UATB

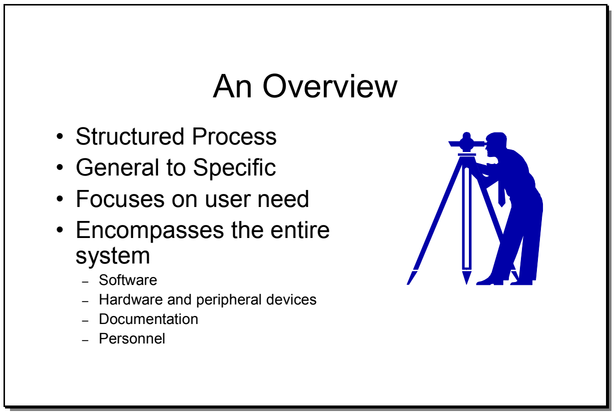
User Acceptance Test Planning

Objectives

- Learn an overview of the User Acceptance Testing Process
- Learn the steps involved in User Acceptance Testing
- Learn how to write the User Acceptance test plan, test cases, and test scripts.

Synopsis

Learn how to plan a complete user acceptance test.



An Overview

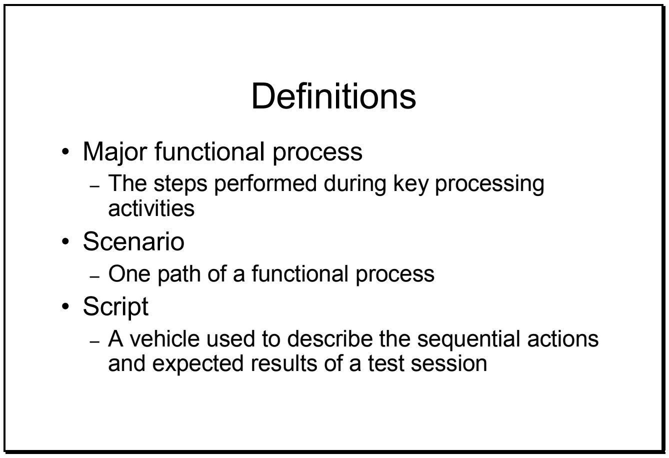
- Structured Process
- General to Specific
- Focuses on user need
- Encompasses the entire system
  - Software
  - Hardware and peripheral devices
  - Documentation
  - Personnel

An Overview

Structured User Acceptance Testing is designed to help users test systems from a functional or “black box” perspective. The strength of this technique is that it lets users design a test focusing on their business, not on the system requirements.

Sometimes systems are so large and complex that users do not know where to start in designing an acceptance test. The structured process allows the user to start at the big picture of the business and break it down into more manageable (and understandable) pieces.

Acceptance testing is more than just testing the software. By definition a system is made up of multiple components such as software, hardware, documentation, and the people that will be using the system. It is very important to be able to test the system from end to end. This might require test cases being performed in several systems to simulate how a person or thing is processed in the business.



Definitions

- Major functional process
  - The steps performed during key processing activities
- Scenario
  - One path of a functional process
- Script
  - A vehicle used to describe the sequential actions and expected results of a test session

Definitions

- **Major Functional Process**

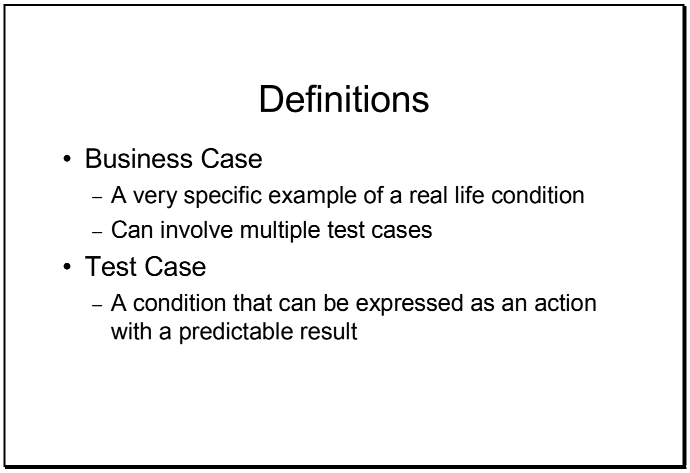
A major functional process describes the steps performed during key business activities. Some business processes are complex and may need to be broken down into several sub-processes. A business will likely have many major processes, depending on the size of the operation.

