**Development and Delivery Best Practices**

**Implementation Kit**

Version 1.0 November 2019

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Changes | Modified By | Creation Date |
| 0.1 | Creation of Document | Indra | 08 Nov 19 |

**DOCUMENT UPDATE HANDLING**

The RPA CoE Core team will have a quarterly cadence in order to evolve the documents and to improve the guidelines based on the experiences from previous quarter. The document will have the same people assigned to review and sign-off. In case there are changes in the approvers, the new person assigned to review and update the document must be listed in the version control table of the document

**Contents**

[**1.** **Introduction** 5](#_Toc27475132)

[**1.1.** **Purpose** 5](#_Toc27475133)

[**1.2.** **Scope** 5](#_Toc27475134)

[**1.3.** **Audience** 5](#_Toc27475135)

[**2.** **Development Best Practices** 7](#_Toc27475136)

[**2.1.** **Development Roadmap** 7](#_Toc27475137)

[**2.1.1.** **Development Monitoring** 7](#_Toc27475138)

[**2.1.2.** **Design of Solution** 7](#_Toc27475139)

[**2.1.3.** **Test Plan and Development Checklist** 9](#_Toc27475140)

[**2.2.** **Definition of best practices of development** 10](#_Toc27475141)

[**2.2.1.** **Tool** 10](#_Toc27475142)

[**2.2.2.** **UiPath** 10](#_Toc27475143)

[**2.3.** **Definition of QA checklist for passing between environments in testing phase** 16](#_Toc27475144)

[**2.4.** **Queuing Mechanism** 16](#_Toc27475145)

[**3.** **Agile Lifecycle Delivery Approach** 17](#_Toc27475146)

[**3.1.** **Team Configuration** 17](#_Toc27475147)

[**3.1.1.** **Product Owner** 17](#_Toc27475148)

[**3.1.2.** **Development Team** 17](#_Toc27475149)

[**3.1.3.** **Scrum Master** 18](#_Toc27475150)

[**3.2.** **Artifacts** 19](#_Toc27475151)

[**3.2.1.** **Solutions Design Document (SDD)** 19](#_Toc27475152)

[**3.2.2.** **UAT Test Cases** 20](#_Toc27475153)

[**3.2.3.** **Operations and Deployment Handbook (ODH)** 20](#_Toc27475154)

[**3.2.4.** **Release Notes** 20](#_Toc27475155)

[**3.2.5.** **Product Backlog** 21](#_Toc27475156)

[**3.2.6.** **Agile Journal** 21](#_Toc27475157)

[**3.3.** **Scrum Lifecycle – Sprint** 21](#_Toc27475158)

[**3.3.1.** **Sprint 0 / Release Planning** 22](#_Toc27475159)

[**3.3.2.** **Sprint Planning** 24](#_Toc27475160)

[**3.3.3.** **Daily Scrum** 26](#_Toc27475161)

[**3.3.4.** **Backlog Refinement** 26](#_Toc27475162)

[**3.3.5.** **Definition of Ready (DoR)** 28](#_Toc27475163)

[**3.3.6.** **Sprint Review** 28](#_Toc27475164)

[**3.3.7.** **Definition of Done (DoD)** 30](#_Toc27475165)

[**3.3.8.** **Sprint Retrospective** 30](#_Toc27475166)

[**3.4.** **RPA Delivery** 32](#_Toc27475167)

[**3.4.1.** **Process Map Creation** 32](#_Toc27475168)

[**3.4.2.** **Creation of DS/US or Product backlog** 32](#_Toc27475169)

[**3.4.3.** **Definition of Ready** 32](#_Toc27475170)

[**3.4.4.** **Development** 32](#_Toc27475171)

[**3.4.5.** **Definition of Done** 35](#_Toc27475172)

[**3.4.6.** **UAT** 35](#_Toc27475173)

[**3.4.7.** **Regression Testing** 35](#_Toc27475174)

[**3.4.8.** **Deployment** 35](#_Toc27475175)

[**3.4.9.** **Stabilization / Hypercare** 35](#_Toc27475176)

1. **Introduction**

This document includes the Agile Lifecycle Delivery Approach and Development Best Practices. It discusses the phases in the development roadmap and best practices in creating workflows, as well as each deliverables and its corresponding guidelines.

