Software-Development Life Cycle

JUNE 2024

Table of Contents

Overview

Edgio's Software-Development Life Cycle (SDLC) formalizes software-development processes to improve efficiency and reduce implementation time.

The SDLC focuses mainly on the developmental aspects of the life cycle and serves as a reference for specific information on the development process as a whole and on the detailed process steps for each phase. Edgio's software development cycles through five phases: Conception, Design, Validation, Implementation, and Support. This document defines the process flow and detailed information for each process activity.

Scope

This document details updates, improvements, and fixes to current Production products and systems.

Document Information

This document is organized in sequential phases of Edgio's software-development life cycle. The tasks in the flowcharts are numbered and correspond with the items of the same number in the phase's *Detailed Activities* section. Roles/ responsibilities in the *Detailed Activities* section and in the ARCI matrices are marked as A (Accountable), R (Responsible), C (Consulted), or I (Informed).

References: PDF of diagrams | Version history

Roles and Responsibilities

This section describes the main team/role responsibilities during the software-development life cycle. The *Detailed Activities* sections and ARCI matrices of this document also describe tasks and responsibilities.

Change Management

The Change-Management team reviews all updates to Production systems and products. Change Requests are submitted and tracked using ServiceNow. These tickets are reviewed weekly in the Change-Management meeting with all stakeholders, where changes are discussed, questions are answered, and the Change Manager verifies that all Change Requests are scheduled during an appropriate change window for that product or system; the internal and client impact is detailed; and the change has pre- and post-maintenance verifications and test, rollout, and rollback plans. The Change Manager also decides if the change's impact requires Edgio to notify clients and creates and sends notifications to meet Service-Level Agreements (SLAs) LLNW has established with clients. Change Management policies, procedures, workflows, and timelines are documented in the Change Management Confluence space.

Change Review Board (CRB)

The CRB may be added as approvers to changes that involve Sarbanes Oxley (SOX) compliance, EdgeQuery billing changes, and Emergency Change Requests. The Senior Vice President of Operations must approve changes to Change Review Board membership and the list of valid approvers. Currently, LLNW's CRB comprises members from content-delivery operations, client support, network/ infrastructure, information technology, and billing/ SOX.

Database

The Database team reviews release requests that touch or impact Edgio databases to verify that the change will not cause any unexpected impact to LLNW databases. If Database engineers have questions or concerns about a release, they confer with the appropriate resources assigned to the release until they are comfortable signing off on the release. They note their approval in the Release Request (RR) ticket, and their approval is required for the release to be complete and move forward.

Development

The Development team collaborates with the Product-Management team to scope and prioritize updates to Production products and systems. According to priority, the Development team adds items to a scheduled release and creates all required release information and supporting tickets, including internal and client release notes, QA testing templates, additional scoping requests, client documentation updates, etc. The Development team coordinates with the QA team throughout the release

cycle to ensure appropriate testing and allows for QA (and user) testing in a sandbox environment. Responsible for all release-associated tasks, the Development team verifies that each release has all the required approvals.

DevOps/ Engineering

The Development Operations (DevOps) role or Engineering resource reviews each release request. Any questions or concerns about a release are worked out with the appropriate resources assigned to the release. DevOps/ Engineering approval is captured in the Release Request (RR) ticket, and their approval is required for the release to be complete and move forward.

Emergency Manager

When a Change Request cannot follow the processes for a Standard Change Request and is submitted as an Emergency Change Request, the manager of the Change Requester must review the ticket. After reviewing the request, scheduled start and end dates, justification, and impact, the Emergency Manager either approves or rejects the request via ServiceNow. An Emergency Change Request requires approval from the Change Requester's manager before it can be implemented.

Feature Requester

Any Edgio resource can request a new product feature. These resources usually interface with clients and report new feature requests to the Product-Management team by submitting a feature request. Features may also be requested by the Product team. Before requesting a new product feature, the Feature Requester researches the history and potential of the new feature before submitting a JIRA ticket to justify the business value and benefits of implementing the feature. Feature Requesters also assess the type of feature so they can submit the appropriate ticket type: accelerated (AFR) or internal (IFR). These requests are then reviewed with the Product-Management team so the team can decide to accept or reject the request or add it to the Product roadmap or backlog.

Incident Management

The Incident-Management team is responsible for managing all Incident Alerts (IAs). When an IA is submitted, the Incident-Management team collaborates with Technology Resources via Slack and a phone bridge until the issue is resolved. This team also keeps the ServiceNow ticket updated and sends out client notifications per Service-Level Agreements (SLAs) and at major milestones throughout issue resolution. If a Root Cause Analysis (RCA) is requested, the Incident-Management team completes the analysis and sends the results to clients and stakeholders.

Incident Requester

An Incident (IA) can be reported by any resource via the ServiceNow ticketing portal. When an IA ticket has been submitted, ServiceNow automatically sends the Incident Requester an email with the IA number.

Network Operations Center (NOC)

The NOC team actively monitors systems and network functions 24/7 for any unexpected behavior and also responds to system-generated alerts and alarms. NOC engineers also perform server, circuit, and network troubleshooting and system repair. This team tracks issues and their progress via ServiceNow EVENT tickets and either resolves issues using runbooks or other expertise or invokes the Incident-Management process if the issue needs to be escalated.

Platform Engineering

The Platform-Engineering team reviews release requests that touch or impact the operation of Edgio systems to verify that the change will not cause any unexpected impact. If engineers have questions or concerns about a release, they confer with the appropriate resources assigned to the release until they are comfortable signing off on the release. This team also usually peer reviews the Change Request for the release. When this team signs off on the release, they note their approval in the Release Request (RR) ticket on behalf of *Operations*, and their approval is required for the release to be complete and move forward.

Operations Emergency Group

When a Change Request cannot follow the processes for a Standard Change Request and is submitted as an Emergency Change Request, the Operations Emergency Group reviews the ticket. After reviewing the request, scheduled start and end dates, justification, and impact, this group either approves or rejects the request via ServiceNow. An Emergency Change Request requires approval from this group before it can be implemented.

Peer Reviewer

All submitted Change Requests require peer review to verify that scheduled start and end dates are scheduled during an appropriate window; client and internal impact is clearly defined; and the pre- and post-maintenance verifications and test, rollout, and rollback plans are complete. The Peer Reviewer also determines whether the Change Request can be submitted as a Standard Change or if the change cannot follow the standard process and must be submitted as an Emergency Change. For both change types, Change Requests must be peer reviewed before the tickets can move forward in the process for further review. Peer Reviewer approval is tracked via ServiceNow.

Performance Quality Assurance (Perf QA)

Perf QA is conducted by the QA team for each release that has the potential to impact the performance of any system, service, or product to verify that any release updates test within the margin of error for that configuration item. All releases that may impact performance require Perf QA review and approval before the release can advance.

Problem Management

Problem Management reviews and triages all Incident Alerts (IAs). Any possibility of the re-occurrence of an Incident (IA); multiple, related Incidents (IAs); or multiple clients experiencing the same issue warrant review by Problem Management, and a Problem ticket (PRB) in ServiceNow. This team creates new tickets, consolidates identical tickets, and creates and monitors progress of Problem tasks for other teams to resolve the issue. If a Problem cannot be resolved, this team explains the issue to stakeholders, executives, and management teams to accept the risk.

Product Management

The Product-Management team processes and prioritizes new projects, products, features, fixes, and other updates. They collaborate with Development and other teams and stakeholders to prioritize and scope items for a timely release based on business value, effort, resources, and other considerations.

Quality Assurance (QA)

This team collaborates with the Product-Management and Development teams in the Design Phase to create a test design, test cases, automation, and test hosts for requested updates to Production systems. This team also has an important role in the Validation Phase. They conduct final regression testing, review the release notes, and confirm that all *Critical*, *Showstopper*, and *Major* release items are fixed and that their associated tickets are closed. Quality Assurance signoff is required on the JIRA release ticket before a release can progress.

Sales

The Product-Management team consults with the Sales team when deciding the priority and business justification for feature requests. The Sales team gives insight into client interest, revenue, and value of implementing certain features before others.

Security

The Security team reviews release requests that touch or impact Edgio security to verify that the change will not cause any unexpected impact and that it complies with all Sarbanes Oxley (SOX) and Service Organization Control (SOC) requirements. If the Security team has any questions or concerns about a release, they confer with the appropriate resources assigned to the release until they are comfortable signing off on the release. They note their approval in the Release Request (RR) ticket, and their approval is required for the release to be complete and move forward.

Service-Reliability Engineering (SRE)

The SRE team supports the Development team with some tasks during the release process. Specifically, they often create, populate, and submit the Change Request for Development releases. They also represent the release during the Change-Management process and serve as the Subject-Matter Experts for Change-Request approvals.

Stakeholders

Stakeholders are representatives from various teams across the organization to advise on or participate in various aspects of the software-delivery life cycle.

Subject-Matter Expert (SME)

As part of the Change-Management process, SMEs are required to approve each Change Request. The Change-Management team assigns SMEs to each Change Request, and SMEs can add other approval resources at their discretion. SMEs may include engineers, management, developers, and anyone specialized in the product or service being changed. These approvals and review responses are tracked on Change-Request tickets in ServiceNow, and approval of each listed resource is required for a change to progress.

Support

This team comprises three groups of support, and each ticket is first investigated by Client Support, then escalated as required through Client Operations, and then Platform Engineering. The Client Support team fields emails and phone calls from Edgio clients 24/7; they either resolve the client request, assign the ticket to a more appropriate team, request an Incident Alert, or submit a JIRA ticket if a fix requires code or system updates. Each client request is tracked in ServiceNow, where updates, progress, resolution, and other details are recorded. ServiceNow also sends clients emails about their tickets based on ticket updates.

Technology Resources

Technology Resources represent various teams and roles throughout the organization. They are the technological subject-matter experts for a specific product or system and are responsible for resolving Incident Alerts (IAs) when they are submitted. This team is committed to troubleshooting client, network, and system issues as soon as possible and continue troubleshooting until resolution.

User-Acceptance Testers

When a release is deployed to a sandbox/ test environment, this gives various stakeholders and resources the opportunity to assess the functionality of new features and fixes and their impact on products and systems. The variety of reviewers is advantageous because it provides a wide array of insight from various viewpoints and testing strategies.

1. Conception Phase

The goal of the Conception Phase is to qualify product and system updates for business value and overall feasibility and answers these types of questions:

- What specific problem will the change solve?
- In what way will the change solve the problem?
- Is the change worth doing?
- What benefits will be achieved from implementing the change?

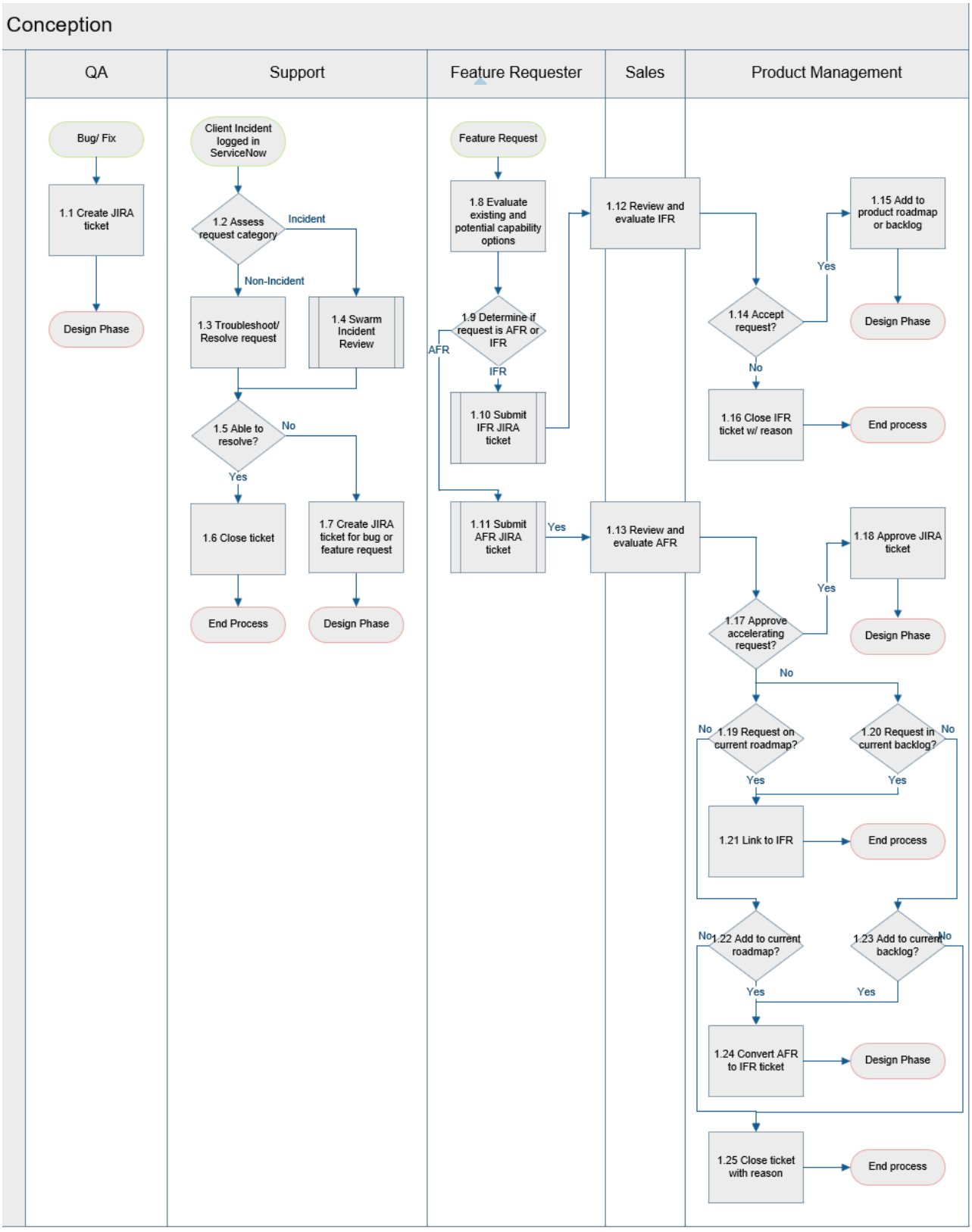
Updates to Production systems are generally submitted by LLNW teams that work with the code or the client. Bugs, fixes, and client issues are generally evaluated in the Design phase, while new feature requests are assessed for business value to be candidates for implementation.

Primary Objectives

- Establish a vision and clear objectives and initial scope for the change
- Validate the business case for the concept/ idea
- Complete an initial assessment of business value with a relative delivery priority

SOFTWARE-DEVELOPMENT LIFE CYCLE

Flowchart



Detailed Activities

ARCI matrix

1.1. Create JIRA ticket

Description: The Quality Assurance team identifies and documents improvements to the code or system in a JIRA ticket assigned to the Development team.

Input: Failed QA test, observation of need for system or code improvement or fix

Output: JIRA ticket describing bug/ fix/ improvement submitted to Development team

Roles/Responsibilities: Quality Assurance (R), Development (I)

1.2. Assess request category

Description: The client calls or emails the Client Support team with an issue, request, or inquiry, creating an Incident (INC) ticket in ServiceNow (manually for calls; automatically for email requests). The Support team assesses the type of request: Incident or non-Incident, which includes general support needs or questions, such as account-related issues, configuration issues, product purchase requests, and non break-fix issues.

Input: Client Incident ticket logged in ServiceNow

Output: Incident ticket that the Support team has investigated and classified

Roles/Responsibilities: Support (R)

1.3. Troubleshoot/ Resolve request

Description: The Support team attempts to resolve the issue. Tickets generally begin with Client Support and are escalated to Client Operations and then to Platform Engineering until a resolution is achieved.

Input: Client Incident logged in ServiceNow

Output: Incident that the Support team has investigated and attempted to resolve

Roles/Responsibilities: Support (R)

1.4. Swarm Incident review

Description: Edgio's swarming model connects the resources required to collaborate and resolve high-priority incidents efficiently, utilizing the many talents across the Edgio organization to join forces to resolve an issue. Client Support, Client Operations, and Client Success troubleshoot in real-time as a team and consult in any other internal resources needed to resolve an issue.

Input: INC client incident ticket

Output: Determination if the issue is for the Support or Client Success team to action

Roles/Responsibilities: Support (R), Technology Resources (C), Client Success (C)

1.5. Able to resolve?

The Support team swarms/ troubleshoots a client Incident (INC) and attempts to resolve the issue. Client Support, Client Operations, and Client Success troubleshoot in real-time as a team and consult in any other internal resources needed to resolve an issue.

Input: Incident that the Support team has investigated and attempted to resolve

Output: Resolved or unresolved incident

Roles/Responsibilities: Support (R), Technology Resources (C), Client Success (C)

1.6. Close ticket

Description: When the Support engineers resolve an Incident, they close the INC ticket, detailing the fix. If the Support resources create an associated Incident Alert (IA) ticket, they close the INC ticket when the IA has been closed. If Platform Engineering creates a JIRA ticket, the Support team closes the INC ticket at the engineer's discretion based on implementation dates and client business impact. Closing the INC ticket also alerts the client to the date and details of the resolution.

Input: Incident resolved by the Support team/ other resources

Output: Updated and closed JIRA ticket; client email of incident resolution

Roles/Responsibilities: Support (R)

1.7. Create JIRA ticket for bug or feature request

Description: For unresolvable issues, Platform Engineering creates a JIRA ticket detailing the Incident and assigns to the appropriate team.

Input: Unresolved Incident that the Support team has investigated and is unable to resolve

Output: JIRA ticket with details of the issue assigned to the appropriate team for resolution

Roles/Responsibilities: Support (R), Development (I)

1.8. Evaluate existing and potential capability options

Description: Before submitting a feature request ticket, the Feature Requester researches Confluence to see if this request has already been made and evaluates the existing and potential options for the feature. The Feature Requester investigates what has already been done: what worked and what did not work. The requester also consults with Technology Resources to ensure that the feature request contains as much information as possible.

Input: Need for feature update

Output: Deeper knowledge of the history and potential for the feature request

Roles/Responsibilities: Feature Requester (R), Technology Resources (C)

1.9. Determine if request is AFR or IFR

Description: The Feature Requester assesses the request to determine the correct type of ticket to submit. An Accelerated Feature Request (AFR) is a formal request for a client feature with revenue and business justification and business value; an Internal Feature Request (IFR) is a way to request development of a feature or bug fix not tied to revenue.

Input: Desire or need for a new feature

Output: Appropriate JIRA ticket type identified

Roles/Responsibilities: Feature Requester (R)

References: AFR process | IFR process

1.10. Submit IFR JIRA ticket

Description: Complete and submit a JIRA ticket for an IFR, including all the existing and potential capabilities previously investigated.

Input: Determination that the request is an IFR

Output: Submitted IFR JIRA ticket

Roles/Responsibilities: Feature Requester (R), Product Management (I)

Reference: Refer to the *Creating a New IFR* section of the IFR Confluence page for details on ticket fields and requirements.

1.11. Submit AFR JIRA ticket

Description: Complete and submit a JIRA ticket for an AFR, including all the existing and potential capabilities previously investigated.
Input: Determination that the request is an AFR

Output: Submitted AFR JIRA ticket

Roles/Responsibilities: Feature Requester (R), Product Management (I)

Reference: Refer to the *Creating a New AFR* section of the AFR Confluence page for details on ticket fields and requirements

1.12. Review and evaluate IFR

Description: The Product-Management team assesses the request by validating the business value, the level of effort that the request will take to deliver, the impact to other timelines, and the fit within the Product roadmap and Edgio's company goals. This validation process includes talking to the Feature Requester, Development resources, and other teams within the company to understand the full impact of choosing whether or not to approve an IFR that has been submitted.

Input: Submitted IFR JIRA ticket

Output: Due diligence research of IFR's suitability for Product roadmap and organizational goals

Roles/Responsibilities: Product Management (R), Feature Requester (R), Sales (A, C)

1.13. Review and evaluate AFR

Description: The Product-Management team assesses the request by validating the business value, the level of effort that the request will take to deliver, the impact to other timelines, and the fit within the Product roadmap and Edgio's company goals. This validation process includes talking to the Feature Requester, Development resources, and other teams within the company to understand the full impact of choosing whether or not to approve an AFR that has been submitted.

Input: Submitted AFR JIRA ticket

Output: Due diligence research of AFR's suitability for Product roadmap and organizational goals

Roles/Responsibilities: Product Management (R), Feature Requester (R), Sales (A, C)

1.14. Accept request?

Description: Decide if the IFR is aligned with Product Management and LLNW goals, is cost-feasible, and contributes sufficient value.

Input: Completed IFR review

Output: Determination to accept the request or not

Roles/Responsibilities: Product Management (R)

1.15. Add to Product roadmap or backlog

Description: If the Product-Management team accepts the IFR, they decide to add the request either to the product roadmap or backlog of product requests.

Input: Accepted IFR

Output: IFR added to product roadmap or backlog

Roles/Responsibilities: Product Management (R), Feature Requester (I)

1.16. Close IFR ticket with reason

Description: If the Product-Management team decides not to action the IFR request, they close the JIRA ticket with an explanation for their decision.

Input: IFR that Product Management has decided not to accelerate or add to roadmap or backlog

Output: Closed IFR ticket with explanation why it was not accelerated or added to roadmap or backlog

Roles/Responsibilities: Product Management (R), Feature Requester (I)

1.17. Approve accelerating the request?

Description: Decide if the request is aligned with Product Management and LLNW goals, is cost-feasible, and contributes sufficient value.

Input: Completed AFR review

Output: Determination to accelerate the request or not

Roles/Responsibilities: Product Management (R)

1.18. Approve JIRA ticket

Description: Update the JIRA ticket to reflect the decision to accelerate the request.

Input: Product Management-approved AFR

Output: JIRA ticket with the state in a *PM-Approved* state

Roles/Responsibilities: Product Management (R), Feature Requester (I)

1.19. Request on current roadmap?

Description: Product Management reviews the current product roadmap to see if the AFR already exists.

Input: AFR that Product Manager has decided not to accelerate

Output: Determination if AFR is on the current product roadmap

Roles/Responsibilities: Product Management (R)

Reference: Product roadmap

1.20. Request in current backlog?

Description: Product Management reviews the current product backlog to see if the AFR already exists.

Input: AFR that Product Manager has decided not to accelerate

Output: Determination if AFR is on the current product backlog

Roles/Responsibilities: Product Management (R)

1.21. Link to IFR

Description: The Product-Management team updates the AFR JIRA ticket to associate it with an existing IFR.

Input: AFR that Product Management has decided not to accelerate because it already exists on the current roadmap or backlog.

Output: AFR linked to existing IFR already on the roadmap

Roles/Responsibilities: Product Management (R)

1.22. Add to current roadmap?

Description: Product Management assesses the request to determine if it brings value, just not on an accelerated track.

Input: AFR that Product Management has decided not to accelerate that is not on the current product roadmap

Output: Determination to add the current request to the Product roadmap

Roles/Responsibilities: Product Management (R)

Reference: Product roadmap

1.23. Add to current backlog?

Description: Product Management assesses the request to determine if it brings value, just not on an accelerated track.

Input: AFR that Product Management has decided not to accelerate that is not in the current product backlog

Output: Determination to add current request to the product backlog

Roles/Responsibilities: Product Management (R)

1.24. Convert AFR to IFR ticket

Description: Product Management clones the AFR ticket, moves this ticket to the IFR project, and links the tickets to each other.

Input: AFR that Product Management has decided not to accelerate but wants to add to the IFR project

Output: IFR ticket cloned from the original AFR ticket that is now part of the IFR project

Roles/Responsibilities: Product Management (R)

1.25. Close ticket with reason

Description: The Product-Management team closes/ cancels the AFR request if they do not want to add it to the roadmap or backlog.