- **Scenario**

A scenario describes the steps performed during one variation or path of a business process.

- **Script**

A script describes the actions and expected results of a test session. Scripts are based on scenarios.



Definitions

- Business Case
  - A very specific example of a real life condition
  - Can involve multiple test cases
- Test Case
  - A condition that can be expressed as an action with a predictable result

Definitions

There are two critical parts of a test: the first is the process to be followed. This describes the steps to be followed. The second part is what will be put through the process.

For example, your business might have a process that takes orders from customers. A test script would describe the steps to be followed in testing the process. The customer and their order would be put through the process. In fact, there will likely be many customers with many orders that could be tested through the order process.

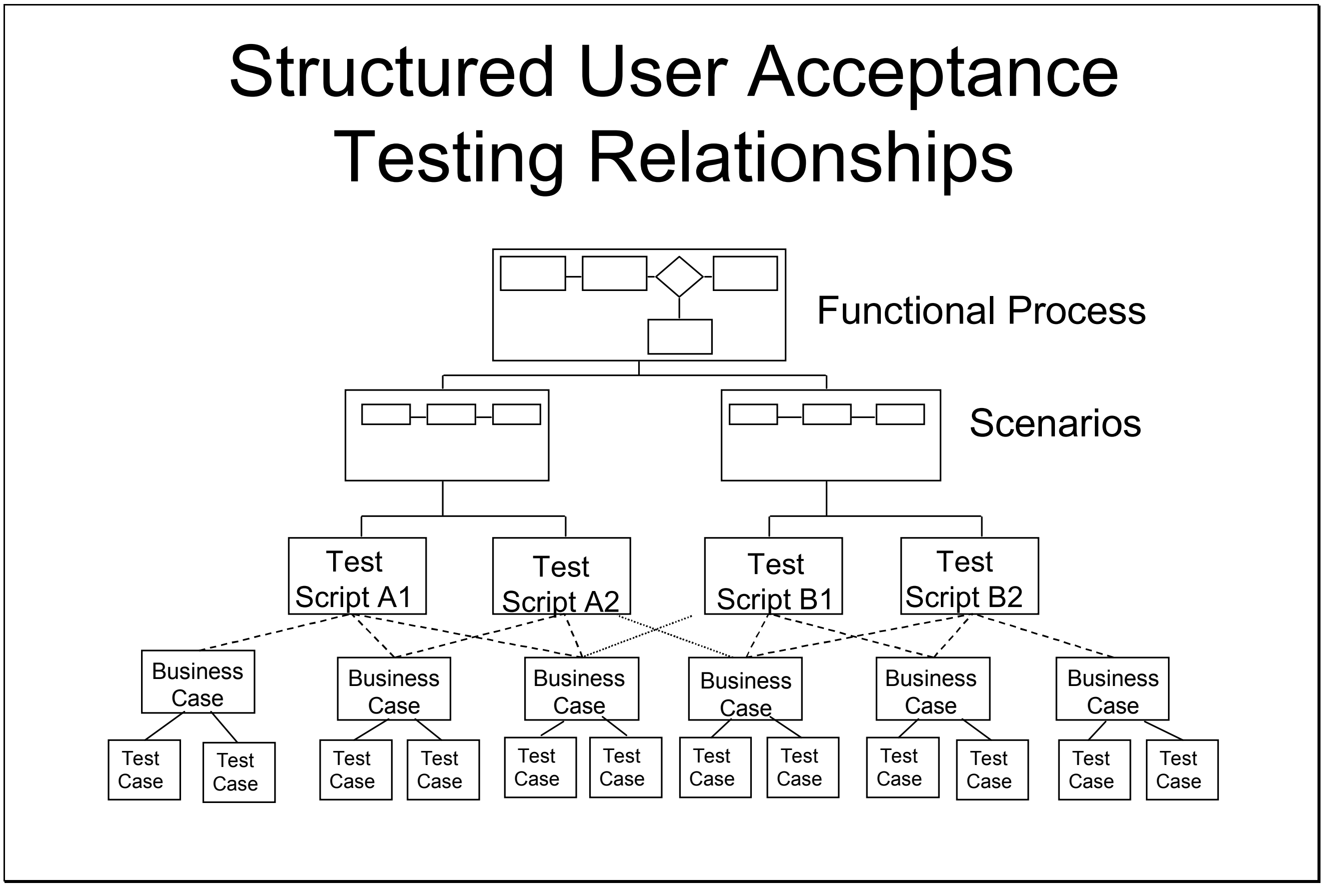
This concept describes two other definitions we need to know:

