**Enhanced Service Order Product Team**

**- Agile Product Development process & guidelines-**

**Product Backlog:** Business leads and SMEs enters their User Stories to ESO Jira space and goes under "Product Backlog". The user story authors are responsible to introduce their user stories to the Business Leads and Business SMEs and the group will review the user stories for acceptance and prioritization. During this process, the ESO's Product Owner, Business Leads, and other Business SME's (if needed) shall make sure the User Stories are clearly explained with clear expected results and have correctly prioritized. The final authority and ownership of any user stories that gone through review, prioritization, ranked for a future Sprint is the Product Owner.

<https://jira.ameren.com/projects/ESO/summary>

**Product Backlog Grooming:** During product backlog grooming, the Product Owner or any of the Business leads introduces the User Stories to the IT/Tech/development/QA Automation team. The Business leads and Business SMEs should answer all the questions IT/Development team have and provide all the clarification Development team needed. The expected results should be agreed up on by the whole ESO Product team and the Product team shall make sure there is no "Blocks" to the do-ability of the user story to its given priority by the Product Owner.

Each User story will be allocated a "Story Point" by the IT/Development team

**Sprint Planning:** During sprint planning, the ESO Product team will choose all the User Stories doable for the next sprint based on the priority and do-ability. The story point per Sprint is known as velocity of the agile team. Each user story must be assigned to an application group. Development team members can pick up the user story based on their band width at a given time.

**Active Sprint - IT User Story review / Sub Task creation & Estimation:** After Sprint Planning, IT/Development team should review each user story in the sprint in detail with IT and Development centric and create necessary sub tasks for each user story. Each sub tasks must me estimated in hours or days in Jira.

**High Level Design:** The Tech Lead shall develop the High level design for the feature for the deployment. It shall be presented and reviewed by the ESO Product team, and must be approved.

**Technical Design:** Each application team (Web, API, and CSS) shall create their own technical Design based on the High Level Design and shall be reviewed by ESO Development Team, and approved.

**Detail Design/Detail Technical Design:** Each application team (Web, API, and CSS) shall create their own Detail technical Design or Detail Design based on the High Level Design and shall be reviewed by all applications/development teams, and approved.

**Development and Unit Testing:** Each Developer shall complete the development of the User Story assigned to him/her and conduct successful unit testing

**Code Peer Review:** Any code changes, made by anybody in the ESO Product team must be Peer reviewed and approved (documented) prior to System Testing. The owner of the code change must provide sufficient details to the Reviewer to ensure the code review met the QA standards and was efficient. The recommended and agreed changes must be done, documented, confirmed, and approved.

**Integration/System Test:** The developers from all impacted application teams must make sure their changes are integrated end to end and the sub-feature (user story) successfully works end to end. The changes and/or new features are working and meeting expected results.

****The Integration/system test shall be conducted in the colored flow (test environment) above.

**UAT Testing on CSS Build/CSSPX4:** Business Lead shall conduct thorough UAT testing on PLZ01- CSS Build/CSSPX4 with UAT test Scripts in TestRail. Please refer to the following environments bellow



UAT test shall be conducted in the colored flow (test environment) above.

**Regression testing: on QA - CSS PTST/PTST-Customer:** Business Lead shall conduct a regression testing on QA - CSS PTST/PTST-Customer prior to move UAT testing on STAG environment. Please refer to the following environments bellow

The regression testing will be automated for ESO product team

Web

QA

API

QA

CSS Code

PTST

CSS Test Data

PTST-Customer

**UAT testing on STAG – Web QA - CSS STAG/PTST-Customer:** Business Lead shall conduct a regression testing on QA - CSS PTST/PTST-Customer prior to move UAT testing on STAG environment. Please refer to the following environments bellow

Web

QA

API

QA

CSS Code

STAG

CSS Test Data

STAG

**GO / No Go:** A Go / No Go call will be made day before the deployment/release date. A Go or No Go decision will be made by the team based on the UAT test results and the readiness of the feature to go to production. There shouldn't be any unresolved severity 1 or critical defect and successful regression testing.

**General ESO Agile Guidelines:**

**The Sprint:**

ESO Sprint, a time-box of two weeks during which a “Done”, useable, and potentially releasable product Increment is created. ESO Sprints start on Wednesdays and ends on Tuesdays. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

• No changes are made that would endanger the Sprint Goal;

• Quality goals do not decrease; and,

• Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

**Sprint Goal:**

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

**Daily Scrum:**

The Daily Scrum is a 15-minute time-boxed event for the Development Team. The Daily Scrum is held every day of the Sprint. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity.

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint.

The structure of the meeting is set by the Development Team and can be conducted in different ways if it focuses on progress toward the Sprint Goal. Some Development Teams will use questions, some will be more discussion based. Here is an example of what might be used:

• What did I do yesterday that helped the Development Team meet the Sprint Goal?

• What will I do today to help the Development Team meet the Sprint Goal?

• Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or re-plan, the rest of the Sprint’s work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master teaches the Development Team to keep the Daily Scrum within the 15-minute time-box.

The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting. Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team’s level of knowledge. This is a key inspect and adapt meeting

**Sprint Review:**

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.

This is at most a four-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendees understand its purpose. The Scrum Master teaches everyone involved to keep it within the timebox.

The Sprint Review includes the following elements:

• Attendees include the Scrum Team and key stakeholders invited by the Product Owner;

• The Product Owner explains what Product Backlog items have been “Done” and what has not been “Done”;

• The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;

• The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;

• The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed);

• The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;

• Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,

• Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

**Sprint Retrospective:**

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose.

The Scrum Master ensures that the meeting is positive and productive. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

• Inspect how the last Sprint went with regards to people, relationships, process, and tools;

• Identify and order the major items that went well and potential improvements; and,

• Create a plan for implementing improvements to the way the Scrum Team does its work.

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of “Done”, if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation