CSSC Release Management Plan

Change Order: XRN4627

Title of Programme: CSS Consequential

Authors (for this version):	Sajad Hussain
Version:	V0.4
Status:	Industry Review
Date:	16th August 2021
Programme:	CSS Consequential

This document contains confidential or privileged information; it should not be copied or disclosed to any third party without the express permission of Xoserve Ltd. All rights reserved.

1 Document Control

1.1 Version History

Version	Status	Date	Author(s)	Summary of Changes
V0.1	Draft	07/12/19	Tim Banks	Draft document
V0.2	Draft	30/04/20	Tim Banks	Updated
V0.3	Draft	19/06/20	Steve Jackson	Updated
V0.3.1	Draft	19/06/20	Jo Galloway	Updated
V0.4	Draft	16/08/201	Sajad Hussain	Updated

1.2 Reviewers

Name	Role	Organisational Unit	Reviewed
Ed Krupa	CSSC Programme Manager	CSSC Programme	Y
Peter Hopkins	Quality Assurance Manager	CSSC Programme	Y
Chan Dabare	Programme Test Manager	CSSC Programme	Y
Smitha Pichrikat	CSSC Project Manager	CSSC Programme	Y
Chami Leeniyagoda	CSSC Defect Manager	CSSC Programme	Y
Amerjeet Bharj	CSSC Test Manager	CSSC Programme	Y
Jo Galloway	Shipper Support Test Manager	CSSC Programme	Y
Shipper Testing Participants	Shipper Test Managers	Shipper Testing Participants	

1.3 Approvers

Name	Role	Organisational Unit	Outcome (1,2 or 3)
Ian Leitch	CSSC Programme Director		

Contents

2 Introduction

The purpose of the CSS Shipper Testing Release Plan Document is to define the process and assign responsibility for the release & deployment into the test environment during CSS Shipper Testing. This process will provide a clear view of what code is deployed for the system components and an audit trail for the list of defects/changes included in each code drop.

The intended audience of the Release Plan (CSSC-RP) is all industry participants through the CSS Shipper Test Support Forum.

Note: The process for releasing code/configuration changes for defects/fixes generated during Shipper Testing will be agreed with the relevant stakeholders for that test phase.

3 Background

Ofgem, as part of their commitment to make the energy market work better for consumers by improving their experience of switching leading to greater engagement in the retail market, is proposing to implement changes to the current switching regime via the Ofgem Switching Programme.

The Ofgem Switching Programme will deliver faster switching – including the capability for next-day switching and improved reliability of the switching process through better management and oversight of industry data. The key objective of the Ofgem Switching Programme is to ensure that energy consumers can easily, and with confidence, switch their energy supplier. The current switching arrangements, developed in the late 1990s, are complex for suppliers and can lead to delays, errors and costs, which are often borne by consumers. Through the Switching Programme Ofgem are taking steps to radically transform current switching arrangements to deliver faster, more reliable switching for consumers. It will introduce a new Central Switching Service solution that will serve both gas and electricity switches.

4 Scope

4.1 In Scope

This release strategy will define the process for:

- How code/configuration is released to the environment during the CSS Shipper test phase.
- How code/configuration versions are managed.
- The promotional model - (route to live).
- Frequency of regular, planned code drops.
- The process for ad hoc critical defect fixes.
- Interaction with BAU code lines.
- Content/format of release notes.

It will also cover working closely with the SI for management of releases during the external test phases, and with our customers for releases during CSS Shipper Testing.

4.2 Out of Scope

The following items are not in scope.

- Final productionisation of the code into “live”. This will be managed within the CSSC programme.

4.3 Release Manager

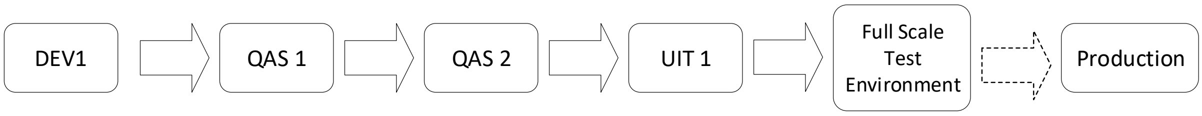
The responsibilities for the Release Manager are:

- To manage the end-to-end release process
- To review and update the Release Strategy
- To facilitate the bundling and progression of code/configuration changes into the UIT environment and maintain a list of the versions of each component being tested.
- To coordinate planning and preparation for “final” deployment into test
- To coordinate the creation of the deployment plans
- To ensure coordination between the build and test teams to ensure the smooth transition of releases into the appropriate environment
- To provide industry reports on release progress

Note: Releases to the UIT test environment requires the approval of the Release Manager, or delegate.

4.4 Promotional Model

All changes to the CSSC code will follow the promotional model as shown below.



Code/configuration can only be deployed to the UIT environment after deployment and successful internal testing in QAS 1 and QAS 2 Environments.

Code/configuration must be deployed and tested within a production like environment before deployment to production. The production deployment is out of scope for this document.

5 Release types used within CSSC Programme

This document will define the approach for the following three release types:

- Baseline Release – the initial release to an environment at the start of a test phase
- Regular Planned – These will be planned in with agreement with the programme.
- Emergency – a defect or issue requires immediate deployment to allow progression of testing

5.1 Baseline release

The first Shipper Testing will be conducted on the baseline release which will contain changes to the CSS code for modified business processes.