- **Business Case**

A business case is a person or thing processed through the business or operation. In the structured user acceptance testing process, business cases drive the process. A business case will be processed through many scripts and will simulate what happens to someone or something all the way through the business.

- **Test Case**

A test case is a condition that can be expressed by an action and a predictable result. A business case may have many values associated with it to test certain system functions. For example, a customer may have a customer code that takes an automatic 10% off the total amount of an order.



Structured User Acceptance Testing Relationships

Functional Process

Scenarios

Test Script A1

Test Script A2

Test Script B1

Test Script B2

Business Case

Business Case

Business Case

Business Case

Business Case

Business Case

Test Case

Test Case

Test Case

Test Case

Test Case

Test Case

Test Case

Test Case

Test Case

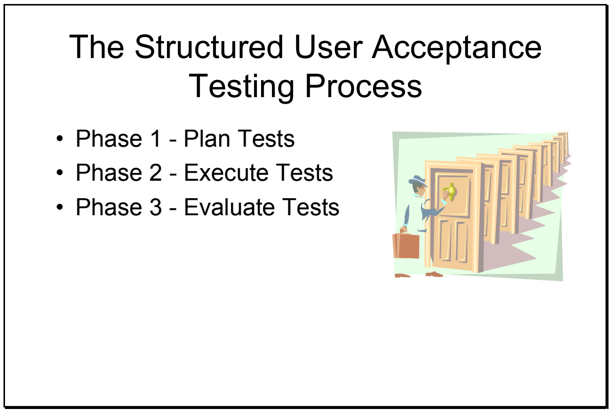
Test Case

Test Case

Test Case

Structured User Acceptance Testing Relationships

This picture shows how the structured user acceptance test process starts at the general, high-level, and decomposes to the lower, detailed level.



The Structured User Acceptance Testing Process

- Phase 1 - Plan Tests
- Phase 2 - Execute Tests
- Phase 3 - Evaluate Tests

The Structure User Acceptance Testing Process

There are three main phases in the structured user acceptance testing process:

Phase 1 - Plan Tests

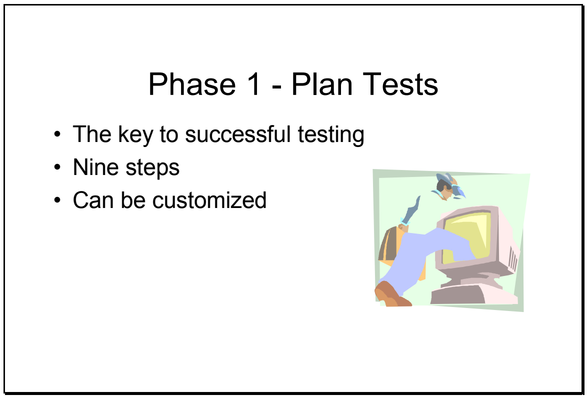
This is the most involved step and also the most critical. The better the plan, the easier the test becomes.

