Managing technology risk

Software
User Acceptance Testing (UAT)
Strategy Development

May 30, 2014

Agenda

- UAT Strategy Development
  - UAT Principles and Objectives

- UAT Strategy Considerations

- The Top Ten Reasons UAT Fails

UAT Principles and Objectives

- **Objective is confidence in and acceptance of the product to be implemented**
- **Objective IS NOT a defect-free system**
  - Users understand that some bug fixes will be prioritized and addressed post-implementation.
  - Focus must remain on functional abilities and critical paths (risk-based testing).
- **UAT IS NOT a second run at system testing**
  - Everything that will be tested in UAT should be tested in integration and system testing first.
- **UAT IS the chance for users to confirm that they can perform their jobs using the new system**
  - Validates that a system of products is of sufficient quality to be accepted by the users.
  - Validates that requirements were given, documented, interpreted, designed, and developed correctly.

Managing technology risk

UAT Strategy Considerations

- ❖ **Definitions**
- ❖ **Scope**
- ❖ **Entry and Exit Criteria**
- ❖ **Participants**
- ❖ **Incident Reporting and Management**
- ❖ **S/W Release Configuration Management**
- ❖ **Risks**

Managing technology risk

UAT Strategy Considerations

- **Definitions**
  - Define what UAT is and what it isn't
  - UAT Purpose, Scope, Success Criteria, and Exit Criteria must clearly embrace the project's Goals and Objectives
- **Examples:**
  - UAT is a form of testing to verify the system can support day-to-day business and user scenarios; to validate rules, various workflows, and overall software system fitness for use; and to ensure the system is sufficient and correct for business usage
  - UAT Strategy documents the approach that will be used to verify and ensure the application successfully supports its business requirements
  - UAT Test Cases identify sequences of steps to test the correct behavior of an application for specific scenario objectives

UAT Strategy Considerations

Managing technology risk

- **Scope**
  - It’s important to document what is and is not in scope for the testing
  - The focus of UAT is workflow and business process testing – does the system meet the needs/requirements of the users – and the scope should clearly identify what is and isn’t included
- **Examples of areas that are in scope are:**
  - Workflow/Use Case testing (basic and exception flows to comprise the majority of the tests with later focus on alternate flows)
  - User Interface (are all user and system interfaces functional to demonstrate end-to-end system functionality in supporting the user needs)
  - Security (can the users get to the functionality they need and are they prevented from getting to functionality they shouldn’t have)

UAT Strategy Considerations

- **Entry/Exit Criteria**
  - Document reasonable and measurable exit criteria from System Test and entrance criteria to UAT, and abide them
  - Do not waiver
  - Rushing into a test phase before entry criteria are met does not save time; rather, it usually costs time as stability bugs prevent forward progress
  - Declaring a test phase complete prior to validating exit criteria perpetuates an immature system and, in the case of UAT, results in discovery of additional issues after the system is moved to production

UAT Strategy Considerations

- **Sample UAT Entrance Criteria**
  - UAT Plan is approved by the system owner, user groups, and project stakeholders (Who, what, where, when, how)
  - UAT test scenarios are documented and agreed upon by responsible parties
  - All System test cases (Functional, Performance, and Interface) have been executed and all required functionality has been successfully demonstrated
  - No open critical or showstopper System Test defects (Severity level 1 or 2 defects)
  - UAT environment is up and running -- all applications and test data have been loaded. (Utilize converted data)
  - All users have logins/passwords
  - All users have received training on the new system
  - Resources are sufficient and available to support test execution and issue resolution

UAT Strategy Considerations

Managing technology risk

- **Sample UAT Exit Criteria**
  - All UAT Test Cases have been executed and passed
  - All showstopper and critical defects have been resolved and retested (No severity level 1 or 2 defects)
  - Any remaining severity 3 or 4 defects have been prioritized and placed in a build schedule
  - All business processes can be carried out with an acceptable level of effectiveness using the new system
  - Any overlooked requirements have been negotiated and a plan is in place for addressing them
  - UAT sign off by business owners
  - Successful executive go/no-go decision for pilot or launch

Managing technology risk

UAT Strategy Considerations

- **Testing Environment Definition**
  - Define software, hardware, interfaces, networks, physical space and any other details that are required for an effective UAT effort
- **Test Cases**
  - The strategy should contain a high level outline of the functionality being tested as well as the individual test scenarios that will be run for each piece of functionality
  - The actual Test Cases are not part of the strategy document
- **Participants**
  - Identify the Development and User participants early
  - Define participant roles and responsibilities as well as reporting structure, if necessary
  - Keep participants in the loop as the test strategy develops in order to minimize last minute omissions

UAT Strategy Considerations

- **Incident Reporting/Management**
  - Define the process for reporting defects, prioritizing them, tracking them and correcting them
  - As with the scope section, define which category of bugs will receive immediate attention (i.e. need this fixed to continue UAT) and define which bugs may be deferred to a later phase of development
- **S/W Release Configuration Management**
  - Define the process for tracking and implementing software versions throughout the test effort
  - How often will new builds be deployed?
  - Will the data remain stable between builds? How will the test database be refreshed, if required? Will users lose all their test data to date or will they be able to continue with the previous data?
- **Risks**
  - Every strategy should contain a section on risks, complete with proposed mitigations. Figure out what can go wrong and plan for it
  - Look at each section in the strategy document and examine the possible risks associated with that section

Managing technology risk

Top Ten Reasons UAT Fails

UAT Strategy Development

#10 - Expecting Real Life Business Process Testing With Less Than Real Life Data

❖ Business Scenario

- ❖ Effective UAT requires live or near-live data. Too often, the UAT data deck is insufficient to allow for multiple passes through the test cases. Nothing brings the UAT process to a standstill more often than lack of usable data

- ❖ Often, only a portion of the required data is made available for UAT. In order to execute the UAT scripts, every user winds up having to start at square one – setting up a user account, creating new records from scratch. There is not enough data in various stages of completeness to allow them to test from different business process starting points

❖ Recommendations

- ❖ Load sufficient production data to allow complete end-to-end UAT. When test data is used instead of production data, there are often not enough user accounts or existing data records to allow for multiple UAT test runs

- ❖ Test the UAT environment – take various scenarios from various points in the process and ensure there is sufficient data for them to run

#9 - Process, Expectations and Outcomes are Not Clearly Defined

Managing technology risk

- **Business Scenario**
  - UAT is almost an afterthought following system testing.
  - A comprehensive UAT strategy was not developed. There is no plan for who will do the testing, how they will arrange coverage in their jobs while they participate, how they will be trained, and how the Organization will keep them motivated, positive, and effective throughout the test cycle.
  - Users are not educated as to the goals and objectives of UAT. They don't know what to expect; therefore, they create their own, often unrealistic, expectations.
  - Users feel undue pressure for their performance, thinking that if they find too many bugs or don't test enough, they will be responsible for the delay or failure of the project.
- **Recommendations**
  - Get the Development Team and the Users involved in UAT strategy and discussions early and often throughout development.
  - Avoid putting "go-live" pressure on the testers. Make sure they know that their responsibility is to validate the business functionality of the system and report their findings, and that the rest is up to others.
  - Make sure the users know that they will be getting a fully tested system to test; that their job isn't to test every single field, but to use the system and determine if they can do the expected functions.

#8 - Exit and Entry Criteria are not Adhered To

- **Business Scenario**
  - Entry/exit criteria are either not defined or are treated as guidelines or suggestions rather than solid process points
  - Entry/exit criteria aren’t realistic or measurable; they are either ignored because they can’t be met or they hold up the schedule unnecessarily because the participants can’t agree on whether or not they have been satisfied
  - Schedule takes priority over process -- UAT is started prior to the environment being fully prepared or before system testing is completed because the Users’ schedules can’t be shifted without a serious interruption in workflow or because the deployment date is rigid
- **Recommendations**
  - Define clear, realistic, measurable criteria. For example, “Defect-free system test conclusion” is unrealistic; “Completed system testing with no showstoppers remaining” is realistic and necessary in order to provide a stable UAT code base
  - Ensure that all participants agree on the criteria and are committed to making any and all necessary effort to adhere to them
  - Have a solid reporting/decision making strategy in place to address issues

#7 - Failure to Select the Right People for the Job

- **Business Scenario**
  - The Organization relies on the same small subset of users that contributed to the original requirements
  - Only the “power users” are selected for UAT with the anticipation that the testing process will progress more quickly. Less involved users are overlooked, which leads to a skewed perception of how usable the new system will be to the average user
  - The misperception that UAT is designed to test every condition and validate every field, UAT participants are selected from the testing teams. This actually detracts from the effectiveness of UAT, as their focus is too narrow rather than being on the business process
- **Recommendations**
  - Don’t constrain the UAT pool to the SMEs that helped develop the requirements
  - Make use of customer service representatives, help desk, and other people that are familiar with the problem areas in the business process
  - Bring in users from other offices, bureaus, regions to get the greatest possible representation for how the system will be used
  - Include users that are less proficient in order to get a clear indication of how easy or complicated the new system will be to use

