**Drake University**
**Computer Information Systems**
**Application Software User Acceptance Test**
**Policy**

This document describes how user acceptance test policies and procedures are applied to the application environments.

Applications Covered Under this procedure are:
DUSIS which includes: SCT Banner including Self Service, Bookshelf and Online Help, SCT Banner Xtender Solutions, SCT Workflow, Eprint

Luminis which includes: SCT Luminis Content Management System, SCT Luminis Platform System

AdAstra Room Scheduling

Blackboard

RDBMS and Applications servers such as Oracle, Oracle Application Server, MicroSoft SQLServer.

## User Acceptance Testing

User acceptance testing is performed any time there is a change required to the production environment. The change may be in the form of an application release or upgrade delivered by the software vendor, an underlying platform upgrade such as an Oracle RDBMS upgrade or a code change or enhancement delivered by internal Drake University programmers.

The User Acceptance Test Phase of the Software Development Life Cycle covers all of the activities the stakeholders perform to ensure the new software or new software releases meet their requirements and do not negatively impact business processes. Formal test plans written by the stakeholder are required for vendor releases and underlying platform upgrades but not for code changes or enhancements delivered by Drake since these are much smaller in scope. However, the stakeholders are expected to perform all testing necessary to ensure the code performs as expected whether a Drake enhancement or vendor release. Test scripts are not required at this time but may be required in the future if it is determined that testing being performed by the stakeholders is not adequate. In all cases, the user is required to sign off to the CIS director that the code has been tested and is ready for promotion into the production environment.

## The User Acceptance Test Plan

The first step in User Testing is to develop the test plan. In many cases, the test plan may be reusable from previous tests, for example a test plan from Banner Release 7 may be

reusable for Banner Release 8. Or, a Banner release test plan may be used to test an underlying platform change because the scope of testing is the same.

## *Key Components of a User Acceptance Test Plan*

**Test Items and Scope**
In order to develop a test plan, the user must first understand what needs to be tested. This can be accomplished by reviewing key documentation. In the case of a Banner or other vendor release, the vendor will distribute documentation with the release. This documentation will outline the changes the vendor has made to the software. The user should review this documentation in detail to determine what has been changed. Implementation of some changes may be required, for example new validation tables may need to be populated.

In the case of a Drake deliver modification or enhancement to a software application, the user's detailed requirements for the modification should be used.

In the case of an underlying platform change, there will not be formal documentation in the change of the system and a detailed understanding of how the software application should work is required.

Each change that is to be implemented must be tested to ensure that it works as expected.

Once an understanding of what the changes are is obtained, the next step is to look at the scope of the change and the impact to other areas. For example, if the change involves an interface between both systems then the verification that the change works must take place in both systems. If the change is to a particular module within the application that later feeds data into another module, then both modules need to be tested to verify that the change works. Additionally, when changes are system wide, all major business processes should be tested.

In all cases, the user should list all function/business processes that need to be tested during the user acceptance test.

**Items Not to Be Tested**
This is a list of items that are out of scope or that will not be tested in this plan.

**Other Impacted Business Units**
List other business units that might be impacted by the changes, how they are impacted and how they will be communicated to and how their signoff on this change will be obtained.

**Drake University**
**Computer Information Systems**
**Application Software User Acceptance Test**
**Policy**

**Other Impacted Applications**
List applications that are connected to the changing application and which could be impacted. For example, Ad Astra scheduling or the Luminis Portal are connected to DUSIS. A change in how we do course scheduling would impact AdAstra. A change in MyDUSIS could impact the portal.

**Pass Fail Criteria**
Specify the pass/fail criteria for this plan. In other words, what will need to happen to keep the entire project from being promoted to production and what will need to happen for promotion to take place on schedule.

**Risks, Assumptions and Constraints**
Identify any risks and the associated mitigation plan. Identify underlying assumption that must occur in order for testing to be completed.

**Test Resources and Schedule**
Identify the specific persons that will be responsible for performing the test and the specific dates and times they will be required to do the test.

**Test Environment**
Identify the environment in which testing will occur. At Drake it will either be the Test Environment or the Test Release Environment. In either case, identify whether a refresh of data from the production environment will be required in order to do the test.

# Communicating the Test Plan

When the test plan is complete it needs to be communicated to all impacted persons identified in the plan. These include but are not limited to supervisors of resources, owners of other modules that are impacted, owners of impacted systems, etc. All test plans should be reviewed formally within the Team Leader's group with the appropriate personnel. A formal test plan review will be scheduled when appropriate.

# Test Scripts/Scenarios

Once the test plan is built, the next step is to build individual test scenarios for each test item identified in the plan. These scenarios describe in more detail what will be tested. For example, if the test item is the Online Registration Process, individual scenarios might be Registration of a Summer Web Course Student, Registration of a Student With Holds, Registration of a Graduate Student, Registration in a Close Course, etc. For each test scenario, expected results should be documented. In the example of Online Registration Process, Registration of a Student with Holds scenario, the expected results might be: The Student receives an error message that they cannot register due to a hold. Test scenarios should be as detailed as necessary to thoroughly test the application.

Test scenarios should also identify when the expected results should occur and when they should not occur. Using the Registration of a Student with Holds scenario, persons without a hold on their accounts should be able to register as appropriate, however, students with a hold on their account should not be able to register. Both scenarios should be tested.

Test scenarios should be defined by someone who knows the business processes, application and underlying data in detail.

## Test Data

The next step in User Acceptance Testing is to identify data in the system that will be used to test each scenario. Data with the specific attributes to fit the scenario must be identified. Using the example above of Online Registration, Registration of a Student with Holds scenario, a person with hold(s) on their account must be tested. It is recommended that more than one set of test data be identified as this will improve the quality of the test.

Again, someone with knowledge of the business process, application and underlying data should be responsible for identifying the test data and the amount of data that needs to be tested.

## Performing the Test

When performing the test, the person assigned to do the testing based on the test plan should execute each scenario using the identified data. They should record whether or not the expected results were received. If the expected results were not received they should attempt to determine why. Technical assistance may be required to do so and technical assistance should be requested from either the application vendor or Drake technical resource. If the reason why the expected results were not received is a bug within the system, the tester should identify whether or not this bug is severe enough to keep the application from being promoted to the production environment. These types of bugs are often called “show stoppers”. If a work around is available, then the bug is generally not a show stopper. All show stoppers should be reported immediately to the Director of CIS so that a plan to mitigate the effects of the bug, a fix for the bug or a new production promotion schedule that will allow the bug to be fixed can be created and communicated.

For bugs that aren’t show stoppers, the tester should build a plan to communicate the work arounds and ensure everyone who is going to use the software is aware of the workaround and trained appropriately prior to the promotion to production. User training

should also be planned and executed for any new function that will be promoted to production.

## Testing Completion

When the testing is complete, the team leader should report that testing is complete and they are ready for the promotion to production . If testing is for a Drake development project, communication should occur to the programmer/analyst developing the application or application enhancement unless otherwise directed by the Director of CIS. A formal test results review will be scheduled by the Director of CIS when appropriate.