Introduction

The purpose of this process is to guide successful development, testing, and release management of ADS Web Application services.

The scope of this plan includes all ADS managed web applications co-hosted on ESYS’s shared IIS infrastructure.

Process Overview

A. Process Objectives

This process is designed to guide ADS developers and ESYS OSAs through an efficient and predictable development and release lifecycle for custom web application services. Primary emphasis has been placed on enabling quality assurance controls and introducing new resource efficiencies.

B. Benefits

- Increased staff efficiency by limiting frequency of ESYS tickets.
- Increased service uptime through weekly release window.
- Increased service quality through testing and review controls.
- Increased service security through review and scanning controls.
- Increased transparency through CAB RFC process and related campus communication.

C. When to Use this Process

This process should be adopted in the following cases:

- ADS Staff commence development of a standard (scheduled) web application release.
- ADS Staff commence development addressing a production defect (hotfix) on an existing web application release.

Process Detail

This process has been broken into two distinct activity lifecycles: Standard (Scheduled) Releases and Emergency (Hotfix) Releases. Standard (Scheduled) Releases will follow a strict timeline with regularly scheduled deployment windows. This will be the preferred method of deployment.

Emergency (Hotfix) Releases will address exceptional circumstances wherein expedited deployment is required for service delivery. This method will be executed only with proper priority and approval to address service outages or broken functionality.

A. Standard (Scheduled) Release (A1)

This activity is characterized by a unit of web application development work, identified as standard priority by the ADS development lead or ADS manager in cooperation with the application stakeholder(s). The scope of work is usually defined in a work request or IRES project charter, is approved for development by ADS management, and has a fixed-delivery date or
standard SLA for delivery. The deliverable of this activity is typically a functional web application comprised of a set of user-requested, testable functionality.

This activity consists of 4 phases: Active Development, Quality Assurance, Scheduling, and Production. See Diagram A1 for detail.

**Active Development (A1.1)**

During active development, ADS development staff document, implement and test the approved deliverables. Several iterations may occur with source code residing in a private GitHub Fork, testable code residing on the Development IIS server, and technical documentation residing in common ADS shared folders. Other staff may be consulted throughout the development/testing process. When the developer is confident a deliverable is complete and ready for Quality Assurance, he submits his code (via GitHub Pull Request) for review.

Upon submission of GitHub Pull Request, the ADS Development lead (or designee) performs a visual code-inspection and documentation review. The original developer(s) should be available for comment during this process. Emphasis will be placed on coding consistency, security best practices, proper error handling, and viable business logic. Any corrective changes will be addressed via Pull Request feedback, and subsequent code commits or documentation changes. This step may consist of several iterations.

**Quality Assurance (A1.2)**

Once a Pull Request is accepted, the ADS Lead or Manager will use the Jenkins tool to publish code from the master branch of the ADS organization GitHub repository to the Staging IIS environment. Quality Assurance will consist of 4 steps: Qualys Web Application Scanning, ADS Functional Review, ADS Web Accessibility Review, User Acceptance Testing and Sign-Off.

The primary ADS staff developer is responsible for assuring the deployed code contains no Level 4 or Level 5 security vulnerabilities as identified by Qualys WAS, and no red “Errors” in the WAVE Web Accessibility Evaluation Tool. If QA fails at this step, the activity halts and Active Development restarts to address issues.

Concurrent with Qualys and accessibility scanning, a functional review of the application will be performed by the ADS Lead (or designee) and primary ADS developer(s). Emphasis will be placed on feature completeness, authorization, regression testing, data integrity, accessibility, and user experience. If issues are identified at this stage, the activity halts and Active Development restarts to address issues.

Finally, User Acceptance Testing (UAT) is commenced via notification to end users. Communication to the end-user(s) should consist of staging URLs for testing, a comprehensive list of features or user-stories for testing, and testing tips or instructions where appropriate. If issues are identified at this stage, the activity halts and Active Development restarts to address issues. If UAT is successful, ADS will request a written “Sign-off” via email.

