dea White Papers

Doreen Evans Associates, Inc.

Leveraging Analysis for User Acceptance Testing

As systems become increasingly complex and a greater number of business critical process are being automated, User Acceptance Testing (UAT) has become more than a simple check once development is complete. UAT is now a detailed analysis which focuses on ensuring that requirements are being implemented and that the final system actually meets the original user need*. Recent market research has discovered that 60% of technology projects fail to meet all user needs. Well defined and executed user acceptance testing is a powerful tool which organizations can use to ensure that their technology projects effectively meet the end users' needs.

Traditionally, user acceptance testing has been relegated to the end of a project's development lifecycle. In most cases not enough time is scheduled and even more is taken away as the design and development phases drag out and project completion dates stay the same. And it is not uncommon for even basic development projects to contain hundreds of requirements captured upon pages of specifications making the task even more challenging.

As the regulatory demands placed on business, such as Sarbanes Oxley and HIPA, increase so do the pressures placed on system development projects that service these organizations. The need to thoroughly test all requirements in a shortened time frame has prompted the development of new techniques to improve the testing phase of the system development lifecycle. One successful approach has been to leverage system use cases, developed during the analysis phase to create Design Use Cases. This new breed of use case captures additional post development system details which then can be used to drive the automatic generation of UAT scripts. Currently, 76% of test cases are generated by hand from requirements specifications**. Automated generation of test scripts can substantially lessen the time commitment required and the resources needed, and can improve the accuracy of user acceptance testing.

Use Cases – More then Defining System Functionality

Use cases have become commonplace in many system design projects. They are most often used to capture an architectural view of the system and describe system functionality during the analysis phase of a project. However, once a project has moved past early stages of the System Development Life Cycle (SDLC), they are often placed on a shelf and seldom referred to again. The new approach to UAT development, outlined within this paper, leverages and refines use cases that have already been developed for use in the automatic generation of UAT scenarios. This is made possible due to a unique multi-level methodology for developing use cases. This methodology develops three separate levels of use cases, the Business Use Case, the Solution Use Case, and the Design Use Case. At each use case level additional detail is added, expanding the view of the system. The final level, the Design Use Case, represents the true desired system design which can be used as a source for the generation of user acceptance testing scenarios. For additional information on the development of multi level use cases please see the white paper: Obtaining Continuous Return on your Use Case Investment

Capturing System Detail

Proper system testing relies heavily on the availability of accurate information provided in the proper context to the system tester. The sources of such system design information are varied and can include: requirements specifications, architecture products, functional design documents, data schemas, and user interface mockups, among others.

*VokeStream, *Market Snapshot Report: The Role of the Business Analyst* (Voke Media LLC., 2008), 14
**VokeStream, *Market Snapshot Report: The Role of the Business Analyst* (Voke Media LLC., 2008), 11

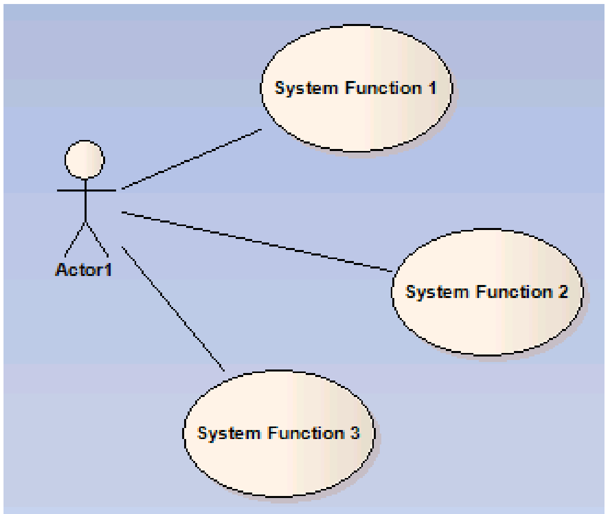


Figure 3 – Use Case Diagram

Capturing use cases and other system details in a proper UML modeling tool is crucial to successfully leveraging them as part of acceptance testing. Testing should focus on anticipated usage scenarios and should consider key users as well as use cases when evaluating software's acceptability*. Sparx System's Enterprise Architect provides the ability to easily and accurately capture uses cases in UML and relate them to the proper actors using use case and activity diagrams. Enterprise Architect is also capable of capturing and maintaining a variety of additional system design information such as: requirements, data models, and user interface mockups. Each of these design elements represents a piece of the puzzle, which when combined, forms the basis for our user acceptance testing scripts.

Many times what is absent from system testing is a high level understanding of how each of the separate parts of the system or the system functionality relate to each other in order to form an end product. To combat this knowledge gap, the refinement of use cases for the generation of testing scripts includes the development of use case diagrams. These high level representations of system function highlight the interaction of relevant actors and flows of functionality, represented by each use case. Placing the system functionality being tested in the proper context leads to greater system understanding. Better informed testers are able to identify errors and functionality gaps sooner and reduce the cost of rework.

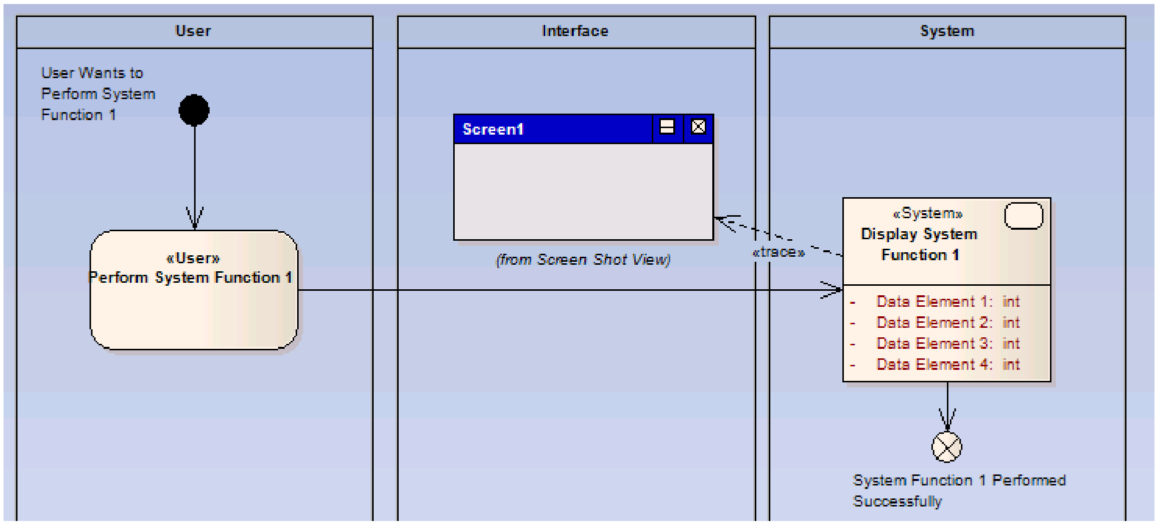

User
Interface
System
User Wants to Perform System Function 1
Screen1
«User» Perform System Function 1
(from Screen Shot View)
«trace»
«System» Display System Function 1
- Data Element 1: int
- Data Element 2: int
- Data Element 3: int
- Data Element 4: int
System Function 1 Performed Successfully

Figure 4 – Activity Diagram with Interfaces (Screen shots) and Related Data

*Wiegers, Karl E., *Software Requirements Second Edition* (Microsoft Press 2003), 280