Phase 2 - Execute Tests

This is the step where you perform the tests designed in the planning phase.

Phase 3 - Evaluate Tests

Actually, phases 2 and 3 are performed in parallel. Test evaluation is performed each time a test is performed. Other test evaluation tasks include writing periodic status reports and the final acceptance test summary report.



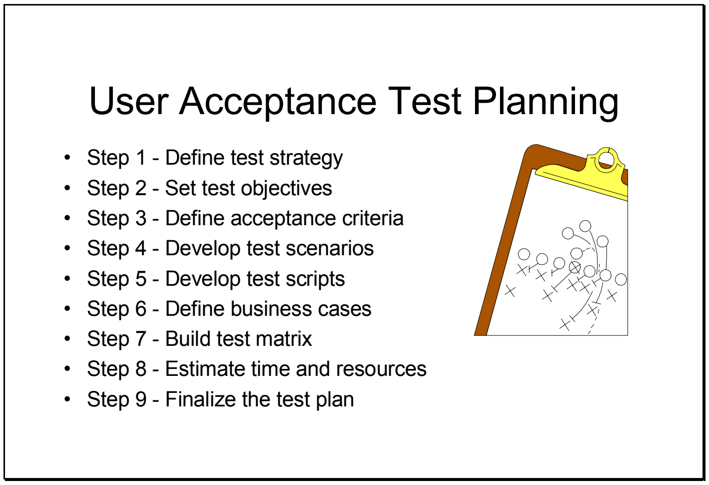
Phase 1 - Plan Tests

- The key to successful testing
- Nine steps
- Can be customized

Phase 1-Plan Test

The planning phase has nine steps. Each of the nine steps has a specific purpose to achieve a rigorous user acceptance test. If these nine steps appear to be overwhelming, just keep in mind that the better the plan, the easier the test becomes.

The planning process described in this course is flexible and can be customized to fit your own organization's needs. The determination of what to add, change or delete in the process should be driven by the relative risk of the project.



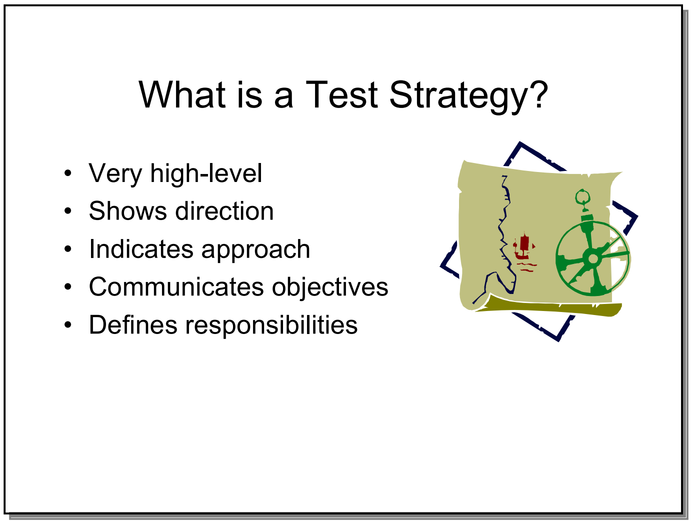
User Acceptance Test Planning

- Step 1 - Define test strategy
- Step 2 - Set test objectives
- Step 3 - Define acceptance criteria
- Step 4 - Develop test scenarios
- Step 5 - Develop test scripts
- Step 6 - Define business cases
- Step 7 - Build test matrix
- Step 8 - Estimate time and resources
- Step 9 - Finalize the test plan

User Acceptance Test Planning

Structured user acceptance test planning process is comprised of the following nine steps, which will be explained in more detail in this module.