**Scheduling (A1.3)**

Following Quality Assurance, Scheduling will commence. The ADS Lead or Manager will issue a new Production Deployment Request for Change (RFC) or modify an existing RFC. Requests are to be scheduled on a weekly basis during a pre-determined release window. The RFC will contain a list of web applications to be updated/deployed to production with a list of major new features/fixes.

At the scheduled date/time, referencing the RFC the ADS Lead or Manager will use the Jenkins tool to publish code from the master branch of the ADS organization GitHub repository to the Production IIS environment.

**Production (A1.4)**

ADS developers, lead and/or manager will verify successful deployment. ADS will communicate to end-users. Once production deployment has been verified, the deployment lifecycle activity is closed.

**B. Emergency (Hotfix) Release (A2)**

This activity is characterized by a unit of web application development work, identified as critical priority by the ADS development lead or ADS manager in cooperation with the application stakeholder(s). Work is typically initiated via a reported defect or service outage. The primary deliverable of this activity is the restoration of service through defect remediation. This activity consists of 4 phases or states: Discovery, Patch Development, Quality Assurance, and Production. See Diagram A2 for detail.

**Discovery (A2.1)**
This phase is initiated upon the discovery of a defect or service outage in a production web application. The discovery may occur by ADS Staff, End Users, or any other entity. Once discovered, the defect should be reported as a production defect or outage via the campus ticketing system, email, or direct communication. The ADS lead or manager will consider the reported defect for immediate remediation. Consideration will be given for impact, risk, resource availability, and business process (end-user) concerns. If the defect warrants emergency priority, Patch Development will commence. If the defect can be addressed at a future time, it will enter a standard backlog and exit this activity.

**Patch Development (A2.2)**

Patch Development directly mirrors the Active Development (A1.1) process of Standard (Scheduled) Releases defined above. Notable difference is that critical priority may necessitate an accelerated time frame for development. Additional development resources may be allocated to aid expedited delivery.

**Quality Assurance (A2.3)**

Quality Assurance directly mirrors the Quality Assurance process of Standard (Scheduled) Releases defined above. Notable difference is the discretionary application of Qualys WAS scanning, Web Accessibility scanning and UAT Sign-Off. Required in Standard (Scheduled) Releases, these steps become optional at the discretion of ADS lead or manager. If immediate remediation is deemed critical, Qualys WAS scanning and/or Web Accessibility scanning may be delayed until immediately following release. In some cases UAT Sign-Off may be explicitly deferred by the user(s). In such a case, the user may simply request an immediate fix and forgo explicit sign-off.

**Production (A2.4)**

In lieu of a scheduled release, Emergency (Hotfix) deployment will be initiated via an Emergency RFC. ADS Lead or Manager will use the Jenkins tool to publish code from the master branch of the ADS organization GitHub repository to the Production IIS environment.

ADS developers, lead and/or manager will verify successful deployment. ADS will communicate to end-users. Once production deployment has been verified, the deployment lifecycle activity is closed.

**Historical Context**

ADS has a demonstrated demand for better Quality Assurance controls. Historically, any ADS developer could request deployment of a web application, without regard for quality, impact, or ESYS resource availability. While certain quality controls exist, they have not been explicitly enforced or adequately defined. The process seeks to better enforce these controls, by defining high-level process steps, and limiting developer access to deployment activities. Under this new process, ADS lead and/or manager are required to initiate staging/production deployment tasks. This places responsibility on ADS lead and/or manager to enforce process steps outlined in this document.

**Future Improvement**

This process attempts to define and implement a standard for deploying web applications with existing tools and resources. There are several constraints and limitations as a result. Most notable is a lack of automation in any of the outlined steps. It is the goal of the process owner to continually evaluate and propose process optimizations as the process matures. Some identified optimizations include:

- Code migration automation – provide manual or automatic triggers to execute migration of deliverable code through the ESYS IIS hosting environments. (Achieved via continuous integration.)