* 1. **Purpose**

The purpose of this document is to establish a standard model for the Agile Development and Delivery Lifecycle Approach. This will be followed by the development and delivery teams involved in RPA CoE.

These good development and delivery practices are core for the scalability of the operation model, obtaining:

* **Collaboration and reuse** of the developed code
* **Standardization** of processes and consequent **maintenance** techniques locally and globally
  1. **Scope**

The scope of this document covers the:

* Composition of the Scrum teams and its ceremonies
* The process of Development and Deployment of the Development Teams
* How the teams deliver enhancements and handle Backlogs for the succeeding processes to automate.

This document applies to the UiPath Studio RPA software.

* 1. **Audience**

This document is directed to the following roles:

Mandatory:

* Scrum Master
* Business Process SME
* RPA Architect
* Developers
* Testers
* Product Owner

Optional:

* Process Owner
* Process Analyst

1. **Development Best Practices**

Development best practices ensure that the workflows are of quality. This will allow each developer to follow an approach in the design that is maintainable when developing new features, enhancement or bug fixes.

* 1. **Development Roadmap**

**Development Phase (Development):**

Fundamental coding techniques and a collection of coding practices are provided. Coding Techniques are those that improve the readability and maintainability of the code, and are also called coding standards. Programming practices are for performance improvements, this is also called coding best practices.

**Test Phase (Testing):**

Task of creating test cases for the developed program to for the assessment of each functionality of the product. This is a very important phase before Production Deployment and Go-Live.

**Delivery/Closing Phase (Handover to the Deployment in Production Team):**

Delivery of project documentation and release of resources to the operations and support team. This also includes the lessons learned to as means of continuous improvement in the process.

* + 1. **Development Monitoring**

A project management platform can be considered during the development process to be able monitor the status and information of shared development and ensure that it is **accessible to all parties involved.**

If there is a tool available (JIRA, etc) locally, it is best utilized in this phase where a ticket/profile for the process must be created.

It should also have a folder by process, shared and **accessible to all parties involved**, where they should create and edit the documents of each process, thus keeping the information always updated and accurate. Use of SharePoint is also recommended.

For Agile Development Teams, the use of a Physical Scrum Board is advised as the developers see the project movement every time they do the Daily Stand-up.

* + 1. **Design of Solution**

Solution design is based in the functional definition of the process, as in Process Definition Document (PDD). The PDD represents either the AS-IS flow of the process or the TO-BE in the case of processes that have been improved, andit is created by the Process Analyst with consultation of the Business Process SME.

The solution proposal will be documented into the Solution Design Document (SDD), which is to be created by the RPA Developers and reviewed by the RPA Lead Developer. The SDD represents the TO-BE process, which shows how the robot will be executing the process defined in the PDD.

* + - 1. ***Design of Workflow***

The following chapter defines the most appropriate use case for each of the referred UiPath components.

* + - * 1. **Layout Diagrams**

Sequence

A simple linear representation of activities that flows from top to bottom. Best suited in a simple scenarios when activities follow each other.

Flowchart

Presents multiple branching logical operators that enable developers to create complex business processes and connect activities in multiple ways. Best suited for showcasing decision points within a process.

State Machine

A complex structure that can be seen as a flowchart with conditional arrows, called transitions. It enables a more compact representation of logic and UiPath finds this suitable for a standard high-level process diagram of transactional business process templates.

Global Exception Handler

Designed to be used in both small and large automation projects, for identifying execution errors and most importantly, determining the workflow behavior when such an error occurs.

* + - * 1. **Choices**

It is important to use the appropriate representation of a condition as it will have an impact on the overall visual structure and readability of a workflow.

* + - * 1. **Data**

Variables

Bound to a container inside a workflow and may only be used locally.

Arguments

Used to pass data between multiple workflows.

* + - * 1. **Naming Conventions**

Workflows, activities, variables and arguments must be named meaningfully - describing what they are used for and must follow proper naming convention.

* + - * 1. **Comments and Annotations**

Every workflow and activity must have comments and annotations that describes how they are used and their purpose in the whole process which they are used.

* + 1. **Test Plan and Development Checklist**
       1. ***Unitary Development Test Plan (Technical)***

The unit test is done by RPA developers. Each workflow within a project shall be tested individually to ensure the workflow works and is able to handle exception.