- **Step 1 - Define test strategy**
- **Step 2 - Set test objectives**
- **Step 3 - Define acceptance criteria**
- **Step 4 - Develop test scenarios**
- **Step 5 - Develop test scripts**
- **Step 6 - Define business cases**
- **Step 7 - Build test matrix**
- **Step 8 - Estimate time and resources**
- **Step 9 - Finalize the test plan**



What is a Test Strategy?

- Very high-level
- Shows direction
- Indicates approach
- Communicates objectives
- Defines responsibilities

Step 1 – Determine Test Strategy and Test Objectives

What is a Test Strategy?

The test strategy is the foundation of the test plan and can be written very early in the project. Before you can start building the test plan, you must first define your test strategy, or approach. There are several reasons for this:

- Test strategy is **a tool for communication** to the entire organization.
- Test strategy **makes test planning easier** by resolving some of the basic testing issues early.
- Test strategy **helps you understand the project** and the system you are planning to test.
- Test strategy **helps you understand the project's risks** and critical success factors.

The test strategy is the foundation of the test plan and can be written very early in the project.

A test strategy is a document that indicates a general direction or approach for testing. The test strategy is a foundational document for test planning and can usually be developed early in most projects.

Some characteristics and purposes of test strategy are:

- **Very high-level**

If the test plan is the “big picture”, the test strategy is the frame around the big picture.

- **Shows direction**

By defining scope, purpose and objectives, the reader can tell what is to be accomplished in testing.

Example: A test strategy may indicate that user acceptance testing will be a key part of testing. This indicates that users will need to be lined up in advance for this type of testing.

- **Indicates approach**

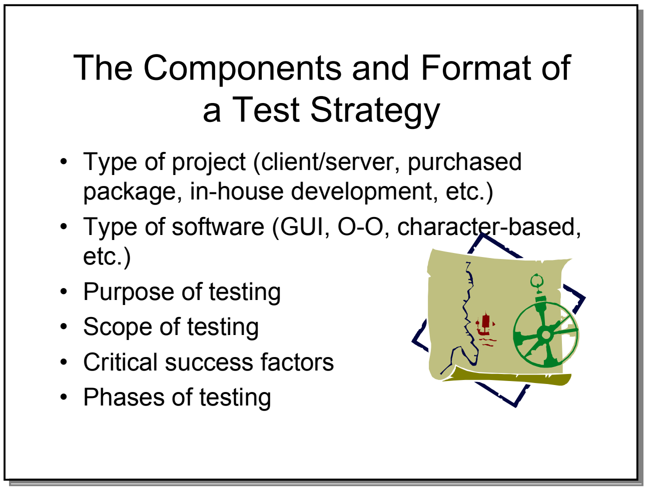
Example: A test strategy may indicate that automated test tools will be used heavily in the test. This means that if tools have not been used in the past, new test environment consideration will be needed.

- **Communicates objectives**

This helps reduce confusion and assumptions as to what testing will accomplish and the timeframes expected for testing.

- **Defines responsibilities**

This clearly defines the types of people needed for various testing activities and what their responsibilities will be.



The Components and Format of a Test Strategy

- Type of project (client/server, purchased package, in-house development, etc.)
- Type of software (GUI, O-O, character-based, etc.)
- Purpose of testing
- Scope of testing
- Critical success factors
- Phases of testing

The Components and Format of a Test Strategy

- **Type of project (client/server, purchased package, in-house development, etc.)**

This will be one of the most critical factors in determining a test strategy.

Examples

Project Type	**Test Strategy**
Client/Server	End-to-end functional tests that include hardware, software, networks, servers, GUIs, and procedures. Use of automated tools for rapid testing. Test cases based on solid specs and requirements.
Purchased Package	Black box functional testing. System testing to validate technical requirements, User acceptance testing to validate customer requirements.
In-house Developed	White box structural testing to validate coding structure, black box functional testing to validate system functionality.