Input: AFR that Product Management has decided not to accelerate or add to roadmap or backlog

Output: Closed AFR ticket with explanation why the feature was not accelerated or added to roadmap or backlog

Roles/Responsibilities: Product Management (R), Feature Requester (I)

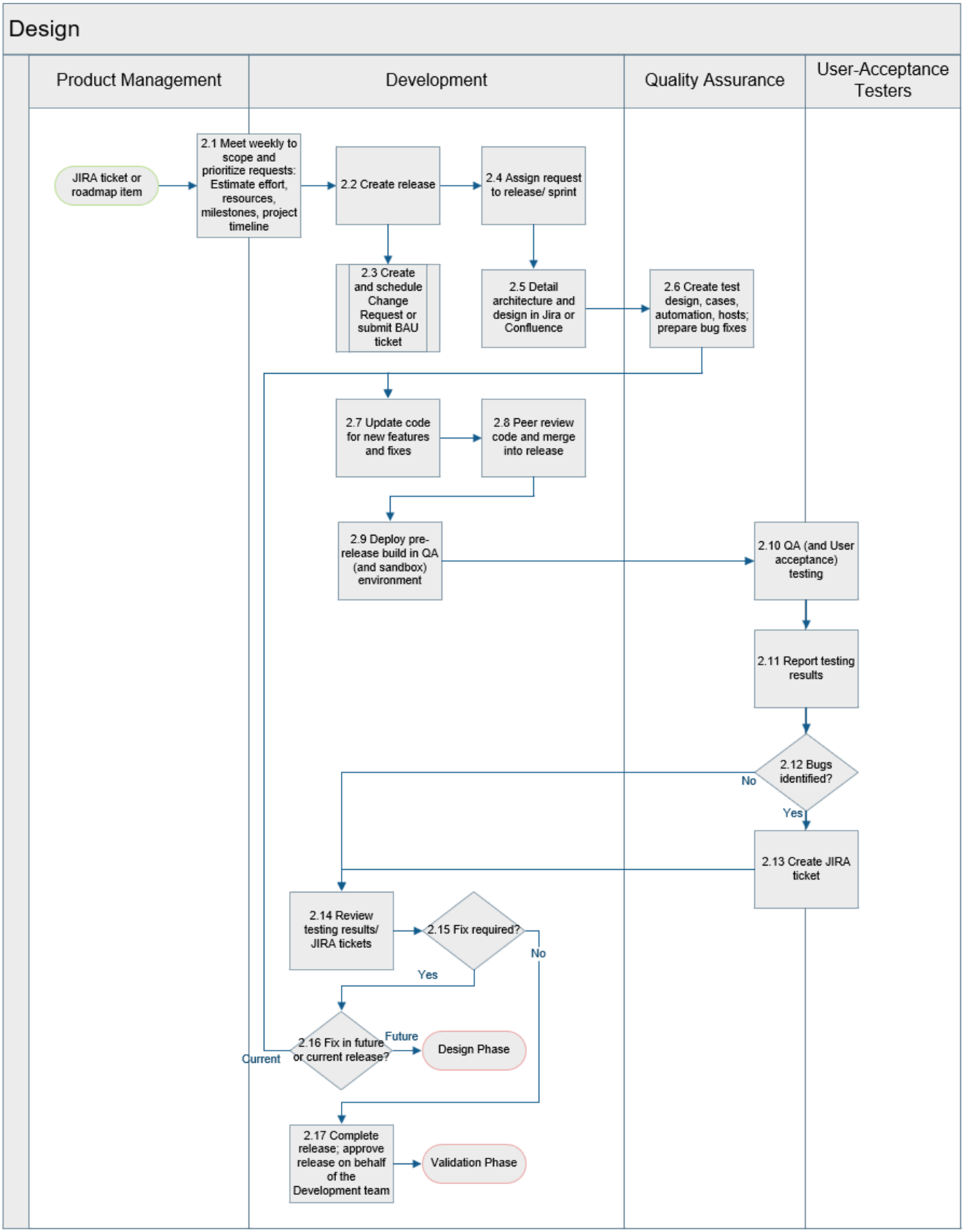
2. Design Phase

The goal of the Design Phase is to achieve concurrence among all stakeholders on the life-cycle objectives of the project: how it will be controlled, how activity status will be communicated, what the requirements are, what risks exist, and how risks will be mitigated. The Definition Phase is likely to be longer for new development efforts. Requests are reviewed for business value before being scoped and prioritized. Since the tasks in this phase are performed by various teams, protocols and cadences vary, so this section represents a baseline of the process across the organization.

Primary Objectives

- Validate the project's scope and boundary conditions, including operational vision and product inclusions and exclusions
- Identify the critical use cases of the system, the primary scenarios of operation
- Estimate potential risks, effort, resources, milestones, project timelines, etc.
- Prepare the supporting environment for the project
- Identify tasks required from other teams and submit tickets
- Schedule Change Request or BAU request

Flowchart



Detailed Activities

ARCI Matrix

2.1. Meet weekly to scope and prioritize requests: Estimate effort, resources, milestones, project timeline

Description: Many of the Product-Management and Development teams meet weekly to collaborate on the scope, priorities, estimated effort, resource allocation, project milestones, and the project timeline. If scoping is too complex for a particular item, additional scoping requests are made via a JIRA ticket and submitted to the appropriate team. The contents of a release varies by the team, the product, the deployment methodology, and other factors.

Input: JIRA tickets for bug fixes and client issues; items from Product Management's roadmap

Output: Scoped, prioritized requests, project resources, milestones, and timelines; requests for additional scoping

Roles/Responsibilities: Product Management (R), Development (R)

2.2. Create release

Description: The Development team creates the release, which varies by team protocol and cadence, but often includes tasks like creating the release branch and the release JIRA (RR) ticket; templates for internal and external release notes, support guides, and QA results; and tickets required for the release. These tickets may be for the Development team or other teams.

Input: Scheduled release date (lead time varies by team and release and change windows)

Output: Release created in JIRA, templates for artifacts required for the release, release branch created

Roles/Responsibilities: Development (R)

2.3. Create and schedule Change Request or submit BAU ticket

Description: The Development or Service-Reliability Engineering (SRE) team creates and submits a Change Request or a Business as Usual (BAU) ticket in ServiceNow during the appropriate time frame for the product/ service and with the appropriate lead time to adhere to Change-Management timelines.

Input: Scheduled release

Output: Submitted Change Request or BAU ticket in ServiceNow

Roles/Responsibilities: Development (R), Service-Reliability Engineering (R)

2.4. Assign request to release/ sprint

Description: The Development team adds existing stories, bug fixes, and feature requests as determined by collaborating with the Product-Management team to the current release or sprint.

Input: Prioritized and scoped JIRA tickets for bug fixes, client issues, and items from Product Management's roadmap

Output: Bug fixes, client issues, and items from Product Management's roadmap assigned to release/ sprint

Roles/Responsibilities: Development (R)

2.5. Detail architecture and design in JIRA or Confluence

Description: The Development team describes the architecture of the release. For simple architecture explanations, the Development team adds the information to the Release Request (RR) ticket; for architecture diagrams or more complex architectural descriptions, the Development team publishes in Confluence.

Input: Release, scoped update requests

Output: Architecture details or diagrams posted in either the JIRA ticket or in Confluence

Roles/Responsibilities: Development (R)

References: Core Specification Process

2.6. Create test design, test cases, automation, test hosts; validate bug fixes

Description: The Quality Assurance team prepares to test the release by drafting testing artifacts, automating testing when possible, creating test hosts, and testing of bug fixes.

Input: Release and associated architecture information

Output: Preparation for release testing

Roles/Responsibilities: Quality Assurance (R)

2.7. Update code for new features and fixes

Description: The Development team updates the user interface, code, or other system elements to accommodate the new features and fixes.

Input: Features and fixes identified for current release

Output: Updated code

Roles/Responsibilities: Development (R)

2.8. Peer review code and merge into release

Description: The Development team peer reviews the updated code to verify that the release meets the team's established standards and guidelines. After peer review is completed, the Development team merges the code into the appropriate release, either the current or a future release.

Input: Updated code

Output: Peer-reviewed code merged into the appropriate release ready to deploy to test environment

Roles/Responsibilities: Development (R)

2.9. Deploy pre-release build to QA (and sandbox) environment

Description: The Development team installs the new code into the test environment(s) for QA (and user) evaluation. Not all releases are deployed to a sandbox environment, but all are deployed to a QA environment.

Input: Updated code merged into the release

Output: Code deployed to test environment(s)

Roles/Responsibilities: Development (R)

2.10. QA (and user-acceptance) testing

Description: The QA team tests the current release in the QA environment, and various other teams participate in user-acceptance testing; user testing resources vary depending on the product and the items being released. Updates involving the client interface and/ or experience may warrant Product Management review. All releases are tested by QA; not all releases are user tested.

Input: Code for new release deployed to test environment(s)

Output: Release has been reviewed and tested by QA (and other stakeholders)

Roles/Responsibilities: Quality Assurance (R), User-Acceptance Testers (R)

2.11. Report testing results

Description: Each team reports their testing feedback. QA reports testing results from the test plan and publishes the test results in Confluence and Testrail. User-acceptance testers may add comments to the release ticket. (Bugs are reported via a JIRA ticket.)

Input: User-acceptance testing

Output: User-acceptance testing results submitted for Development review

Roles/Responsibilities: Quality Assurance (R), User-Acceptance Testers (R), Development (I)

2.12. Bugs identified?

Description: QA and User-Acceptance Testers look for issues or improvements during testing. Any bugs found are reported.

Input: Code deployed in test environment(s)

Output: Issue or improvement found during testing

Roles/Responsibilities: Quality Assurance (R), User Acceptance Testers (R)

2.13. Create JIRA ticket

Description: If an issue or need for improvement is discovered during testing, the Quality Assurance team or the User-Acceptance Testers submit a JIRA ticket describing the fix and assign it to the Development team.

Input: Bug identified during testing

Output: JIRA ticket describing the issue or improvement found during testing

Roles/Responsibilities: Quality Assurance (R), User Acceptance Testers (R), Development (I)

2.14. Review testing results/ JIRA tickets

Description: The Development team reviews the testing results and any JIRA tickets from QA and user-acceptance testing.

Input: Testing results and reported bugs

Output: Reviewed testing results and bugs

Roles/Responsibilities: Development (R)

2.15. Fix required?

Description: The Development team reviews the feedback from user-acceptance testing and determines if the reported issue is expected behavior or requires a fix.

Input: Testing results and reported bugs

Output: Determination if the issue reported during testing requires a fix

Roles/Responsibilities: Development (R)

2.16. Fix in future or current release?

Description: The Development team reviews the feedback from testing and determines if the reported issue that requires a fix should be remedied in the current release or in a future release. This decision could be based on the (client) impact or the severity or age of the bug.

Input: Reported bug that requires a fix

Output: Determination if the bug should be fixed in the current or a future release

Roles/Responsibilities: Development (R)

2.17. Complete release; approve release on behalf of the Development team

Description: The Development team fills out all the release build artifacts, updates the Operations Guide if required, and confirms all release documentation is completed before giving final approval on the release ticket. Teams are notified of the approval via Slack or a JIRA notification.

Input: Release build that has been fully tested

Output: Development approval on Release Request (RR)

Roles/Responsibilities: Development (R), Quality Assurance (I), Development Operations/ Engineering (I), SRE (I)

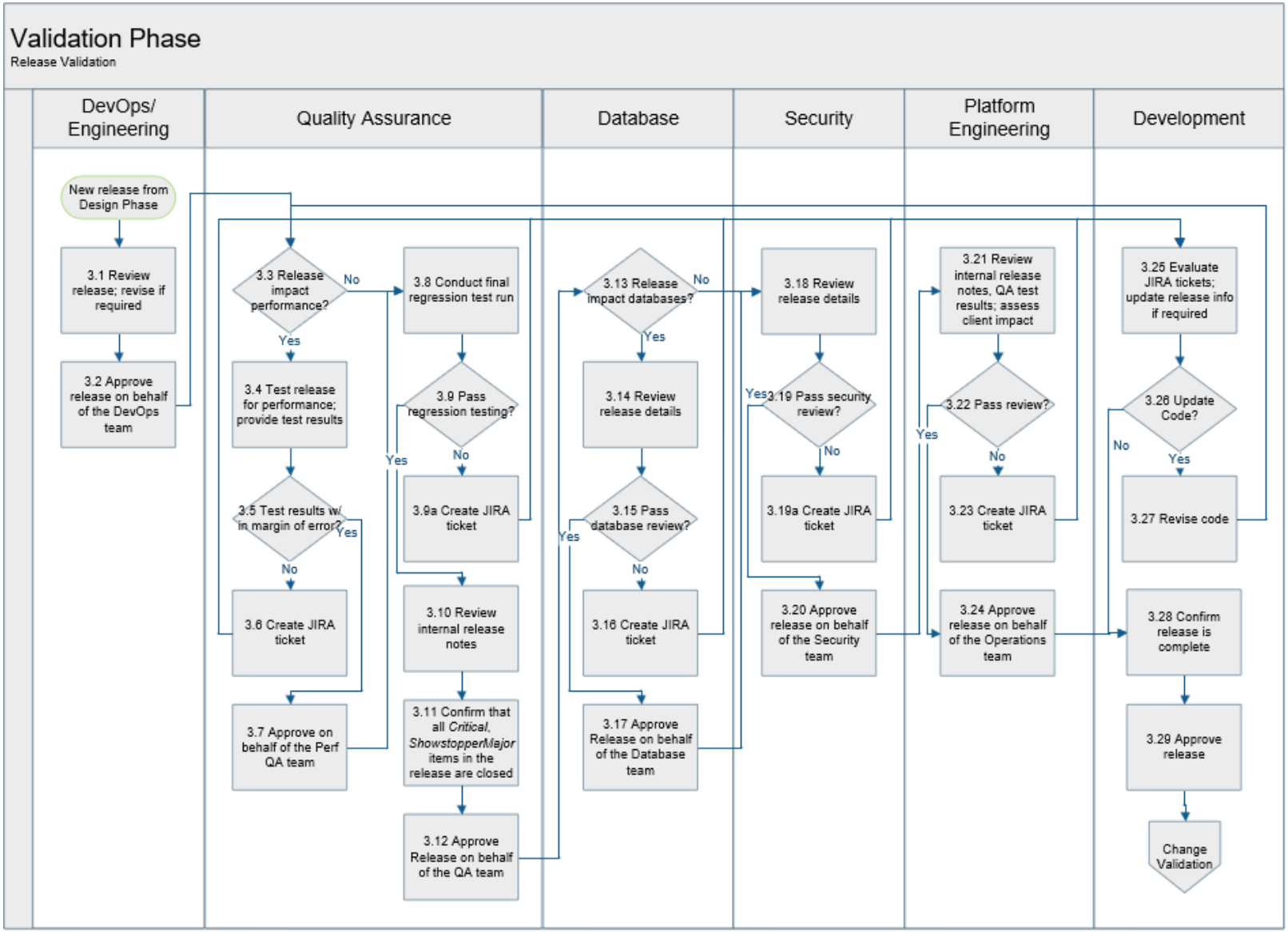
3. Validation Phase

The Validation Phase verifies updates to the Production environment in two sub-phases: release validation and change validation. Release validation checks that the code functions as expected and does not impact the environment or other systems, while change validation represents the Change Management process, which invokes a broader group of SMEs associated with the configuration items being updated. The Validation Phase ends with an approved Release Request (RR) and an approved Change Request (CHG).