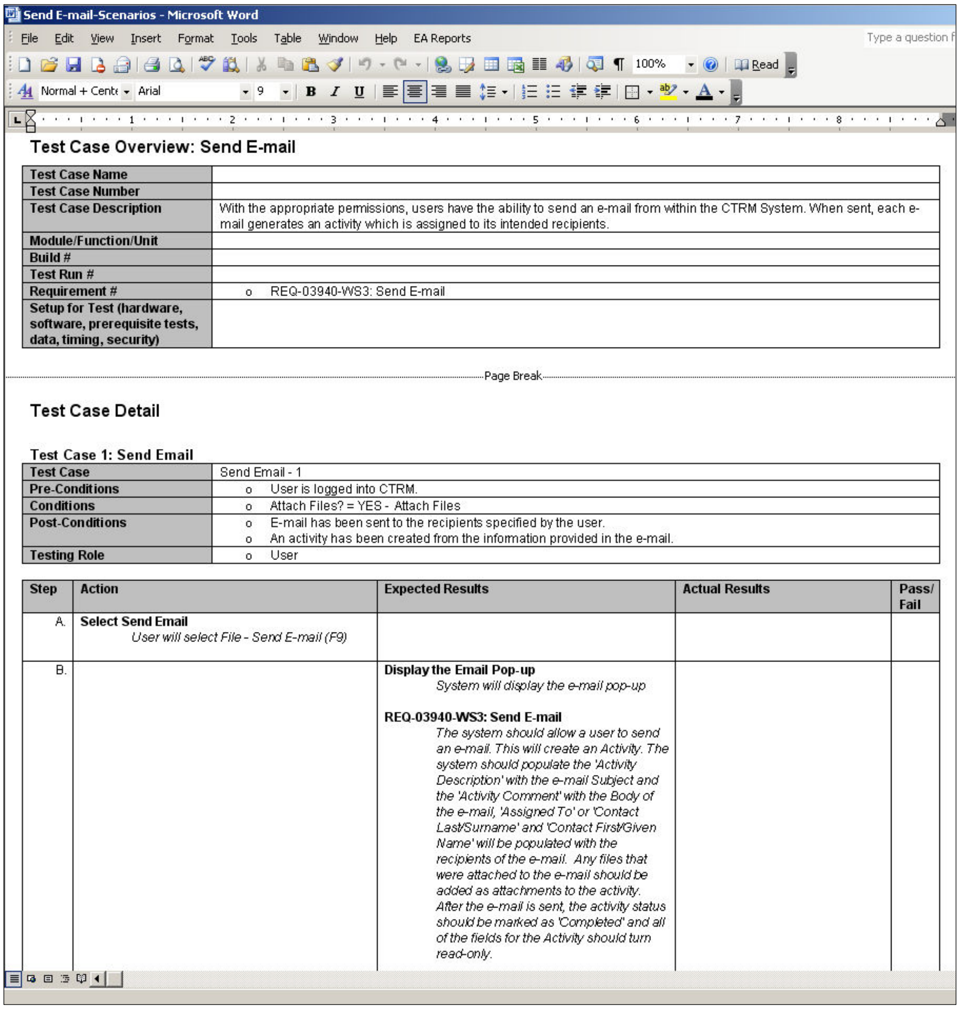
In order to further decompose the system design, activity diagrams are used to describe the detailed steps of each use case. Each diagram is used to represent all possible flows (primary, alternate, and exception) stemming from a use case, capture the interaction between actors, and serve as the linking point to additional system design information.

For the purposes of UAT the activity diagram needs to be expanded to include user interface mock ups, data elements, as well as system requirements. Using Enterprise Architect it is possible to enhance the standard diagram and embed each of these design elements directly into the diagram and link them to the appropriate activities. This presents the user with a total design view and provides an accurate system baseline which represents current system design and functionality.

The Right Information in the Right Place

The key to improving user acceptance testing while decreasing the time required for it lies in the use of the activity diagrams described above. These diagrams, which contain all key system design information, serve as the roadmap for the automated development of testing scripts. By leveraging Enterprise Architect and Microsoft Word templates, it is possible to automatically generate customized test scripts for the functionality covered in each system use case. The test scripts take all of the testing critical design information and put that information where it belongs – in front of the tester.

Each test script provides a step by step walk-through of perceived system functionality as depicted in the information design sources. Alternate and exception flows from use cases are treated as separate test cases, allowing for all possible system paths to be exercised, and ensuring complete testing. Additional system design information such as user interface mockups, system requirements, and data elements are included in each test script at the proper steps through the use of the links that were established in the activity diagrams.