It should be noted that many projects will blend two or more project types. For example, a company may purchase a client/server system as opposed to building their own client/server system.

Other types of projects include: Object-oriented, web-based, component-based, traditional waterfall, prototyping, and maintenance.

- **Type of software (GUI, O-O, character-based, etc.)**

There are many different types of software that each has unique testing considerations. Examples include GUIs, Object-oriented, component-based, traditional online transactional (such as CICS), batch, real-time, etc.

Software Type	Test Strategy
GUIs	Test case descriptions that validate requirements, use of automated tools to facilitate rapid changes.
Object-oriented	Heavy emphasis on unit and module testing for reuse considerations. Integration testing for validating compatibility.
Batch	Black box functional testing based on a reusable test bed of data that reflect test cases.

- **Purpose of testing**

Most testing experts agree that the most rigorous test is the one designed to push the system to its limits for the purpose of finding defects. It is relatively easy to test the conditions you know will work to prove the system works. The exception to this philosophy is in user acceptance testing, where the user seeks to prove the system will work in the real world by supporting business processes. Even then, test cases can be designed to fully exercise the system.

- **Scope of testing**

This defines in a broad sense which parts or functions of the system will and will not be tested. This is one of the most critical aspects of defining the testing strategy. If you set the scope too large, you will not have time to complete the test. If you set the scope too small, you will not achieve the coverage levels needed to have a high level of confidence in the test results.

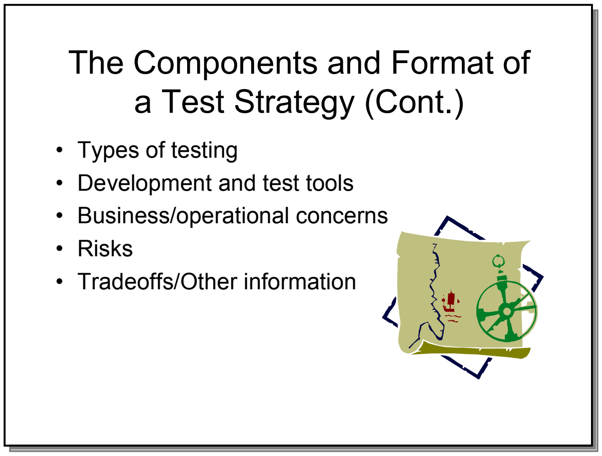
- **Critical success factors**

These are the attributes that the system must possess to be considered a success. Failure in any of these areas can overshadow the others. For example, the system might work fine functionally, but have a response time of 30 seconds. This will be highly frustrating to users and will not be workable in the business.

A list of critical success factors will be shown later in this module.

- **Phases of testing**

This simply indicates which phases or levels of testing will be needed. Examples include unit, integration, system, and user acceptance testing.



The Components and Format of a Test Strategy (Cont.)

- Types of testing
- Development and test tools
- Business/operational concerns
- Risks
- Tradeoffs/Other information

The Components and Format of a Test Strategy (Cont.)

Other items that should be addressed in the test strategy include:

- **Types of testing**

This can include regression testing, structural testing, functional testing, stress and performance testing, and conversion testing.

- **Development and test tools**

Development tools are important to consider because they might require certain test tools. The test tools can include both manual and automated tools.

- **Business/operational concerns**

These are the external and many times “non system” concerns that will impact the project and should be considered from the testing standpoint.

- **Risks**

These are the possible failure points in the system and the business. The degree of testing should be relative to the degree of risk the system poses to the business. A high-risk system may cause injury to people or the loss of large amounts of money.

The way to tell the degree of risk for a system is to perform a risk assessment. A risk assessment is an objective to identify and score the risks on a project. There are several risk assessment processes available and tools for performing risk assessments.