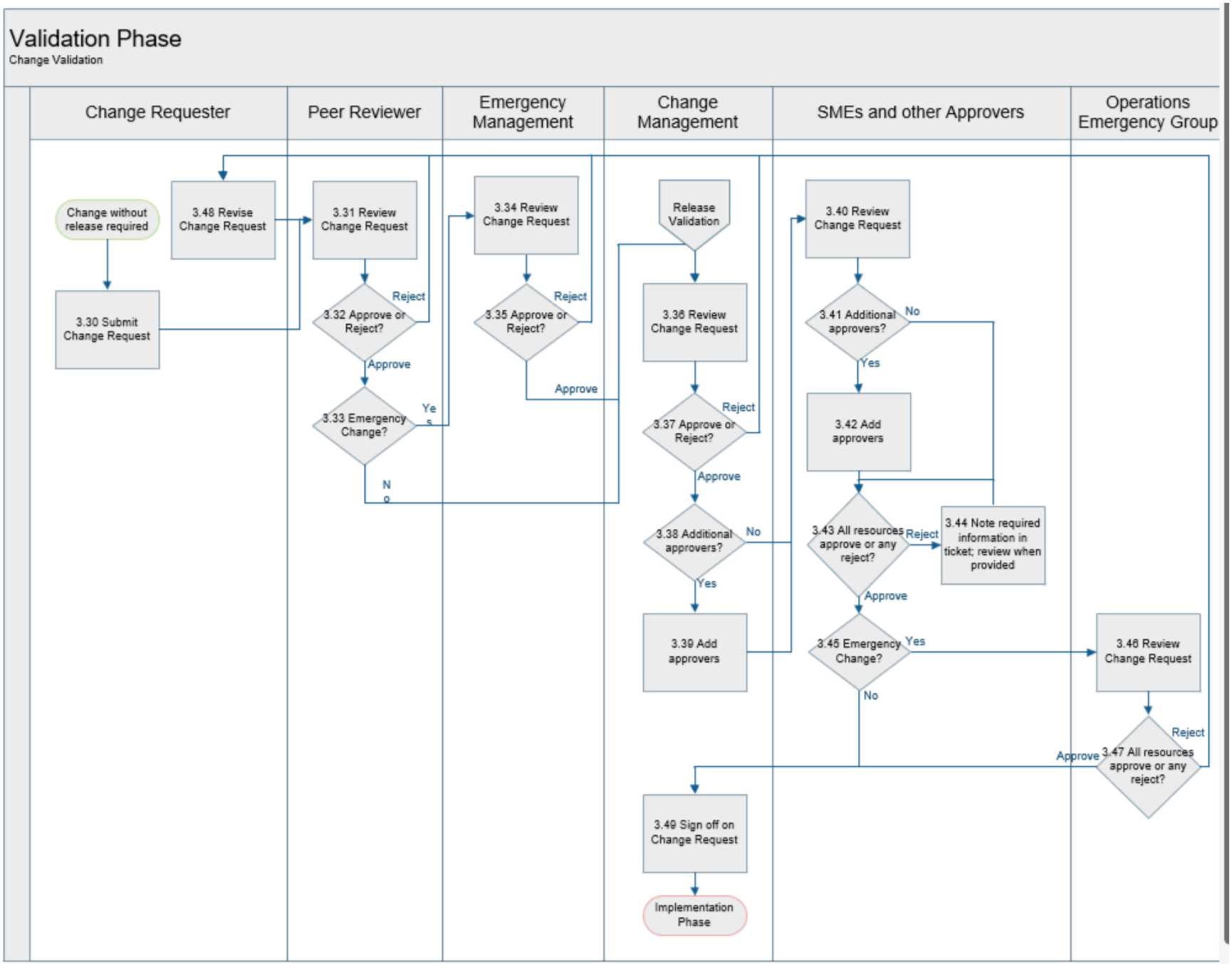
Primary Objectives

- Review, test, approve release
- Schedule Release and Change Request
- Review and approve Change Request

Flowcharts



SOFTWARE-DEVELOPMENT LIFE CYCLE



Detailed Activities for Release Validation

ARCI Matrix

3.1. Review release; revise if required

Description: After the Development team has approved the Release, Development Operations (DevOps) or Engineering reviews the release, including the release notes, updated code, scope, test cases, architecture, etc. These teams make or request updates if required.

Input: Release

Output: Release reviewed by DevOps/ Engineering

Roles/Responsibilities: DevOps/ Engineering (R)

3.2. Approve release on behalf of the DevOps team

Description: After reviewing the release and finding all criteria acceptable, the Development Operations (DevOps) or Engineering team approves the release. DevOps/ Engineering are the only resources who can sign off on behalf of DevOps, and this approval is required before the release can move forward.

Input: Release reviewed by DevOps/ Engineering

Output: Release approved by DevOps/ Engineering

Roles/Responsibilities: DevOps/ Engineering (R)

3.3. Release impact performance?

Description: If the release impacts performance, it requires Performance Quality Assurance review. If review is required, the QA team tests the release; if not, the Development team signs off for Performance QA on the release ticket.

Input: Release requiring review

Output: Determination if Performance QA testing is required for the release

Roles/Responsibilities: Development (R)

3.4. Test release for performance; provide test results

Description: For releases that may affect performance, the Quality Assurance team tests the release to confirm the changes do not affect system performance. They record their testing results in the release ticket.

Input: Release requiring performance testing

Output: Release performance tested

Roles/Responsibilities: Quality Assurance (R), Development (I)

3.5. Test results within margin of error?

Description: Performance test results must be within the margin of error (found in Confluence) before the Performance QA team can approve the release. If test results are not acceptable the QA team creates a JIRA ticket for the Development team; if test results are acceptable, QA approves the release.

Input: Performance-tested release

Output: Determination if release passes performance quality assurance

Roles/Responsibilities: Quality Assurance (R)

3.6. Create a JIRA ticket

Description: If the test results for performance testing are outside the acceptable margin of error, the QA team creates a JIRA ticket for the Development team and does not sign off on the release until it passes performance testing.

Input: Release performance test results outside the acceptable margin of error

Output: JIRA ticket for Development team

Roles/Responsibilities: Quality Assurance (R), Development (I)

3.7. Approve on behalf of the Performance QA team

Description: When QA has completed testing with acceptable results and are ready to approve the release, their approval is recorded in the JIRA Release Request (RR) ticket. If the release affects performance, this team's approval is required for the release to progress.

Input: Release performance tested with acceptable results

Output: Performance QA approval for release on RR ticket

Roles/Responsibilities: Quality Assurance (R)

3.8. Conduct final regression test run

Description: The QA team usually regression tests releases in the Design Phase in the sandbox environment. They run the regression tests again before signing off on the release.

Input: Release requiring QA review

Output: Release that has been regression tested

Roles/Responsibilities: Quality Assurance (R)

3.9. Pass regression testing?

Description: After regression testing, the QA team determines if the release passes before signing off on the release. If the release does not pass QA regression testing, QA logs a JIRA ticket for the Development team.

Input: Release that has been regression tested

Output: Determination if the release passes QA testing

Roles/Responsibilities: Quality Assurance (R)

3.9a. Create a JIRA ticket

Description: If the release does not pass regression, the QA team creates a JIRA ticket for the Development team and does not sign off until the release passes regression testing.

Input: Release that does not pass regression testing

Output: JIRA ticket for Development team

Roles/Responsibilities: Quality Assurance (R), Development (I)

3.10. Review internal release notes

Description: QA reviews the internal release notes to verify that the description accurately reflects the features and fixes contained in the release, updating if necessary.

Input: Release requiring QA approval

Output: Internal release notes reviewed by QA

Roles/Responsibilities: Quality Assurance (R)

3.11. Confirm all *Critical*, *Showstopper*, and *Major* tickets in the release are closed

Description: The Quality Assurance team verifies that the important tickets associated with the release are closed, meaning they have been resolved.

Input: Release requiring QA approval

Output: All *Critical*, *Showstopper*, and *Major* tickets in the release are verified as closed

Roles/Responsibilities: Quality Assurance (R)

3.12. Approve release on behalf of the QA team

Description: After reviewing the release and finding all criteria acceptable, the Quality Assurance team approves the release. Only QA resources can sign off on the release on behalf of QA, and this approval is required before the release can move forward.

Input: Release reviewed by QA

Output: Release approved by QA

Roles/Responsibilities: Quality Assurance (R)

3.13. Release impact databases?

Description: If the release impacts any of Edgio's databases, it requires Database team review. If review is required, the Database team tests the release; if not, the Development team signs off for the Database team on the release ticket.

Input: Release requiring review

Output: Determination if the release affects any databases

Roles/Responsibilities: Development (R)

3.14. Review release details

Description: The Database team evaluates the release and its impact on any databases.

Input: Release requiring database review

Output: Release reviewed by Database team

Roles/Responsibilities: Database (R)

3.15. Pass database review?

Description: The Database team reviews the release and its impact on databases to determine if the release's impact is acceptable. If the impact is acceptable, the Database team can sign off on the release; if not, the Database team creates a ticket for the Development team so the issue can be resolved before the release moves forward.

Input: Release reviewed by Database team

Output: Determination if release does not impact any databases

Roles/Responsibilities: Database (R)

3.16. Create a JIRA ticket

Description: The Database team creates a JIRA ticket for the Development team if they discover that the release impacts any database in an unacceptable way.

Input: Release that adversely impacts any database

Output: JIRA ticket for Development team

Roles/Responsibilities: Database (R), Development (I)

3.17. Approve release on behalf of the Database team

Description: After reviewing the release and finding all criteria acceptable with no impact to Edgio's databases, the Database team approves the release. Only the Database team can sign off on the release for database review, and this approval is required before the release can move forward.
Input: Release reviewed by the Database team

Output: Release approved by the Database team

Roles/Responsibilities: Database (R)

3.18. Review release details

Description: The Security team evaluates the release and any security-related impact while confirming that the release adheres to internal established security standards, as well as Sarbanes Oxley (SOX) and System and Organization Controls (SOC) compliance.

Input: Release requiring security review

Output: Release reviewed by Security team

Roles/Responsibilities: Security (R)

3.19. Pass security review?

Description: The Security team reviews the release and its impact on established security standards to determine if the release's impact is acceptable. If the impact is acceptable, the Security team can sign off on the release; if not, the Security team creates a JIRA ticket. Only Security team resources can sign off on the release for security review, and their approval is required before the release can move forward.

Input: Release reviewed by the Security team

Output: Determination if release impacts any established security standards

Roles/Responsibilities: Security (R)

3.19a. Create JIRA ticket

Description: The Security team creates a JIRA ticket for the Development team if they discover that the release compromises LLNW's security standards in an unacceptable way.