Test Case Overview: Send E-mail

Test Case Name	
Test Case Number	
Test Case Description	With the appropriate permissions, users have the ability to send an e-mail from within the CTRM System. When sent, each e-mail generates an activity which is assigned to its intended recipients.
Module/Function/Unit	
Build #	
Test Run #	
Requirement #	o REQ-03940-WS3: Send E-mail
Setup for Test (hardware, software, prerequisite tests, data, timing, security)	

Test Case Detail

Test Case 1: Send Email

Test Case	Send Email - 1
Pre-Conditions	o User is logged into CTRM.
Conditions	o Attach Files? = YES - Attach Files
Post-Conditions	o E-mail has been sent to the recipients specified by the user. o An activity has been created from the information provided in the e-mail.
Testing Role	o User

Step	Action	Expected Results	Actual Results	Pass/ Fail
A.	**Select Send Email** *User will select File - Send E-mail (F9)*			
B.		**Display the Email Pop-up** *System will display the e-mail pop-up* **REQ-03940-WS3: Send E-mail** *The system should allow a user to send an e-mail. This will create an Activity. The system should populate the 'Activity Description' with the e-mail Subject and the 'Activity Comment' with the Body of the e-mail, 'Assigned To' or 'Contact Last/Surname' and 'Contact First/Given Name' will be populated with the recipients of the e-mail. Any files that were attached to the e-mail should be added as attachments to the activity. After the e-mail is sent, the activity status should be marked as 'Completed' and all of the fields for the Activity should turn read-only.*		

Figure 5 – Example of Test Script

More than Just a Print Out

Step by step documents are nothing new to user acceptance testing. In the past, test scripts meticulously written by hand were the norm. But it is more effective to automatically generate highly accurate and easy to understand test scripts by leveraging existing system architecture and augmenting it with additional design information. Benefits that have been realized through the use of this process include:

- **Cost Savings / Schedule Improvement** – Through the reuse of system architecture products for automated test script generation it is possible to dramatically cut down on the amount of resources required to develop similar documents in the past.
- **Greater Traceability** – Success in testing depends on the ability to trace previously specified design details to the developed system. By compiling any and all pertinent design details into one tool and linking them to each other in one work product it becomes possible for the tester to have a total view of the system under test.
- **Improved Testing Accuracy** – Automatic generation of test scripts reduces the possibility of human errors. When coupled with the use of in-depth reporting features, checks and balances are put into place to ensure that all requirements and system features receive the accurate testing they require,
- **A Baseline for Future Development** – Testing is only part of a system's lifecycle. In addition to developing quality system architecture products to drive testing, this process captures the actual detailed design of the system and creates the perfect baseline to plan future development from.

As systems become increasingly complex, so does system testing. To help manage this detailed analysis new tools and techniques must be utilized which improve testing quality and allow for improvements in schedule and budget.

Leveraging existing system architecture work products to capture detailed design information and drive test script generation using Sparx System's Enterprise Architect yields an effective time/cost saving solution to user acceptance testing while simultaneously improving traceability and accuracy. ■

Notes: The figures in this paper were created using *Enterprise Architect*, a UML-based modeling tool from Sparx Systems. Doreen Evans Associates has chosen to use Enterprise Architect because it is flexible, robust and affordable. The tool's adaptability and flexibility, along with its full repository, make it an ideal tool for business analysts to use.

Doreen Evans Associates (DEA), a woman-owned professional services firm in business since 1992, provides a comprehensive, architecturally framed approach to the requirements challenge. DEA has developed a Center of Requirements Excellence which provides the infrastructure necessary to support requirements best practices. The infrastructure includes LINKProcess™, a detailed, defined requirements life cycle process; a suite of integrated tools to support the process; a set of templates, guidelines and macros to generate quality requirements deliverables; and a BA competency education program. Using this infrastructure and its seasoned consultants, DEA can help clients change a business process, build an enterprise architecture, or define and manage requirements for systems.