**Environment to be used:** Development

**Target-systems version:** Test (it should be a Test version with the exact specifications of the live application, a development or UAT version of the application are not suitable)

**Test data type:** Test dataset

**Responsible for the production of test scripts:** RPA Developer

**Responsible for test execution:** RPA Developer

**Responsible for test scenarios, cases and results sign off:** RPA Tester

* + - 1. ***User Acceptance Test Plan (Technical and Business)***

The acceptance test is done by a tester from the Business Unit. It is performed to check and validate if the robot performs a process correctly.

**Environment to be used:** Live

**Target-systems version:** Live

**Test data type:** Production dataset

**Responsible for the production of test scripts:** RPA Developer

**Responsible for test execution:** Business Process SME

**Responsible for test scenarios sign off:** Business Process SME

**Responsible for test cases sign off:** Business Process SME

**Responsible for test results approval:** Business Process SME

* 1. **Definition of best practices of development**
     1. **Tool**

UiPath Studio is the main tool that will be used to automate processes.

* + 1. **UiPath**
       1. ***Studio Organization***

All workflows (.xaml) should be grouped in folders where they are used.

* + - 1. ***Nomenclature Agreement***

Agreement with the objective of defining the name rules for UiPath components, stages and development elements.

* + - * 1. Processes

Process name should be in the following format: [Client/Business\_Process]

* Client / Business – refers to the business unit that owns the process to be automated
* Process – refers to the process to be automated
  + - * 1. Workflows

Workflow name should be in the following format: [Application\_Process]

* Application – refers to the application the robot will be interacting with
* Process – should be in action form (verbs) and must describe the process the robot will be executing
  + - * 1. Credentials

Credentials should be in the following format: [Application\_Credential]

* Application – refers to the application the robot will be interacting
* Credential – username or password
  + - * 1. Schedules

Schedules should be in the following format: [Process\_AdditionalInfo]

* Process – refers to the process to be automated
* Additional Info – information about the schedule, frequency and date/time of when the robot will execute the process
  + - * 1. Environments

Environments should be in the following format: [Process\_AdditionalInfo]

* Process – refers to the process to be automated
* Additional Info – information about the environment
  + - * 1. Variables and Constants
* Variables should be in Pascal case format e.g: VariableName
* Constant should be in Upper case format e.g: CONSTANTNAME
  + - * 1. Arguments

Arguments should be in the following format: [ArgumentDirection\_VariableName]

* Argument Direction – in, out and/or io (in/out)
* Variable Name – should follow Variable guidelines
  + - * 1. Annotations

Should contain the following information about the workflow:

* Main Description – how the process works
* Pre-condition – prerequisites before the workflow
* Post-condition – expected outcome from the workflow
  + - * 1. Data table and data row
* Data tables should be in the following format: [dt\_DescriptionOfTheCollection]
* Data rows should be in the following format: [dr\_DescriptionOfTheItemInCollection]
  + - * 1. Robot Framework