Input: Release that does not conform to established security standards

Output: JIRA ticket for Development team

Roles/Responsibilities: Security (R), Development (I)

3.20. Approve release on behalf of the Security team

Description: After reviewing the release and finding all criteria acceptable with no impact to Edgio's security or security compliance, the Security team approves the release. Members of the Security team are the only resources who can sign off on the release on behalf of security, and their approval is required before the release can move forward.

Input: Release reviewed by the Security team

Output: Release approved by the Security team

Roles/Responsibilities: Security (R)

3.21. Review internal release notes, QA test results; assess client impact

Description: Platform Engineering reviews the details of the release and the Operations Guide to understand the changes made in the release. If the release changes functionality or user interfaces for clients, the engineer verifies they have the training and documentation necessary to address any issues.

Input: Release requiring Operations review

Output: Release reviewed by the Operations resource

Roles/Responsibilities: Platform Engineering (R)

3.22. Pass review?

Description: Platform Engineers review the release and its impact to determine if the release's impact is acceptable. They refer to the release notes, the Operations Guide, and QA test results to confirm they have the information they need to understand the changes made in the release. When the team resources are ready to sign off on the release, they mark their approval on the release ticket on behalf of Operations; if they need more information or find an issue, they create a ticket for the Development team.

Input: Release reviewed by the Platform Engineering team

Output: Determination if release passes Operations review

Roles/Responsibilities: Platform Engineering (R)

3.23. Create a JIRA ticket

Description: The Platform-Engineering team creates a JIRA ticket for the Development team if they discover any issues with the release.

Input: Release that does not pass Operations review

Output: JIRA ticket for Development team

Roles/Responsibilities: Platform Engineering (R), Development (I)

3.24. Approve release on behalf of Operations

Description: After reviewing the release and finding all content to be acceptable, the Platform-Engineering team approves the release. Members of the Platform-Engineering team are the only resources who can sign off on the release on behalf of Operations, and their approval is required before the release can move forward.

Input: Release reviewed by Platform Engineering

Output: Release approved by Platform Engineering

Roles/Responsibilities: Platform Engineering (R)

3.25. Evaluate JIRA tickets; update release information if required

Description: The Development team reviews the JIRA tickets submitted during release review and testing to identify those that need to be actioned before the release can move forward. For tickets that request clarification or updates to release information, the Development team revises the release details on the release ticket.

Input: JIRA tickets from review and testing

Output: Revised release information and identification of items requiring code updates

Roles/Responsibilities: Development (R)

3.26. Update code?

Description: The Development team reviews the JIRA tickets to identify if any tickets require code updates that need to be fixed before the release can proceed. Required code changes that need to be fixed in the current release require approvals from all stakeholders again on the revised release; updates to release details do not require further approval.

Input: JIRA tickets from review and testing

Output: Determination if feedback from review and testing requires a code update

Roles/Responsibilities: Development (R)

3.27. Revise code

Description: After reviewing the JIRA tickets submitted from release review and testing, the Development team updates the code for issues that need to be fixed before the release can proceed.

Input: JIRA tickets from review and testing that require code updates in the current release

Output: Updated code

Roles/Responsibilities: Development (R)

3.28. Confirm release is complete

Description: The Development team confirms that the release has all the required approvals, all artifacts are completed, and the code is ready for Production deployment.

Input: Release ready for approval

Output: Release confirmed ready for approval

Roles/Responsibilities: Development (R)

3.29. Approve release

Description: The Development team approves the release ticket. This comprehensive approval is required before the release can proceed to the next step in the process: Change validation. Only the Development team can provide this approval.

Input: Release confirmed ready for approval

Output: Approved release

Roles/Responsibilities: Development (R)

Detailed Activities for Change Validation

ARCI Matrix

3.30. Submit Change Request

Description: All changes to the Production environment must have a Change Request (CHG) in ServiceNow. Change Requests must be submitted with enough lead time to allow for appropriate handling. Information required on a Change Request includes service interruption expected; areas of impact (e.g. POP, region, etc.); product and service elements involved in the change; desired time and date of change, according to the predefined change windows; desired change window duration; completed risk and impact assessments; Method of Procedure (MOP); monitoring considerations, including server and network equipment squelch procedure; backout procedure; and pre- and post-validation test plans. Changes associated with a release will have already been submitted in the Design Phase and will already have certain approvals (DevOps, Performance QA, and QA) as part of the release process.

Input: Release or other change to the Production environment

Output: Change Request submitted in ServiceNow

Roles/Responsibilities: Change Requester (R)

References: Change Request Submission policy

3.31. Review Change Request

Description: By signing off on a Change Request in ServiceNow, all Peer Reviewers agree to the integrity of the information and methods of change implementation, specifically the time and date of deployment, the code and configuration, and the maintenance plan.

Input: Submitted Change Request in ServiceNow

Output: Peer reviewed Change Request

Roles/Responsibilities: Peer Reviewer (R)

References: Peer Review Sign-off policy

3.32. Approve or reject?

Description: Before signing off on a Change Request, Peer Reviewers review and attest to the integrity of the information and methods of change implementation, specifically the time and date of deployment, the code and configuration, and the maintenance plan. If a Peer Reviewer rejects the Change Request, it goes back to the Change Requester for updates.

Input: Peer reviewed Change Request

Output: Peer Reviewer approval or rejection

Roles/Responsibilities: Peer Reviewer (R), Change Requester (I)

3.33. Emergency Change?

Description: Any Change Request that cannot comply with any of the Change Management policies may be considered an Emergency Change. Exceptions that cannot follow the SLAs for client notifications are considered Emergency Changes. A clear explanation of the potential impact and justification for why this Change Request cannot be performed in a predefined window must be provided.

Input: Submitted and peer reviewed Change Request in ServiceNow

Output: Assessment whether to submit the Change Request as a Standard Change or as an Emergency Change

Roles/Responsibilities: Peer Reviewer (R), Change Requester (I)

References: Emergency Change Requests policy

3.34. Review Change Request

Description: Emergency Change Requests are subject to an accelerated review process. The Change Requester's immediate manager, director, and vice president review Emergency Changes to understand the potential impact and justification for why the Change Request cannot be executed through standard change processes.

Input: Peer reviewed Emergency Change Request

Output: Emergency Management review of Emergency Change Request

Roles/Responsibilities: Emergency Management (R)

3.35. Approve or reject?

Description: The Change Requester's immediate manager, director, and vice president review Emergency Changes and approve or reject based on potential impact and justification for why this Change Request cannot be implemented according to the standard change process.

Input: Peer reviewed Emergency Change Request

Output: Emergency Management approval or rejection

Roles/Responsibilities: Emergency Management (R), Peer Reviewer (I), Change Requester (I)

3.36. Review Change Request

Description: The Change-Management team confirms that the change is requested during the proper change window and verifies the internal and client impact of the change. For Emergency Changes the Change-Management team assesses the potential impact and justification for why this Change Request cannot be performed per the standard change processes. Change Management confirms that these items are clearly defined or contain links to detailed information: pre-maintenance verification, test plan, rollout plan, post-maintenance verification, rollback plan. For changes associated with a release, the Change-Management team confirms that the Release Request (RR) has all the required approvals.

Input: Peer reviewed Standard or Emergency Change Request

Output: Change Management review of Change Request

Roles/Responsibilities: Change Management (R)

References: Change Windows

3.37. Approve or reject?

Description: The Change-Management team decides if all the criteria for a change are met and either approves the Change Request or rejects the Change Request so that information can be corrected or added.

Input: Change Management reviewed Change Request

Output: Change Management approval or rejection

Roles/Responsibilities: Change Management (R), Peer Reviewer (I), Change Requester (I)

3.38. Additional approvers?

Description: The Change-Management team assigns SMEs and approvers according to probable impact and scope of the change. The Change-Management team may request Change Review Board approval and/ or any additional approvals for any change at their discretion. For high-risk changes that are submitted on time but need to be scheduled outside of regular maintenance windows, director and vice-president approvals may be required at Change Management's discretion.

Input: Change Request approved by Change Management

Output: Determination if the Change Request requires additional approvers

Roles/Responsibilities: Change Management (R)

References: Change Request Approvals policy

3.39. Add approvers

Description: The Change Management team adds any supplemental approvers to the Change Request. These approvals are required for the Change Request to advance.

Input: Change Management determination that approvers are required on a Change Request

Output: New approvers assigned to the Change Request

Roles/Responsibilities: Change Management (R), SMEs and other Approvers (I)

3.40. Review Change Request

Description: The SMEs and other Approvers review the content of the Change Request, including the scheduled start and end dates, client and internal impact, pre-maintenance verification, test plan, rollout plan, post-maintenance verification, rollback plan, and the list of approvers.

Input: Change Request requiring SME and other Approver review

Output: Change Request reviewed by SME and other Approvers

Roles/Responsibilities: SMEs and other Approvers (R)

3.41. Additional approvers?

Description: The SMEs and other Approvers determine if the list of approvers on the Change Request represent all facets and touchpoints of the change.

Input: Change Request reviewed by SME and other Approvers

Output: Determination if the Change Request requires additional approvers

Roles/Responsibilities: SMEs and other Approvers (R)

3.42. Add approvers

Description: The SMEs and other Approvers add any supplemental approvers to the Change Request. These approvals are required for the Change Request to advance.

Input: SMEs and other Approvers assessment of existing and required approvers

Output: New approvers assigned to the Change Request

Roles/Responsibilities: SMEs and other Approvers (R, I)

3.43. All resources approve or any reject?

Description: All assigned SMEs and other Approvers must approve or reject the Change Request. All SMEs and other Approvers must approve the Change Request before it can advance. If any resources reject the Change Request at this point in the process, they note the clarification or additional information required, and the approval process continues.

Input: Change Request requiring approval from SMEs and other Approvers

Output: Approved or rejected Change Request

Roles/Responsibilities: SMEs and other Approvers (R, I), Change Management (I), Peer Reviewer (I), Change Requester (I)

3.44. Note required information in ticket; review when provided

Description: If any resources reject the Change Request at this point in the process, they note the clarification or additional information required, and the approval process continues. Subsequent reviewers are informed of the rejection and noted information.

Input: Rejected Change Request

Output: Comments added to Change Request regarding corrections or additional information required

Roles/Responsibilities: SMEs and other Approvers (R) (I), Change Management (I), Peer Reviewer (I), Change Requester (I)

3.45. Emergency Change?

Description: Emergency Changes require approval from the Ops Emergency Group, which consists of executive-level SMEs across multiple teams, including Network and Operations. Standard Changes do not require additional approvals at this juncture.

Input: Emergency or Standard Change Request approved by all SMEs and other Approvers

Output: Assessment whether the Change Request follows the Standard Change or Emergency Change workflow

Roles/Responsibilities: Change Management (R)

3.46. Review Change Request

Description: Emergency Changes require review from the Ops Emergency Group, which consists of executive-level SMEs across multiple teams, including Network and Operations.

Input: Emergency Change Request requiring Ops Emergency Group review

Output: Emergency Change Request reviewed by Ops Emergency Group

Roles/Responsibilities: Ops Emergency Group (R)

3.47. All resources approve or any reject?

Description: All assigned members of the Operations Emergency Group must approve or reject the Change Request. All Ops Emergency Group resources must approve the Change Request before it can advance. If any resources reject the Change Request at this point in the process, they note the clarification or additional information required, and the change workflow is reset to the beginning of the process.

Input: Emergency Change Request requiring approval

Output: Approved or rejected Emergency Change Request

Roles/Responsibilities: Operations Emergency Group (R), SMEs and other Approvers (I), Change Management (I), Peer Reviewer (I), Change Requester (I)

3.48. Revise Change Request

Description: If the Change Request is rejected prior to the SMEs and other Approvers review, the Change Requester provides additional information or revises existing content of the change in the Change Request ticket. Change Requests must be revised and resubmitted with enough lead time to allow for appropriate handling per established Change-Management policies.

Input: Rejected Change Request or request for correction or updates to the ticket's contents.

Output: Updated Change Request in ServiceNow

Roles/Responsibilities: Change Requester (R)

References: Change Request Submission policy

3.49. Sign off on Change Request

Description: The Change-Management team must provide final signoff on the Change Request after reviewing all content, approvals, and notifications are completed. When Change Management signs off, the Change Request is moved to a *Ready* state awaiting implementation on the scheduled start date.