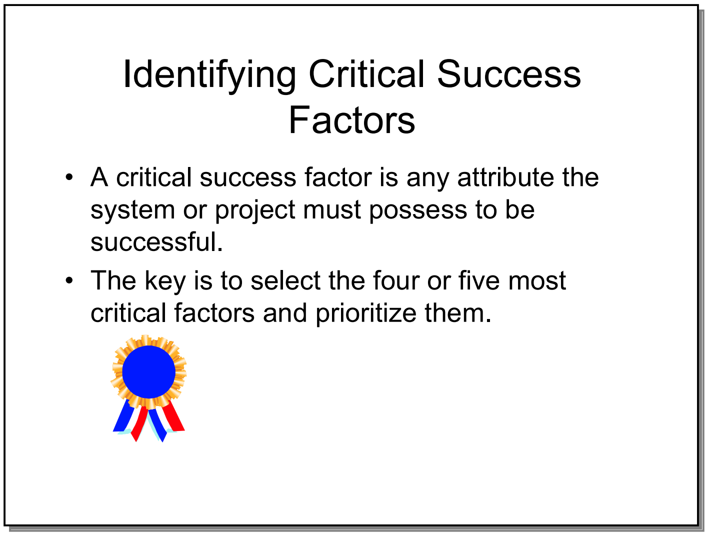
- **Tradeoffs/Other information**

There’s an old saying that says, “Cost, schedule, quality. Pick two.” In other words, it is very difficult to complete a project on schedule, within cost, with high levels of quality. What normally happens is that one of the three factors starts to increase, requiring compensation in one or more of the other factors. For example, if the size (scope) of the project increases and we still aim for the same schedule, we must add people or cut quality.

Knowing the tradeoffs will enable you to prepare for these contingencies.

Who Will Conduct Testing?

In user acceptance testing, the users should “own” the testing effort. This means that the test should be designed and executed by users. Users should make the final acceptance decision. There may be times when users need help from developers to perform special system operations.

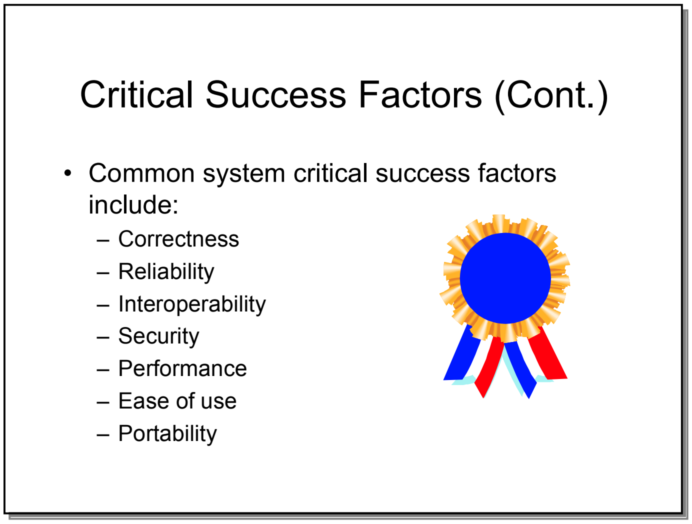


Identifying Critical Success Factors

- A critical success factor is any attribute the system or project must possess to be successful.
- The key is to select the four or five most critical factors and prioritize them.

Identifying Critical Success Factors

A critical success factor is any attribute the system or project must possess to be successful. The key is to select the four or five most critical factors and prioritize them.



Critical Success Factors (Cont.)

- Common system critical success factors include:
  - Correctness
  - Reliability
  - Interoperability
  - Security
  - Performance
  - Ease of use
  - Portability

Critical Success Factors (Cont.)

This list is not exhaustive, but common critical success factors can include:

- **Correctness**

The ability of the system to deliver correct results and functionality.

- **Reliability**

The ability of the system to deliver correct results and functionality over an extended period of time.

- **Interoperability**

The ability of the system to effectively integrate data, software, and hardware.

- **Security**

The ability of the system to restrict access to authorized users and to control transactions that have loss potential.

- **Performance**

The ability of the system to respond and process transactions at stated performance levels.

- **Ease of use**

The ability of the system to be easily used.

- **Portability**

The ability to locate the system on multiple computing platforms.

CASE STUDY

The XYZ Company is migrating its customer service function from a client/server application to a web-based application. There is also an interface to a mainframe batch billing system that will remain intact on the mainframe side. The functionality for the web-based application is simple: Add, change, delete and query customer information. Customer letters and reports will continue to be generated from the batch legacy system.

There have been performance problems in the past, which have caused customer complaints. Customer information contains credit history, bank account numbers, and credit card information.

The new web applications will be developed in Microsoft Front Page, based on existing interface design and specifications that are being developed. An initial set of requirements has been developed from which the specifications will be based.

The development will be outsourced to a web development company.

The company currently owns five copies of a web-based capture/playback tool. They plan to acquire a web-based load-testing tool for this project.

Sample Testing Strategy Worksheet

Type of Computing Environments Mainframe, web-based, client/server, standalone PCs
Purpose of Testing To find defects as defined by non-conformance to specifications and requirements.
Type of Software GUI (web-based), Character-based (client/server), Real-time (PCs), Batch (nightly processing)
Scope of Testing 1. Add customer 2. Modify customer 3. Delete customer 4. Query customer 5. Correspondence 6. Customer reports
Critical Success Factors • Correctness • Performance • Security
Phases of Testing Unit, System, and User Acceptance
Audience Internal corporate users
Tradeoffs • Schedule • Cost
Types of Testing • Functional • Regression • Stress & Performance
Development Tools and Test Tools (e.g., GUI builders, automated capture/playback, etc.) • Development Tools: Customer Info System – MS Front Page, Billing - CASEWORKS • Test Tools: Web-based Capture/Playback tool and load test tool.
Business/operational concerns • Customer information must remain accurate, especially in terms of dates and automatic maintenance that occurs based on date intervals. • Customer service response times must not degrade from their current levels. • Billing information must be accurate. • Billing run times must not increase. • Outsourced development needs to be managed to ensure specifications and requirements are met. In addition, the vendor must be available to fix defects and provide ongoing maintenance.
Risks Inaccurate customer information could result in erroneous cancellations, excessive discounts, and customer dissatisfaction. Customers could take business to competitors. Inaccurate billing information could result in a financial loss to the company, financial loss to the customers Degraded system performance could lead to longer wait times for customer service. Degraded batch run times could delay other critical processing.
Other The customer service and financial services areas have volunteered to help in the testing effort. This is good, but will require training and facilitation.

Sample Testing Strategy Worksheet

Project Phase	Testing Phase	Stakeholders	Purpose/Why
Definition of business need.	Review of project and system objectives.	Software developers Business client/customers Sr. IT Mgt. Sr. Business Unit Mgt. QA	To ensure that the correct project and system objectives have been established. To ensure that a realistic project approach is planned. To ensure that an adequate process is in place to validate the system.
Requirements	Review of requirements	Software developers Testers Business client/customer QA	To ensure that the requirements are complete, accurate, clear, and testable.
Design/Prototyping	Review of design	Software developers Testers Business client/customer QA	To ensure that the design reflects the defined requirements and system objectives.
	Usability testing Code reviews	Testers Business client/customer	To validate that the interfaces are easy to use before major construction begins.
Construction	Unit testing	Software developers Testers QA	To ensure that the software functionality is correct.
	Regression testing	Software developers Testers QA	As new versions of the system are generated, ensure that new defects have not been created.
	System testing	Software developers Testers QA	To ensure that the system functions correctly as a whole, that all critical internal and external interfaces function correctly as defined in requirements.
Integration/Promotion	User Acceptance Testing	Business Users QA	To ensure that the system will work correctly in the business.
Implementation	Post-implementation Reviews	Software developers Business client/customers Sr. IT Mgt. Sr. Business Unit Mgt. QA	To assess the adequacy of the software development and testing process.