5.2 Regular release process

Once the UIT environment has been populated with the baseline release, the following release process will be initiated:

Task	Environment	Day														
		Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
Releases	DEV 1	🟧					🟧					🟧				
	QAS 1	🟧	🟧				🟧	🟧				🟧	🟧			
	QAS 2	🟧	🟧	🟧			🟧	🟧	🟧			🟧	🟧	🟧		
	UIT 1				🟧					🟧					🟧	
Release note (Provisional)			🟧					🟧					🟧			
Release note (Final)					🟧					🟧					🟧	

Notes:

1. Whilst this schedule shows a release each Thursday, a release activity will only be required if there are defects which have been fixes and tested ready for deployment. Please note, Defects will not automatically be fixed and deployed in the week they are raised.
2. Defects would go to the development team for analysis/fix when they are raised
3. Defects that are identified to be released on Thursday would be tested internally by at least Wednesday of that week.
4. A provisional release note would be created by Tuesday to be shared with Shipper Testing participants.
5. Defects that pass internal testing would be collated into a release to be deployed to UIT Environment on Thursdays, and the contents of the release note confirmed.

A release note containing all changes since the last release will be created, and all release notes will be available on the CSS Shipper Support webpage. (Link to be provided)

5.3 Emergency code drops process

If a defect is detected that stops the CSS Shipper test phase, then an emergency code drop will be considered. This process is the exception rather than the rule.

For critical defects, (P1), a release review will be arranged to:

1. Agree if the fix requires an immediate deployment.

2. Agree a schedule.
3. Approve the emergency fix.

A release note will be created to cover the emergency release, and then with agreement with stakeholders, a fix will be deployed.

Any change deployed relating to an emergency/hotfix will be incorporated into the next regular release note.

5.4 Deployment to the UIT environment

Code can only be promoted to the UIT environment once it has been validated internally and a release will be created detailing all changes since the last release, and a release schedule agreed.

Any fix that impacts the CSS components within the UIT environment, will need to be deployed as per the Release Window provided by the SI or by the emergency release window which has been agreed with the SI.

5.5 Deployment Issues/Failures

If a release fails with errors during deployment or during smoke testing, then a ‘fix forward’ approach will be taken. Once the errors are corrected, then the release will be re-deployed to the environment.

Once smoke testing is successfully executed, the release will be considered complete.

6 RACI

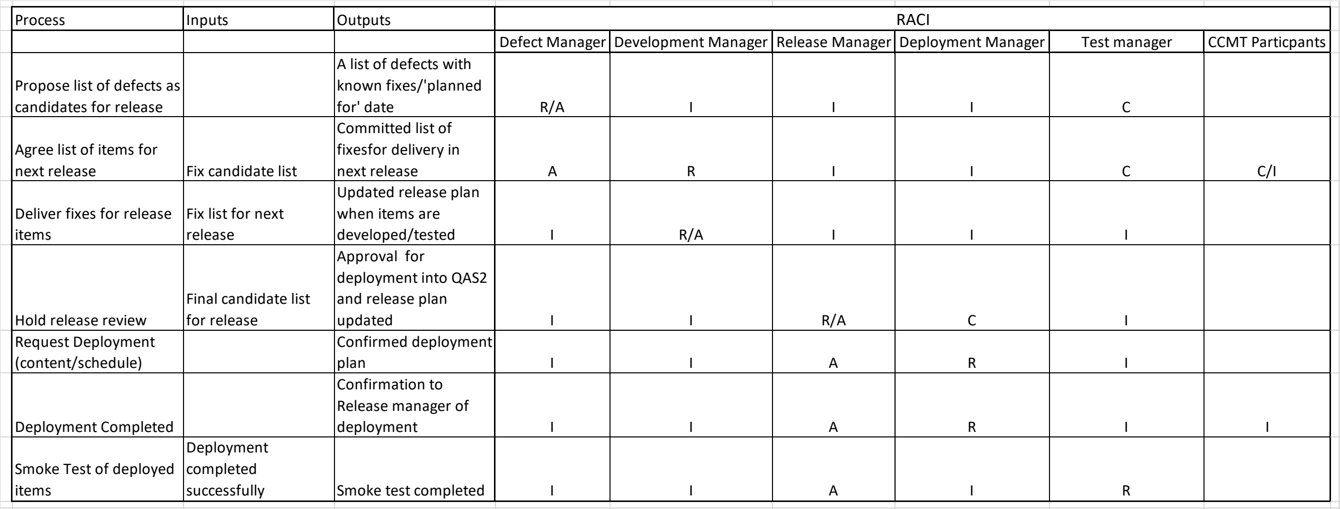
The following RACI Matrix outlines the activities and ownership of these activities.

R - Responsible,

A - Accountable,

C - Consulted,

I - Informed.



Process	Inputs	Outputs	RACI					
			Defect Manager	Development Manager	Release Manager	Deployment Manager	Test manager	CCMT Particpants
Propose list of defects as candidates for release		A list of defects with known fixes/'planned for' date	R/A	I	I	I	C	
Agree list of items for next release	Fix candidate list	Committed list of fixesfor delivery in next release	A	R	I	I	C	C/I
Deliver fixes for release items	Fix list for next release	Updated release plan when items are developed/tested	I	R/A	I	I	I	
Hold release review	Final candidate list for release	Approval for deployment into QAS2 and release plan updated	I	I	R/A	C	I	
Request Deployment (content/schedule)		Confirmed deployment plan	I	I	A	R	I	
Deployment Completed		Confirmation to Release manager of deployment	I	I	A	R	I	I
Smoke Test of deployed items	Deployment completed successfully	Smoke test completed	I	I	A	I	R	