Input: Change Request with all required approvals

Output: Change Request completed and in *Ready* state awaiting implementation

Roles/Responsibilities: Change Management (R), SMEs and other Approvers (I), Peer Reviewer (I), Change Requester (I)

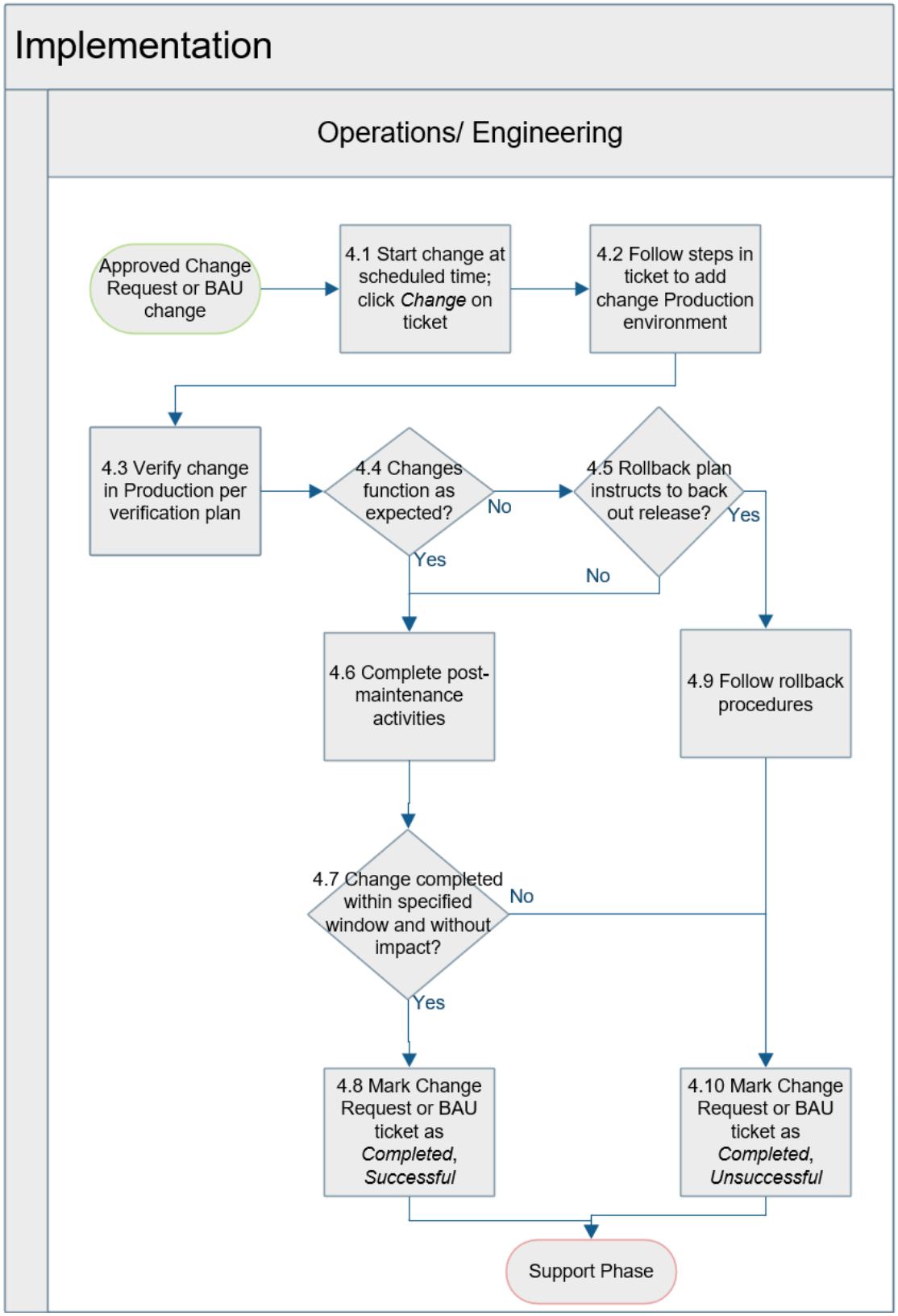
4. Implementation Phase

The goal of the Implementation Phase is to deploy updates to the Production environment per the procedures detailed in the Change Request: Scheduled start and end dates and times, pre- and post-maintenance verification, validation plan, backout plan. If a change causes unexpected impact or goes beyond the specified maintenance window, the change is considered unsuccessful.

Primary Objectives

- Deploy changes to the Production environment per Change Management policies, procedures, and timelines
- Validate the changes in Production, backing the changes out per the prescribed rollback plan, if necessary
- Record implementation details in the Change Request

Flowchart



Detailed Activities

ARCI Matrix

4.1. Start change at scheduled time; click *Start Change* on ticket

Description: The Operations or Engineering team deploys the change at the beginning of the start date designated in the Change Request and clicks *Start Change*, which marks the Change Request as *In Progress*. This status is updated in the associated Slack Channel, and all resources who have opted in for status updates on this change are notified by email. The Operations/ Engineering resource begins with the pre-maintenance activities then follows the steps in the change plan from the Change Request.

Input: Approved Change Request or BAU change ticket

Output: Change deployment started; Change Request in *In Progress* state

Roles/Responsibilities: Operations/ Engineering (R), Change Requester (I), Peer Reviewer (I), SMEs and other Approvers (I)

4.2. Follow steps listed in ticket to add change to Production environment

Description: Each Change Request must include a detailed change plan. The Operations/ Engineering team follows the pre-verification and change-plan steps listed in or linked from the Change Request.

Input: Change implementation in progress

Output: Change installed in Production

Roles/Responsibilities: Operations/ Engineering (R)

4.3. Verify change in Production per verification plan

Description: The Operations/ Engineering team follows the steps in the verification plan listed in or linked from the Change Request to confirm the changes function as expected in the Production environment.

Input: Change installed in Production

Output: Changes verified in Production

Roles/Responsibilities: Operations/ Engineering (R)

4.4. Changes function as expected?

Description: After completing the verification steps, the Operations/ Engineering resource determines if the changes function as expected in the Production environment.

Input: Changes verified in Production

Output: Determination if the change functions as expected in Production

Roles/Responsibilities: Operations/ Engineering (R)

4.5. Rollback plan instructs to back out release?

Description: Each Change Request includes a detailed rollback procedure, which specifies the specific steps to take if the change does not function as expected. Some points of failure may require a rollback, while others do not, as some features can be mitigated without impact or toggled off. The verification plan dictates the action to take at various points of failure.

Input: Implemented changed that does not function in Production as expected

Output: Determination to roll back the changes from Production or to leave the change installed in Production

Roles/Responsibilities: Operations/ Engineering (R)

4.6. Complete post-maintenance activities

Description: The Operations/ Engineering team executes all post-maintenance activities to wrap up the change implementation.

Input: Change installed into Production

Output: Completion of post-maintenance activities

Roles/Responsibilities: Operations/ Engineering (R)

4.7. Change completed within the specified window and without impact?

Description: Each Change Request specifies a start date and time, as well as an end date and time, which complies with the allowed maintenance windows for the product or system (or an exception with justification is granted by the Change-Management team). If a change is not completed before the end date and time specified in the Change Request, the change is considered unsuccessful. If any impact is noticed or reported during or after implementation, the change is also considered unsuccessful.

Input: Change installed into Production

Output: Assessment if the change was completed before the specified end time or caused any impact

Roles/Responsibilities: Operations/ Engineering (R)

4.8. Mark Change Request or BAU ticket as *Completed*, *Successful*

Description: Changes that are installed into Production without unstated impact and within the specified maintenance window and have been verified are considered to be successful, and the Operations/ Engineering resource marks this status on the Change Request or BAU ticket. This action notifies the resources in the change's Slack channel and emails resources watching the ticket.

Input: Change successfully installed into Production

Output: Change Request in *Completed, Successful* state

Roles/Responsibilities: Operations/ Engineering (R), Change Requester (I), Peer Reviewer (I), SMEs and other Approvers (I)

4.9. Follow rollback procedures

Description: Each Change Request must include a detailed rollback procedure, which specifies the specific steps to test the release and the action to take if the change doesn't function as expected. The Operations/ Engineering resource follows the steps listed in the rollback procedure.

Input: Change installed in Production that does not function as expected

Output: Changes backed out of Production

Roles/Responsibilities: Operations/ Engineering (R)

4.10. Mark Change Request or BAU ticket as *Completed*, *Unsuccessful*

Description: Changes that are installed into Production with unexpected impact, not completed within the specified maintenance window, or do not pass verification are considered to be unsuccessful, and the Operations/ Engineering resource marks this status on the Change Request or BAU ticket. This action notifies the resources in the change's Slack channel and emails resources watching the ticket.

Input: Change unsuccessfully deployed to Production

Output: Change Request in *Completed, Unsuccessful* state

Roles/Responsibilities: Operations/ Engineering (R), Change Requester (I), Peer Reviewer (I), SMEs and other Approvers (I)

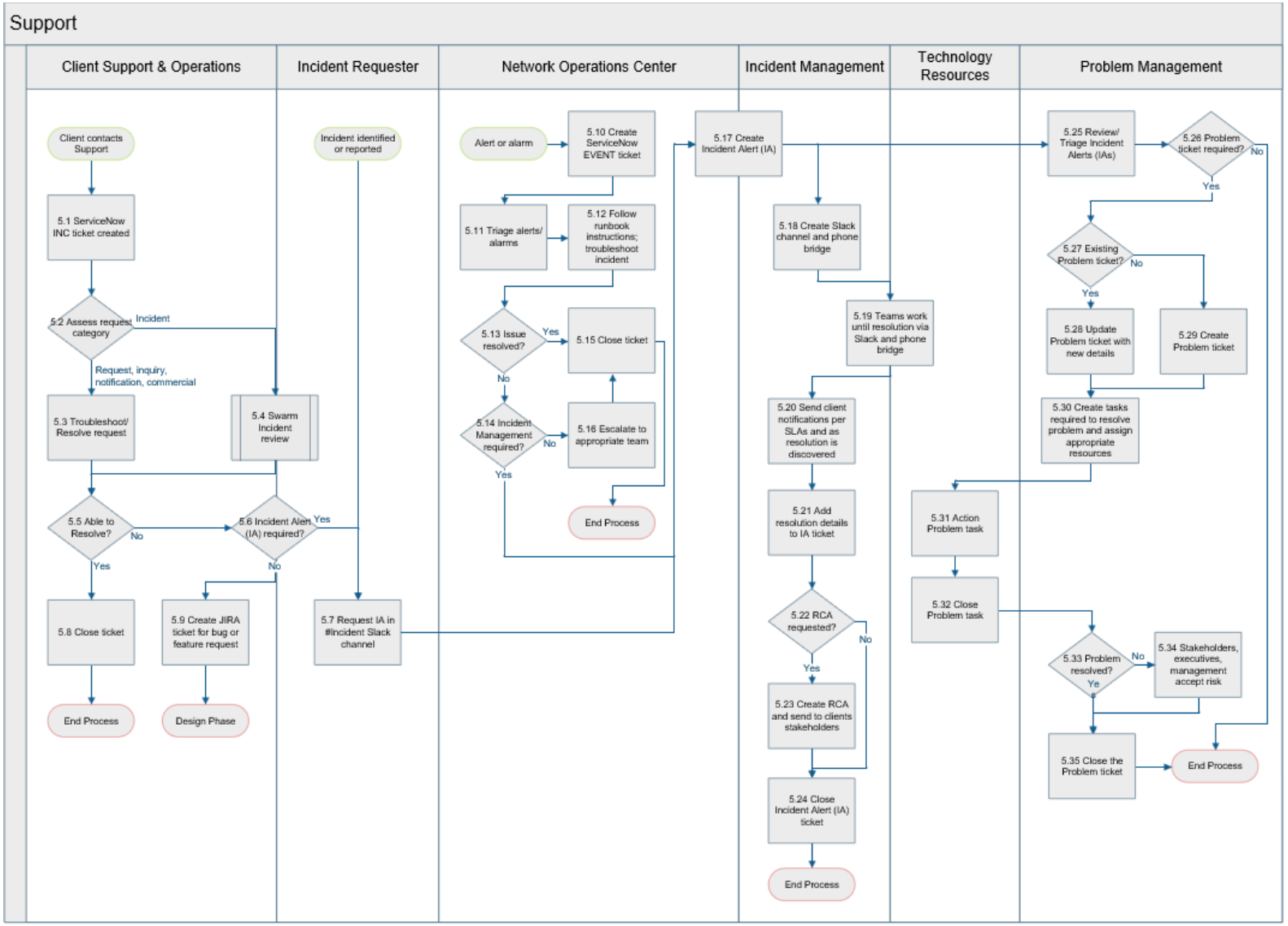
5. Support Phase

The Support Phase represents the ongoing monitoring of responses from clients and systems to ensure services remain operational. Maintaining Edgio's systems requires expertise across several teams. The Support team comprises Client Support, Operations Engineering Support, and Product Engineering Support and responds to, investigates, and resolves client-reported issues, while the Network Operations Center (NOC) proactively and reactively monitors and resolves any perceived issues with Edgio's networks and systems. For major incidents - those that affect multiple clients or systems or an impacting issue that requires immediate attention - the Incident Management team investigates and calls on technical experts across multiple teams across the organization. These incidents are reviewed by Problem Management to identify and resolve recurring issues or those with the potential to occur again.