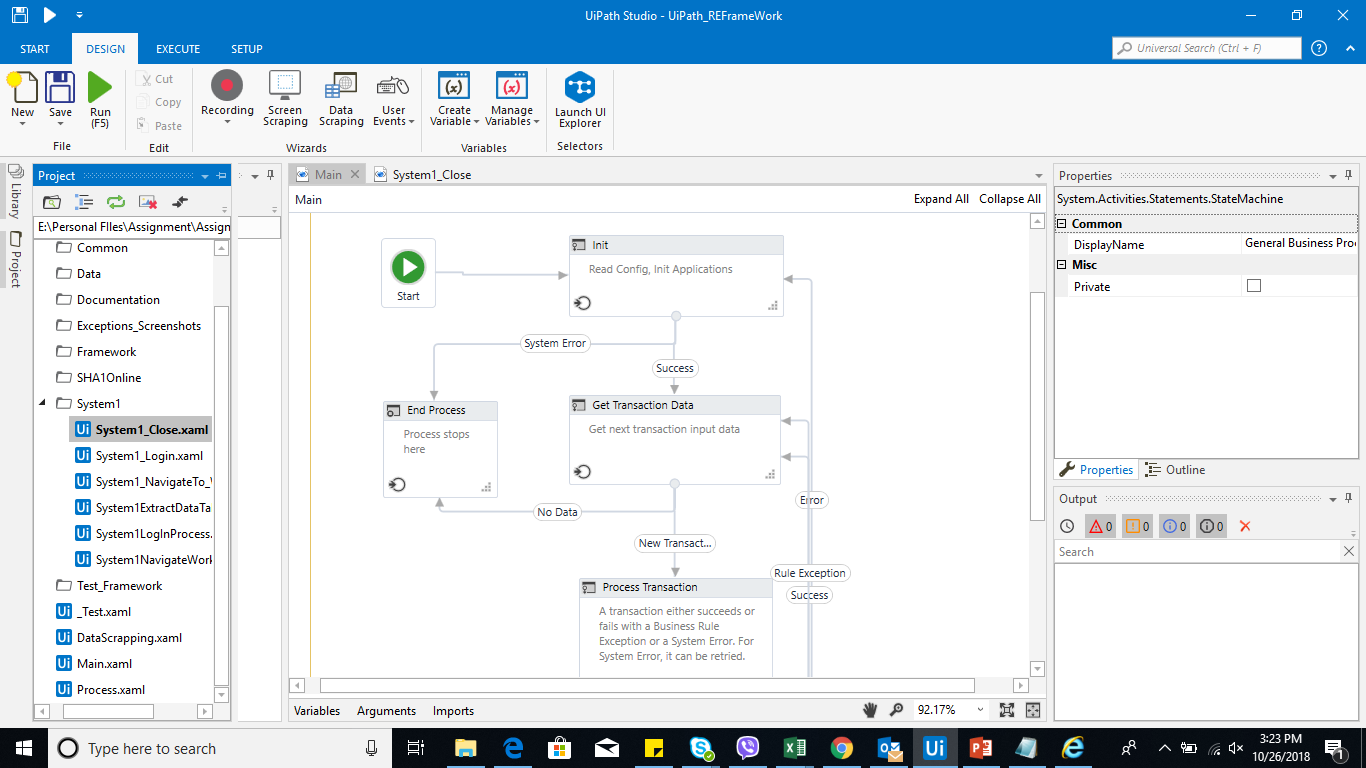
Main Architectural framework from UiPath

* + - 1. ***Development Guidelines***
         1. Design Practices

The following should be checked in a workflow to ensure that the best design practices are followed

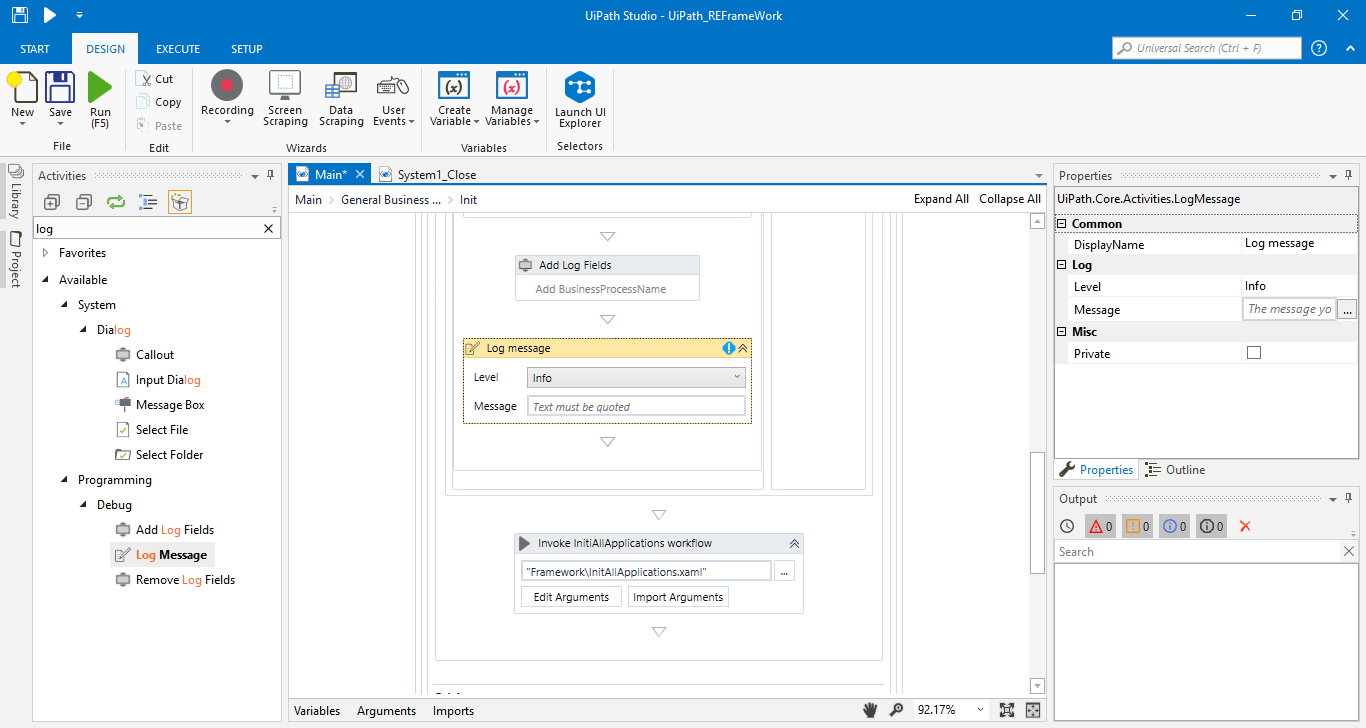
* Are all the workflows sorted in the folder of application they use?

