSDN How to: Associate an Automated Test with a Test Case**
<http://msdn.microsoft.com/en-us/library/dd380741.aspx>



DEFINE TEST AUTOMATION STRATEGIES

SCENARIO: Arlene Huff is supervising testing for an important project at Tailspin Toys. The company is preparing to market a simple, robotic dog for small children. The hardware specifications have been set and the prototypes have been produced, but Tailspin is just getting started on the software that owners will use to customize their toy dogs.

Arlene is working closely with Stuart Railson, Tailspin's project manager. The pair is discussing important points to consider before meeting with the development team to begin preparing for their first sprint.

- 1. Which of the following measures the percentage of code that is tested?**
 - a. Code completion
 - b. Code coverage
 - c. Unit coverage
- 2. Which is the following is most likely to be Arlene's goal for that metric?**
 - a. 50%
 - b. 80%
 - c. 100%
- 3. Which technology can Arlene use to find bugs that are difficult to reproduce?**
 - a. CodeTrace
 - b. DataTrace
 - c. IntelliTrace

HINT 

Automated tests make it easy to collect data about the percentage of code that is tested.

Answers

1. Which of the following measures the percentage of code that is tested?
b. Code coverage
2. Which of the following is most likely to be Arlene's goal for that metric?
b. 80%
3. Which technology can Arlene use to find bugs that are difficult to reproduce??
c. IntelliTrace

Essential details

- **Code coverage** is a metric used to describe the degree to which the source code of a program has been tested and is typically expressed as a percentage of the blocks of code tested over the total blocks of code.
- 80% is a common code coverage goal for a development team, though 70% is not unusual.
- In Visual Studio®, code completion can be calculated based on code blocks (the default) or lines of code.
- **IntelliTrace** is a feature for debugging a managed application at specific points in time. It captures and records what the application does while it is running.
- When an error occurs, you can use IntelliTrace to view the state of the application at any time from the start to the point of the error.

FAST TRACK HELP

- **MSDN® Using Code Coverage to Determine How Much Code is being Tested**
<http://msdn.microsoft.com/en-us/library/dd537628.aspx>
- **MSDN Unit Testing: Testing the Inside**
<http://msdn.microsoft.com/library/jj159340.aspx>
- **MSDN Debug Your App by Recording Code Execution with IntelliTrace**
<http://msdn.microsoft.com/en-us/library/vstudio/dd264915.aspx>



WRITE AUTOMATION TESTS

SCENARIO: Litware, Inc. is developing a web application to organize and manage recreational basketball leagues, with functionality to add teams, schedule games, and track statistics and results. Jay Henningsen, the project manager, is focusing on the test plan and testing logistics while the rest of his team prepares for the next sprint.

He has configured a computer to emulate different operating systems so that he can thoroughly test the application. This process has revealed a potential bug that Jay will need to track down.

- 1. What did Jay create when he emulated a computer via software?**
 - a. A physical machine
 - b. A standard machine
 - c. A virtual machine
- 2. What strategy can Jay use that tests conditions only during debug builds?**
 - a. Assertion
 - b. Automation
 - c. Analytic
- 3. Which of the following is NOT a description of a testing error level?**
 - a. An assertion-level error
 - b. A test-level error
 - c. A run-level error

HINT

Automated tests run during the build of code—and can have errors just like any other code.

Answers

1. What did Jay create when he emulated a computer via software?
c. **A virtual machine**
2. What strategy can Jay use that tests conditions only during debug builds?
a. **Assertion**
3. Which of the following is NOT a description of a testing error level?
a. **An assertion-level error**

Essential details

- An **assertion** is a statement that tests a condition when a debug build of the program is executed. It is a debugging tool and is not executed in a release build.
- Assertions are particularly useful for tracking down and identifying logic errors.
- A **physical machine** is a hardware-based computer, as opposed to a software-based virtual machine. It's what we think of when we hear the general term "computer."
- A **virtual machine** is like a computer within a computer, implemented in software.
- A virtual machine emulates a complete hardware system, from processor to network card, in a self-contained, isolated software environment, enabling the simultaneous operation of otherwise incompatible operating systems.
- Virtual machines make it easier to test applications in different configurations without having to acquire and set up actual physical machines.

FAST TRACK HELP

- **MSDN® Assertions in Managed Code**
[http://msdn.microsoft.com/en-us/library/ttcc4x86\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ttcc4x86(v=vs.110).aspx)
- **MSDN Lab Environments**
<http://msdn.microsoft.com/en-us/library/jj159341.aspx>
- **MSDN Troubleshooting Test Execution**
<http://msdn.microsoft.com/en-us/library/ms182478.aspx>



MANAGE TEST SCRIPTS

SCENARIO: Litware, Inc., is developing a web application to organize and manage recreational basketball leagues, giving league commissioners the ability to easily add teams, schedule games, and track statistics and results. Because there are many complex features to this application, thorough and detailed testing is necessary. The responsibility for testing, and ultimately customer satisfaction, has been assigned to Jay Henningsen, the project manager.

To meet the demands of the task, Jay is employing a variety of strategies. He has selected an unused conference room as his testing room and has set up six different computers.

- 1. Which lab environment best describes Jay's setup?**
 - a. Standard lab
 - b. SCVMM lab
 - c. Virtual lab
- 2. Which of the following is another expression for a *smoke test*?**
 - a. Build Verification Test (BVT)
 - b. Coded User Interface Test (CUIT)
 - c. Virtual Machine (VM) test
- 3. Which tool will Jay use to control his testing environment?**
 - a. Visual Studio Lab Management
 - b. Visual Studio Test Manager
 - c. Team Foundation Server

HINT 

SCVMM stands for System Center Virtual Machine Manager.

Answers

1. Which lab environment best describes Jay's setup?
 - a. **Standard lab.** A standard lab environment can contain both physical and virtual machines.
2. Which of the following is another expression for a *smoke test*?
 - a. **Build Verification Test (BVT)**
3. Which tool will Jay use to control his testing environment?
 - a. **Visual Studio Lab Management**

Essential details

- **Build verification test (BVT)** are used to determine the success of a build.
- Typically, these tests exercise the core functionality to help team members determine whether further testing is worthwhile.
- A BVT is also known as a smoke test.
- A **lab environment** is a collection of computers that are managed as a single unit in which developers deploy the system under test along with test software.
- A lab environment can consist of two types of systems: a physical, hardware-based machine and a virtual machine implemented in software.
- A **virtual machine** emulates a complete hardware system, from processor to network card, in a self-contained, isolated software environment.
- A **standard lab environment** can contain a mix of virtual and physical machines.
- A **SCVMM lab environment** only contains virtual machines that are controlled by System Center Virtual Machine Manager (SCVMM). SCVMM offers management features not available in a standard environment.

FAST TRACK HELP

- **MSDN® How to: Configure and Run Scheduled Tests After Building Your Application**
[http://msdn.microsoft.com/en-us/library/ms182465\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms182465(v=vs.110).aspx)
- **MSDN Lab Environments**
<http://msdn.microsoft.com/en-us/library/jj159341.aspx>
- **MSDN Types of Performance Testing**
<http://msdn.microsoft.com/en-us/library/bb924357.aspx>