Primary Objectives

- Identify and respond to client issues and internal alerts to services and systems
- Troubleshoot and resolve major incidents
- Identify issues with the potential to re-occur and proactively investigate a resolution

Flowchart



Detailed Activities

ARCI matrix

5.1. ServiceNow INC ticket created

Description: When a client submits an email, a client Incident ticket (INC) is automatically generated by ServiceNow and placed into the Support queue. If a client calls into Support with an issue, the Client Support team creates a INC ticket in ServiceNow.

Input: Client issue

Output: Client ticket (INC)

Roles/Responsibilities: Support (R, I)

5.2. Assess request category

Description: The client calls or emails the Client Support team with an issue, request, or inquiry, creating an Incident (INC) ticket in ServiceNow (manually for calls; automatically for email requests). The Support team assesses the type of request: Incident or non-Incident, which includes general support needs or questions, such as account-related issues, configuration issues, product purchase requests, and non break-fix issues.

Input: Client Incident ticket logged in ServiceNow

Output: Incident ticket that the Support team has investigated and classified

Roles/Responsibilities: Support (R)

5.3. Troubleshoot/ Resolve request

Description: The Support team attempts to resolve the issue. Tickets generally begin with Client Support and are escalated to Client Operations and then to Platform Engineering until a resolution is achieved.

Input: Client Incident logged in ServiceNow

Output: Incident that the Support team has investigated and attempted to resolve

Roles/Responsibilities: Support (R)

5.4. Swarm Incident review

Description: Edgio's swarming model connects the resources required to collaborate and resolve high-priority incidents efficiently, utilizing the many talents across the Edgio organization to join forces to resolve an issue. Client Support, Client Operations, and Client Success troubleshoot in real-time as a team and consult in any other internal resources needed to resolve an issue.

Input: INC client incident ticket

Output: Determination if the issue is for the Support or Client Success team to action

Roles/Responsibilities: Support (R), Technology Resources (C), Client Success (C)

5.5. Able to resolve?

The Support team swarms/ troubleshoots a client Incident (INC) and attempts to resolve the issue. Client Support, Client Operations, and Client Success troubleshoot in real-time as a team and consult in any other internal resources needed to resolve an issue.

Input: Incident that the Support team has investigated and attempted to resolve

Output: Resolved or unresolved incident

Roles/Responsibilities: Support (R), Technology Resources (C), Client Success (C)

5.6. Incident Alert (IA) required?

Description: At any time while troubleshooting a client Incident (INC), the Support engineer can request an Incident Alert (IA). This escalates the resolution of the issue, invoking the Incident-Management process, which brings in SMEs from various teams and works towards immediate resolution of the issue. Incident Management is often invoked when multiple clients or systems are affected or when services are not operational and cause major disruption to clients or LLNW.

Input: Client incident (INC)

Output: Decision to open an Incident Alert (IA) or continue resolving the INC

Roles/Responsibilities: Support (R)

5.7. Request IA in #Incident Slack channel

Description: If the Support Team decides to escalate the INC ticket to an IA ticket, the engineer or other resource posts a request in the #Incident Slack channel for either the Network Operations Center supervisor or a member of the Incident Management team to create an IA ticket. An IA escalates the issue to specialized Technology Resources for immediate resolution.

Input: Client Incident (INC) requiring escalation for immediate resolution

Output: Request for an Incident Alert (IA) in the #Incident Slack channel

Roles/Responsibilities: Support (R), Incident Management (I), Problem Management (I), Technology Resources (C, I)

5.8. Close ticket

Description: When the Support engineers resolve an Incident, they close the INC ticket, detailing the fix. If the Support resources create an associated Incident Alert (IA) ticket, they close the INC ticket when the IA has been closed. If Platform Engineering creates a JIRA ticket, the Support team closes the INC ticket at the engineer's discretion based on implementation dates and client business impact. Closing the INC ticket also alerts the client to the date and details of the resolution.

Input: Incident resolved by the Support team/ other resources

Output: Updated and closed JIRA ticket; client email of incident resolution

Roles/Responsibilities: Support (R)

5.9. Create JIRA ticket for bug or feature request

Description: For unresolvable issues, Platform Engineering creates a JIRA ticket detailing the Incident and assigns to the appropriate team.

Input: Unresolved Incident that the Support team has investigated and is unable to resolve

Output: JIRA ticket with details of the issue assigned to the appropriate team for resolution

Roles/Responsibilities: Support (R), Development (I)

5.10. Create ServiceNow EVENT ticket

Description: An EVENT ticket usually gets generated in one of three ways: automatically, manually, or email. When an alarm or alert is triggered, an EVENT is automatically generated by ServiceNow and placed into the NOC's queue. If an engineer identifies an issue, the engineer manually creates a INC ticket in ServiceNow. Any email sent to noc@llnw.com or maint@llnw.com also triggers an EVENT ticket.

Input: Network or system issue

Output: EVENT ticket in ServiceNow

Roles/Responsibilities: Network Operations Center (R, I)

5.11. Triage alerts/ alarms

Description: The Network Operations Center(NOC) engineers constantly monitor systems, alerts, and alarms. The engineers research the appropriate solution using existing materials or runbooks, previous resolutions, etc.

Input: Alert, alarm, or system issue

Output: Decision regarding the best resolution

Roles/Responsibilities: Network Operations Center (R)

5.12. Follow runbook instructions; troubleshoot incident

Description: Many issues in the NOC have documented resolution steps in runbooks. NOC engineers follow the runbooks to resolve the issue. Runbooks may instruct engineers to escalate the issue to another team for resolution or open an Incident Alert (IA).

Input: Identification of a particular issue and its associated runbook or mitigation information

Output: Execution of resolution steps in the runbook and/ or other troubleshooting techniques

Roles/Responsibilities: Network Operations Center (R)

5.13. Issue resolved?

Description: The NOC engineer follows runbook instructions, established fixes, or other documentation or knowledge to resolve the issue.

Input: System or network issue

Output: Resolved or unresolved system or network issue

Roles/Responsibilities: Network Operations Center (R)

5.14. Incident Management required?

Description: The NOC engineer can escalate a system or network issue to an Incident (IA). This may be necessary if multiple alerts have been detected or multiple alarms correlate; if the runbook instructs engineers to open an IA; or if the engineer believes the issue requires escalation.

Input: Issue requiring escalation for immediate resolution

Output: Incident Alert (IA) created

Roles/Responsibilities: Network Operations Center (R), Incident Management (I), Problem Management (I)

5.15. Close ticket

Description: Upon resolution of the incident or the creation of an IA, the NOC engineer closes the EVENT ticket.

Input: Open EVENT ticket

Output: Closed EVENT ticket

Roles/Responsibilities: Network Operations Center (R)

5.16. Escalate to appropriate team

Description: If the NOC engineer is unable to resolve an event and an Incident Alert is not appropriate, the engineer escalates the event to a more specialized team for resolution.

Input: Active EVENT ticket

Output: Closed EVENT ticket

Roles/Responsibilities: Network Operations Center (R), Technology Resources (I)

5.17. Create Incident Alert (IA)

Description: While troubleshooting a client Incident (INC), the Support engineer may decide that an Incident Alert (IA) is required. This escalates the resolution of the issue, invoking the Incident Management process and bringing together SMEs from various teams to work towards immediate resolution of the issue. IAs can be created by the Network Operations Center supervisor or the Incident Management team. Input: Client Incident (INC) or NOC issue (EVENT) requiring IA
Output: Incident Alert (IA)

Roles/Responsibilities: Network Operations Center or Incident Management (R), Incident Management (I), Problem Management (I)

5.18. Create Slack channel and phone bridge

Description: When an Incident Alert (IA) ticket is submitted, ServiceNow generates a Slack channel for collaboration using this format: #[IA number]. The Incident Management team or anyone else in the channel can add (technical) resources as appropriate, or resources can join the channel without invitation. The Incident Management team also creates a phone bridge and posts access information in the Slack channel.

Input: Incident Alert (IA)

Output: Slack channel, phone bridge information

Roles/Responsibilities: Network Operations Center or Incident Management (R), Technology Resources (I)

5.19. Teams work until resolution via Slack and phone bridge

Description: The Incident Management team and the Technology Resources collaborate to resolve the issue. These teams work together until the issue is mitigated or fixed.

Input: Incident Alert (IA), Slack channel, phone bridge

Output: Issue resolution

Roles/Responsibilities: Incident Management (R), Technology Resources (R)

5.20. Send client notifications per SLAs and as resolution is discovered

Description: The Incident Management team sends client notifications from ServiceNow in accordance with client Service-Level Agreements (SLAs) or at key milestones during the troubleshooting process. These notifications are updated with text from the engineers, which populates an Incident-Notification template to send to affected clients.

Input: Incident Alert (IA)

Output: Client notifications

Roles/Responsibilities: Incident Management (R)

References: Total Client Performance Center Guide

5.21. Add resolution details to IA ticket

Description: The Incident Management team details the resolution in the IA ticket for internal record. A post mortem may be included if the root cause is unknown.

Input: Incident Alert (IA) that has been resolved or mitigated

Output: IA with details about the solution for the issue for LLNW future reference

Roles/Responsibilities: Incident Management (R)

5.22. RCA requested?

Description: Client or LLNW resources may request a Root-Cause Analysis (RCA) for any Incident Alert (IA) to understand the impacted services, timeline of events, findings, and corrective actions. LLNW resources usually request an RCA on behalf of a client. This analysis details the

Input: Resolved or mitigated Incident Alert (IA)

Output: Determination if an RCA is required

Roles/Responsibilities: Incident Management (R)

5.23. Create RCA and send to clients and stakeholders

Description: Client or LLNW resources may request a Root-Cause Analysis (RCA) for any Incident Alert (IA). This analysis details the impacted services, the start and resolution dates, ticket numbers, impact, root-cause analysis, timeline of events, findings, and corrective actions. The Service Level Agreement (SLA) for RCA delivery is no more than five business days from the resolution of a major incident or client incident.

Input: Resolved or mitigated Incident Alert (IA)

Output: RCA document/ email sent to requesting client and/ or LLNW resources

Roles/Responsibilities: Incident Management (R), Stakeholders (I)

5.24. Close Incident Alert (IA) ticket

Description: When the issue has been resolved or mitigated and the Root Cause Analysis has been completed and sent (if requested), Incident Management or the NOC team closes the IA ticket. When the IA is closed, the Slack channel is updated with that status.

Input: Resolved or mitigated Incident Alert (IA)

Output: Closed IA ticket

Roles/Responsibilities: Incident Management (R), Technology Resources (I)

5.25. Review/ Triage Incident Alerts (IAs)

Description: Problem Management reviews all IAs for recurring issues and those that should be addressed to improve the client experience and reduce the number of tickets for the Support team.

Input: Incident Alerts (IAs)

Output: Analysis of IAs that need to be actioned

Roles/Responsibilities: Problem Management (R)

5.26. Problem ticket required?

Description: Problem Management analyzes the IAs and decides if action needs to be taken to correct the problem. Any possibility of the re-occurrence of the issue's root cause requires the creation of a Problem ticket.

Input: Incident Alerts (IAs)

Output: Decision to create a Problem ticket (PRB) or not

Roles/Responsibilities: Problem Management (R)

5.27. Existing Problem ticket?

Description: If Problem Management resources decide that an issue warrants a Problem ticket, they check ServiceNow to verify if a Problem ticket has already been created for this issue to consolidate issues and efforts.

Input: IA(s) that warrant a Problem ticket

Output: Determination if an existing Problem ticket already exists for the issue

Roles/Responsibilities: Problem Management (R)

5.28. Update Problem ticket with new details

Description: If a Problem ticket already exists in ServiceNow for the issue(s), Problem Management updates the existing ticket with any new information and ticket numbers.

Input: Incident Alerts (IAs) requiring a Problem ticket; existing Problem ticket for the same issue

Output: Updated Problem ticket

Roles/Responsibilities: Problem Management (R)

5.29. Create Problem ticket

Description: If a Problem ticket does not already exist in ServiceNow for the issue, Problem Management creates a new Problem ticket with issue details and associated ticket numbers.