The workflows (.xaml) should be grouped in folders by application used.



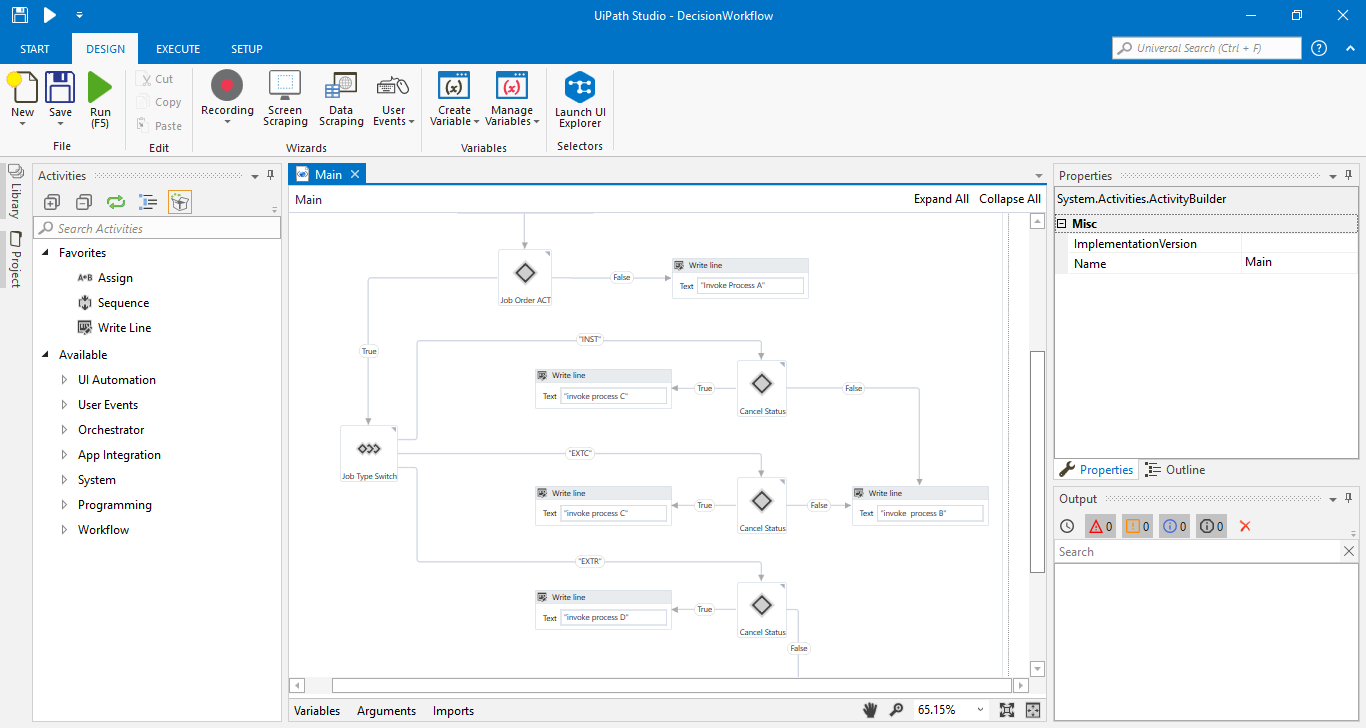
* Are there log activities in every workflow?