- Continuous Integration/Build Automation – provide manual or automatic triggers to build and deploy deliverable code directly from source repository (GitHub). (Achieved.)
Addendum: Information Security Policy

CSU System-Wide Information Security Policy [ICSUAM Section 8000], specifically Policy Number 8070.S000 states:

Campuses must integrate information security requirements into the software life cycle of information systems that contain protected data. The security requirements must identify controls that are needed to ensure confidentiality, integrity, and availability. These controls must be appropriate, cost-effective, and mitigate risks that may result from unauthorized access, use, disclosure, disruption, modification, or destruction of the protected data.

Several process activities outlined in this document seek to directly address requirements of ICSUAM Policy 8070.S000. Specifically:

- Process Activities A1.1 and A2.1 define a development environment, separate from production to address section 1.3 (Application Development and Production Architecture) of Policy 8070.
- Process Activities A1.1 and A2.1 outline an application review process as required in section 1.4 (Application Coding) and section 1.6.1 (Code Reviews) of Policy 8070.
- Process Activities A1.2 and A2.2 outline a functional review with a checkpoints for data integrity and authorization controls. This is intended to address section 1.5.1 (Data Security) and section 1.6 (Web and Application Testing and Change Management) of Policy 8070.
- Process Activities A1.1 and A2.1 outline documentation as specified in section 1.6 (Web and Application Testing and Change Management) of Policy 8070.
- Process Activities A1.2 and A2.2 outline Web Application Vulnerability Scanning to address section 1.6.2 (Web Application Vulnerability Scanning) of Policy 8070.
- Process Activity A1.3 outlines a change management process to address section 1.6.3 (Web and Application Change Management) of Policy 8070.
ADS Web Development: Standard Release Lifecycle
(Weekly Release: Tuesday 0700)

Active Development (A1.1)
1:N iterations per release

Quality Assurance (A1.2)
1:N iterations per release

Scheduling (A1.3)
1 iteration per release

Production (A1.4)

Diagram A1
Diagram A2

**ADS Web Development: Emergency Hotfix Lifecycle**
*(Priority Release: ADS Lead/Manager Approval Required)*

- **Discovery (A2.1)**
  - Issue Enters Standard Backlog

- **Patch Development (A2.2)**
  - 1:N iterations per hotfix
  - Writer/Test Application Code & Documentation
  - Code Deployed to IIS (DEV)
  - UAT Ready?
  - Github Pull Request Submitted

- **Quality Assurance (A2.3)**
  - 1:N iterations per hotfix
  - Review Code/Docs

- **Production (A2.4)**
  - 1 iteration per hotfix
  - Publish to STG via Jenkins
  - WAS Scheduled
  - Web Accessibility Pass

- **End User**
  - Production Defect Reported

- **ADS Lead/Manager**
  - Production Defect Reported
  - Prioritized as Hotfix
  - Changes Requested

- **ADS Developer**
  - Production Defect Reported
  - Writer/Test Application Code & Documentation
  - Code Deployed to IIS (DEV)
  - UAT Ready?
  - Github Pull Request Submitted

- **Verified**
  - Publish to PRD via Jenkins

- **Closed**
  - Published to PRD

- **Verified**
  - Functional Testing
  - UAT Sign-Off

- **Verified**
  - WAS Scheduled
  - Web Accessibility Pass

- **Verified**
  - Github Pull Request Submitted
  - Review Code/Docs

- **NO**
  - Functional Testing

- **YES**
  - WAS Scheduled

- **NO**
  - Github Pull Request Submitted

- **YES**
  - Review Code/Docs

- **REJECT**
  - Changes Requested

- **YES**
  - Changes Requested

- **ACCEPT**
  - Changes Requested

- **NO**
  - Changes Requested

- **YES**
  - Changes Requested

- **NO**
  - Changes Requested

- **YES**
  - Changes Requested