#6 - Expecting Users to Test like Testers

❖ Business Scenario

- ❖ UA test scripts are modeled after the system test scripts rather than focusing on the business process
- ❖ Users receive no training in the testing process, resulting in slower testing and more re-testing
- ❖ Users think that User Acceptance Testing is really Usability Testing and spend too much time on usability suggestions, resulting in a disproportionate amount of low-priority, cosmetic bug reports

❖ Recommendations

- ❖ Train the users *prior to the start of UAT* in how to use the scripts and how to report bugs. Don’t assume they know how to test
- ❖ Let the users be users – The value they bring to the table is their knowledge of the business processes – capitalize on this by making their testing experience mimic their “real jobs” as much as possible
- ❖ Ensure that the UAT scripts are business-centric and focused on the business processes rather than the system-level validation of previous test phases

# 5- Test Scenarios/Scripts did not include input from End Users

- **Business Scenario**
  - Businesses try to capitalize on the massive cost and effort expended in system testing by reusing scripts for UAT. In fact, this may increase costs by increasing the difficulty level for the users
  - It will also predispose the users to finding the same types of bugs that were identified in system testing rather than focusing on validating the business process functionality
  - Users are given all the functional requirements included in the system without any prioritization
  - They are not consulted on which pieces of the functionality comprise the majority of their job or asked which functions are the most critical to the business.
- **Recommendations**
  - Resist identifying and creating the UAT scenarios prior to identifying the UAT participants
  - Once the UAT participants have been identified, get them involved in test scenario identification and development
  - These are the people who know what they expect the system to do – Listen to them and let them guide in structuring and prioritizing the test scenarios

#4 - Interfaces, Data Conversion and Reports

- **Business Scenario**
  - Interface partner testing has not been established
  - Point-to-Point transmission and receipt of data has not been thoroughly tested. Instead a “simulated” send and received is conducted
  - The system is not thoroughly tested with cleansed converted data. Again “simulated data is used that does not truly reflect converted data
  - Mock data conversion are not used in time to be proven in UAT
  - Report development is typically not complete at the start of UAT
- **Recommendations**
  - Provide plenty of notification to interface partners and ensure full partner testing for critical interfaces
  - Do not allow “simulations” to be part of UAT
  - Mock conversion and data cleansing activities should be a part of the UAT process

Managing technology risk

#3 - UAT Environment Late and/or Not Identical to Production

- **Business Scenario**
  - Businesses underestimate the time involved in setting up a UAT environment and start too late in the process
  - Typically, there is some unique configuration to the Production environment that is not considered when setting up the UAT environment
  - This can include operating system and other software versions, special security factors that exist only in Production, interfaces with internal and external sources
- **Recommendations**
  - Create a Technical Environment Plan that addresses all the issues in the Production environment and defines how they will be replicated in the UAT environment
  - Start early
  - Get the right software versions and ensure all the licensing is adequate

Managing technology risk

#2 - Test Data and Test Processes Not Stable

- ## Business Scenario
  - Test Data has not been adequately prepared to be used in a repeatable form
  - Testers are not sufficiently familiar with the test data, test data cycles and business rules
  - Test Environments are not stable and cause a stop and go process environment for the Testers
  - Tester do not understand the entire testing process including logging and following up on defects in a timely manner
- ## Recommendations
  - The creation of test data should be determined and exercised well in advance of UAT
  - Test beds should be created with the intent of re-using the same data to support controlled testing and to be able to clearly replicate a defect using test data

Managing technology risk

#1 - Incomplete Systems Integration Test

- **Business Scenario**
  - Typically there is some phase of Systems Integration that is not completed on schedule
  - Project Management decides to press on with User Acceptance Test (UAT) in parallel with Systems Test (ST) or
  - UAT is delayed pending completion of ST but the UAT schedule does not change resulting in a compressed UAT effort
  - The compressed UAT schedule does not allow sufficient time to detect and correct critical defects
  - The application goes live with an incomplete UAT; production data is compromised, help desk calls are excessive, the defect process is bogged down
- **Recommendations**
  - Carefully monitor the ST schedule and test defect rates and adjust the UAT schedule accordingly
  - Hold firm on ST exit and UAT entrance criteria
  - Do not allow UAT to start out with a compressed schedule

Contact information

- **Steven Esposito**

  Senior Vice President - Government Solutions

  sesposito@sliglobalsolutions.com

  303 575 6881 x 118

- **Robert Tafoya, PMP, CRISC**
- Director of Operations - Government Solutions
- rtafoya@sliglobalsolutions.com
- 720.975.9210

Managing technology risk