Log activity helps the developer and the tester check the progress of the workflow.



* Are there no nested if-else statements?

At least 2 nested if-else statement is allowed. Use flow decision instead of nested IF-ELSE statements. You can also use IF-SELECT activity for multiple cases.



* + - * Does the error and exception handling cover all possible scenarios?

All the possible scenario of the workflow that gives an error should be handled using TRY-CATCH activity.

* + - * Are there no big complicated workflows?

All big workflows should be broken down to smaller workflow as much as possible. This is to ensure that the workflow is not complicated to understand, making maintenance easier, and allowing code reusability

* + - * Are all unnecessary Write Activity and Message Activity should be deleted?

The **Write Activity** and **Message Activity** can be used for testing and debugging purposes. Activities that are not necessary for the production and delivery should be deleted.

* + - * Is proper commenting followed in the process?

Comments are used to put instructions in between activities.

* + - * Are selectors properly used to capture the data?

Selectors shall be configured to be dynamically accurate. You can use **'\*'** and **'?'** to make it dynamic.

* + - * Do the Type Into Activities have checking if the input texts are correct?

There should be a check applied for Type Into activities if the entries are correct. If not, the BOT should retry entering the text.

* + - * Do the errors in Login activities included in the exception criteria?

All errors Login activities in the Initialization Process should be included in the exception criteria.

* + - * Default timeout should be 50 or 60 seconds

Default timeout of UiPath for activities is 3 seconds. This can be reset to the recommended configuration.

* + - * Are System Errors logged as Unhandled Exceptions?

System errors should be logged as Unhandled Exception in Log Message and/or in Excel or Google Sheet logs

* + - * 1. Maintainability

The following should be checked to ensure maintainability of developed workflows:

* + - * + Is the workflow understandable?

The workflow should be simple and can be easily understood.

* + - * + Is the workflow created in such format that it could be reused in other projects?

Workflow should designed in such a way that it can be reused in other projects. I.e. by dividing workflows per functionality, and per application. Therefore each process can proceed to call each individual workflow to achieve specific process rules.

* + - * + Do the annotations and comments help understand the workflow?

Please refer to 2.2.2.2.8 Annotations.

* + - * + Are the coding convention implemented?

Please refer to coding conventions. If there are instructions that are not followed, then the process should not be signed off for production.

* + - * + Is the workflow structured in such a way that it allows easily to feature new invoke workflows?

The workflow created should be flexible in connecting with other workflows.

* + - * Are the proper commenting in the repository ([INITIAL COMMIT], [BUG FIX], [CODE OPTIMIZE]) followed during commits?

In committing the files in the repository, you should follow the proper commenting:

[INITIAL COMMIT] - Initial version of the workflow

[BUG FIX] - Update for the bug or error in the workflow

[CODE OPTIMIZE] - Enhancement or optimization of the workflow.

* + - * 1. Functionality

The following criteria should be met to guarantee designed workflow fulfills requirements

* + - * + Is the process map document followed?

The state machine or system workflow (whole project) should follow the processes (no more no less) in the process map.

* + - * + Did the workflow run continuously?

The project shall not stop at any process (unless included in the process map). There should be an identification of limitations and scope of the robotic process.

* + - * + Are the login credentials correct?

All login credentials should be configured and are accurate to avoid errors.

* + - * + Is/Are the settings for the process correct?

Initial setting of all applications and URLs involving the process should be correct.

* + - * + Are the timeouts of applications handled?

Exception handling for all timeouts should be handled.

* + - * + Did the workflow provide the correct/expected output?

The system should give all the expected output on all the possible inputs.

* + - * 1. Reports

All the reports and metrics identified by the business in the PDD should be used to produce the business reports. If a web dashboard is defined as the preferred communication method Elastic Search engine and Kibana should be used.

* + - * 1. Test Bench / Test Definition
        + Are all in/out/io arguments used in the workflow?

All the arguments used in a workflow or invoked workflow should be used.

* + - * + Are all in/out/io arguments covered in the process map?