Input: Incident Alerts (IAs) requiring a Problem ticket

Output: New Problem ticket

Roles/Responsibilities: Problem Management (R)

5.30. Create tasks required to resolve the problem and assign to appropriate resources

Description: Problem Management identifies required actions and monitors the deployment of a permanent resolution for the root cause. The Problem Management resource creates Problem (PRB) tasks required to address the issue. Multiple requests may be required, and these tasks are assigned to the appropriate resources.

Input: Problem ticket

Output: Problem tasks

Roles/Responsibilities: Problem Management (R), Technology Resources (I)

5.31. Action Problem task

Description: The Technology Resources work on correcting the issue detailed in the Problem task.

Input: Problem task

Output: Completed Problem tasks

Roles/Responsibilities: Technology Resources (R)

5.32. Close Problem task

Description: When the Technology Resources complete work on correcting the issue detailed in the Problem task, they note the details and results of the effort. The Technology Resources state if the issue is resolved or not.

Input: Open Problem task

Output: Closed Problem task

Roles/Responsibilities: Technology Resources (R), Problem Management (I)

5.33. Problem resolved?

Description: Problem Management monitors the progress of the Problem tasks associated with each Problem ticket. When all tasks are completed, Problem Management determines if the problem has been resolved or not.

Input: Completed Problem tasks

Output: Determination if the problem has been resolved or not

Roles/Responsibilities: Problem Management (R)

5.34. Stakeholders, executives, management accept risk

Description: If a Problem can not be resolved, Problem Management schedules a meeting with all the stakeholders associated with the product or system, including executives and management. The purpose of this meeting is for all parties to understand the ramifications of the unresolved problem and agree to accept the risk of leaving the problem unresolved.

Input: Unresolved problem

Output: Consensus among all stakeholders, executives, and management to leave problem unfixed

Roles/Responsibilities: Problem Management (R), Stakeholders (C)

5.35. Close the Problem ticket

Description: When the problem has been resolved or the stakeholders, management, and executives agree to accept the risk of the unresolved problem, the Problem Management resource details the results and decisions and closes the Problem ticket.

Input: Open Problem ticket

Output: Closed Problem ticket

Roles/Responsibilities: Problem Management (R, I)

ARCI Matrices

A = Accountable; R = Responsible; C = Consulted; I = Informed

1. Conception Phase

	SDLC Tasks	QA	Support	Feature Requester	Product Mgmt	Dev	Sales	Technology Resources
1.1	Create JIRA ticket	R				I		
1.2	Support teams attempt to resolve		R					
1.3	Assess request category		R					
1.4	Troubleshoot/ Resolve		R					
1.5	Swarm Incident review		R					
1.6	Able to resolve?		R					
1.7	Create JIRA ticket		R			I		
1.8	Evaluate existing and potential capability options			R				C
1.9	Determine if request is AFR or IFR			R				
1.10	Submit IFR JIRA ticket			R	I			
1.11	Submit AFR JIRA ticket			R	I			
1.12	Review and evaluate IFR			R	R		A, C	
1.13	Review and evaluate AFR			R	R		A, C	
1.14	Accept request?				R			
1.15	Add to Product roadmap or backlog			I	R			
1.16	Close IFR ticket with reason			I	R			
1.17	Approve accelerating the request?				R			
1.18	Approve JIRA ticket			I	R			
1.19	Request on current roadmap?				R			
1.20	Request in current backlog?				R			
1.21	Link to IFR				R			
1.22	Add to current roadmap?				R			
1.23	Add to current backlog?				R			

1.24	Convert AFR to IFR ticket				R			
1.25	Close ticket with reason			I	R			

2. Design Phase

	SDLC Tasks	Product Mgmt	Dev	QA	User Acceptance Testers	SRE	DevOps/ Engineering
2.1	Meet weekly to scope and prioritize requests: Estimate effort, resources, milestones, project timeline	R	R				
2.2	Create release		R				
2.3	Create and schedule Change Request or submit BAU ticket		R			R	
2.4	Assign request to release/ sprint		R				
2.5	Detail architecture and design in JIRA or Confluence		R				
2.6	Create test design, test cases, automation, test hosts; validate bug fixes			R			
2.7	Update code for new features and fixes		R				
2.8	Peer review code and merge into release		R				
2.9	Deploy pre-release build to QA (and sandbox) environment		R				
2.10	QA (and User acceptance) testing			R	R		
2.11	Report testing results		I	R	R		
2.12	Bugs identified?			R	R		
2.13	Create JIRA ticket		I	R	R		
2.14	Review testing results/ JIRA tickets		R				
2.15	Fix required?		R				
2.16	Fix in future or current release?		R				
2.17	Complete release; approve release on behalf of the Development team		R	I		I	I

3. Validation Phase

Change Validation

	SDLC Tasks	DevOps/ Engineering	QA	Database	Security	Platform Engineering	Dev
3.1	Review release; revise if required	R					
3.2	Approve release on behalf of the DevOps team	R					
3.3	Release impact performance?						R
3.4	Test release for performance; provide test results		R				I
3.5	Test results within margin of error?		R				
3.6	Create a JIRA ticket		R				I
3.7	Approve on behalf of the Performance QA team		R				
3.8	Conduct final regression test run		R				
3.9	Pass regression testing?		R				
3.9a	Create a JIRA ticket		R				I
3.10	Review internal release notes		R				
3.11	Confirm all *Critical*, *Showstopper*, and *Major* tickets in the release are closed		R				
3.12	Approve release on behalf of the QA team		R				
3.13	Release impact databases?						R
3.14	Review release details			R			
3.15	Pass database review?			R			
3.16	Create a JIRA ticket			R			I
3.17	Approve release on behalf of the Database team			R			
3.18	Review release details				R		
3.19	Pass security review?				R		
3.19a	Create a JIRA ticket				R		I

3.20	Approve release on behalf of the Security team					R		
3.21	Review internal release notes, QA test results; assess client impact						R	
3.22	Pass review?						R	
3.23	Create a JIRA ticket						R	I
3.24	Approve release on behalf of Operations						R	
3.25	Evaluate JIRA tickets; update release information if required							R
3.26	Update code?							R
3.27	Revise code							R
3.28	Confirm release is complete							R
3.29	Approve release							R

3. Validation Phase

Change Validation

	SDLC Tasks	Change Requester	Peer Reviewer	Emergency Mgmt.	Change Mgmt.	SMEs/ Approvers	Operations Emergency Group
3.30	Submit Change Request	R					
3.31	Review Change Request		R				
3.32	Approve or reject?	I	R				
3.33	Emergency Change?	I	R				
3.34	Review Change Request			R			
3.35	Approve or reject?	I	I	R			
3.36	Review Change Request				R		
3.37	Approve or reject?	I	I		R		
3.38	Additional approvers?				R		
3.39	Add approvers				R	I	
3.40	Review Change Request					R	
3.41	Additional approvers?					R	
3.42	Add approvers					R, I	
3.43	All resources approve or any reject?	I	I		I	R, I	
3.44	Note required information in ticket; review when provided	I	I		I	R, I	
3.45	Emergency Change?				R		
3.46	Review Change Request						R
3.47	All resources approve or any reject?	I	I		I	I	R
3.48	Revise Change Request	R					
3.49	Sign off on Change Request	I	I		R	I	

4. Implementation Phase

	SDLC Tasks	Operations/ Engineering	Change Requester	Peer Reviewer	SMEs and other Approvers
4.1	Start change at scheduled time; click *Start Change* on ticket	R	I	I	I
4.2	Follow steps listed in ticket to add change to Production environment	R			
4.3	Verify change in Production per verification plan	R			
4.4	Changes function as expected?	R			
4.5	Rollback plan instructs to back out release?	R			
4.6	Complete post-maintenance activities	R			
4.7	Change completed within the specified window and without impact?	R			
4.8	Mark Change Request or BAU ticket as *Completed, Successful*	R	I	I	I
4.9	Follow rollback procedures	R			
4.10	Mark Change Request or BAU ticket as *Completed, Unsuccessful*	R	I	I	I

5. Support Phase

	SDLC Tasks	Support	NOC	Incident Mgmt	Technology Resources	Problem Mgmt	Dev	Client Success	Stake-holders
5.1	ServiceNow INC ticket created	R, I							
5.2	Assess request category	R							
5.3	Troubleshoot/ Resolve request	R							
5.4	Swarm Incident review	R			C			C	
5.5	Able to resolve?	R			C			C	
5.6	Incident Alert (IA) required?	R							
5.7	Request IA in #Incident Slack channel	R		I	C, I	I			
5.8	Close ticket	R							
5.9	Create JIRA ticket for bug or feature request	R					I		
5.10	Create ServiceNow EVENT ticket		R, I						
5.11	Triage alarms/ alerts		R						
5.12	Follow runbook instructions; troubleshoot incident		R						
5.13	Issue resolved?		R						
5.14	Incident Management required?		R	I		I			
5.15	Close ticket		R						
5.16	Escalate to appropriate team		R		I				
5.17	Create Incident Alert (IA)		R	R, I		I			
5.18	Create Slack channel and phone bridge		R	R	I				
5.19	Teams work until resolution via Slack and phone bridge			R	R				
5.20	Send client notifications per SLAs and as resolution is discovered			R					
5.21	Add resolution details to IA ticket			R					
5.22	RCA requested?			R					

5.23	Create RCA and send to clients and stakeholders			R					I
5.24	Close Incident Alert (IA) ticket			R	I				
5.25	Review/ Triage Incident Alerts (IAs)					R			
5.26	Problem ticket required?					R			
5.27	Existing Problem ticket?					R			
5.28	Update Problem ticket with new details					R			
5.29	Create Problem ticket					R			
5.30	Create tasks required to resolve the problem and assign to appropriate resources				I	R			
5.31	Action Problem task				R	I			
5.32	Close Problem task ticket				R	I			
5.33	Problem resolved?					R			
5.34	Stakeholders, executives, management accept risk					R			C
5.35	Close Problem ticket					R, I			

Acronyms and Terms

Acronym/ Term	Definition
AFR	A formal request for a client feature with revenue and business justification and business value
ARCI	A matrix that details team responsibilities: Accountable, responsible, consulted, informed
backlog	Product Management's list of features and updates that are not scheduled on the current roadmap
BAU	Business as Usual. An update to a Production configuration item tracked in ServiceNow
Dev	Development team
DevOps	Development Operations role
IA	Incident Alert. Ticket for incidents (e.g. IA00001234)
IFR	A request for development of a feature or bug fix not tied to revenue
INC	Client Incident Ticket prefix (e.g. INC0001234)
Incident	An incident is an unplanned service interruption within Edgio's products. These issues cause interruptions to business activities and must be resolved with greater urgency. The aim is the fast recovery of the service in question.
NOC	Network Operations Center
Perf QA	Performance Quality Assurance team
Production	A non-test, live environment that hosts essential Limelight systems and services
QA	Quality Assurance team
RCA	Root Cause Analysis. A report generated for clients upon request detailing the cause and mitigation for a major incident
release	An approved and packaged version of software that has passed QA metrics and deemed ready for release to Production
roadmap	Product Management's list of items scheduled for release and implementation
ServiceNow	Edgio's internal and external ticketing system
SLA	Service-Level Agreement. Terms Edgio has established with clients regarding service availability and communication expectations
SME	Subject-Matter Expert
SRE	Service-Reliability Engineering team

Version History

Version	Writer	Approver(s)	Date	Activity
1.3	Julie Villarreal	Jeff Starr (Change Enablement) Kyle Faber (Product) Nadiia Gut (QA)	18 July 2024	Combined Performance QA and QA responsibilities
1.2	Julie Villarreal	Kyle Faber (Product)	13 June 2023	Updated text and diagrams to reflect minor changes to existing process
1.1	Julie Villarreal	Brian Zinn (Database) Chris Doehler (Dev) Eric Armstrong (Sales) Graham McCullough (Architecture) Greg Hallman (Support) James Kelsall (DevOps/Engineering) Jay Robertson (Dev, SRE) Jeff Starr (Change Management) Kyle Faber (Product Management) Mykola Dobush (Core QA) Randy Kaelber (Security) Ulyana Dobush (Control/MWS/QA)	26 July 2022	-Changed company name to Edgio -Revised Support process -Updated team/ role names
1.0	Julie Villarreal	Chris Doehler, Colin Barr, Ed Kosten, Jay Robertson, Jimmy Kelsall, Julien Vaught, Matt Stock, Mike Shulman, Mike Worthington	27 October 2021	Published original document