All in/out/inout parameters of a workflow (in a process map) should be covered when creating arguments.

* + - * + Are all items in the acceptance criteria list covered?

All the acceptance criteria (per user story) should be implemented and delivered. Each criteria that is not met will cause a bug fix or update from the development side.

* + - * + Are all items in the acceptance criteria list passed the test?

All items in the acceptance criteria (per user story) should pass the test. Meaning, all the expected output on a set of input should be met.

* + - * + Are all exceptions covered?

All possible exceptions like timeout, login error, etc. should be covered as well.

* 1. **Definition of QA checklist for passing between environments in testing phase**

After the Unit and Integration Testing passes in the Development Environment, the process will then be handed over to the UAT for UAT Testing.

Once UAT is signed off by the tester, preparation for Production Deployment will begin.

* 1. **Queuing Mechanism**

It is mandatory to include a queuing mechanism for every process. Queues can be UiPath Queues, spreadsheet file, web tool that acts like a queue e.g. ticketing tools.

* Dispatcher-Performer Methodology
  + A **Dispatcher** uploads transaction to Queues. If UiPath Queue is used, the Dispatcher would be a Robot as only robots can upload to UiPath Queues. For other types of queues, the Dispatcher can either be a Robot or a human.
  + The **Performer** part of the process must be designed in a way that it can be executed by multiple robots in parallel.

This methodology ensures RPA scalability – multiple robots perform in their own multiple instances of a single business process.

1. **Agile Lifecycle Delivery Approach**

The activities and the roles involved and their responsibilities in each activities in the whole agile lifecycle delivery lifecycle is included in this section. The inputs and the deliverables for each activities and the responsible person are also in this section.

* 1. **Team Configuration**

Basic team configuration of the Scrum team is: the Product Owner, the Scrum Master, and the Development Team.

* + 1. **Product Owner**

The Product Owner or the PO is typically the project’s key stake holder. It is responsible for:

* Maximizing the value of the product, i.e. the Robotic process, and work of the Development Team
* Maintaining and updating Product Backlog
* Writing User Stories

In addition, this role:

* Is responsible for the success of the product
* Is the one person representing the stakeholders, users, customers (Business Unit)
* Decides what features go into the product – Develops the backlog
* Decides the order in which content is delivered – Orders the backlog
* Works with development team to determine specific sprint content
* Communicates expectations
* Sets project vision, project goals
* Is available to the development team throughout the sprint
* Helps teams make decisions
* Helps the development team to produce estimates work necessary to deliver product increments;
  + 1. **Development Team**

The Development Team should be:

* Delivering a potentially releasable increment of feature of the product (Robotic Process);
* Self-managing and cross-functional

Common composition of the RPA Development team can be seen below:

* + - 1. ***RPA Lead (Optional)***

Serves as a Senior or Lead developer in the team and guides the team regarding the Development and Delivery Standards. It reviews the RPA codes and also help in the development of the product (Robotic process).

* + - 1. ***RPA Developers***

The developers are the ones developing the product (Robotic process). They are responsible for creating and programming the workflows for the Robot to follow. They do unit and integration testing before they deliver the robotic process package for RPA Testing. They assist in UAT by fixing all bugs seen by the UAT tester (End users) when running the robotic processes. They are also the first line of support during the Hypercare period of the Robot in production.

* + - 1. ***RPA QA (Quality Assurance) or RPA Testers***

RPA testers are responsible for end to end testing of the integrated workflow of the delivered package or feature of the developers (RPA Lead and RPA Developers). They create test cases and/or test scripts to test the functionality of the product (Robotic process). They also help or assist in UAT in identifying test cases and fallout that are valid or not.

* **Internal**

Internal testers are responsible for the internal testing phase – which are UAT and the optional Regression Testing.

* **Testing Center of Excellence**

TCOE can be tapped for UAT and Regression Testing, during such instances they are involved already in Sprint Planning.

For cases that there is a need for the creation of test data, please refer to the TCOE Primer – Service Catalog.

* + 1. **Scrum Master**

The Scrum Master is considered to be the Scrum “Champion”, whose role is to coach or facilitate scrum ceremonies, and should possess servant-leader qualities. Other functions are:

* Being an instrument of motivation, and embodies agile values.
* Encourages and demands self-management and personal accountability and responsibility
* Enables cooperation across all roles and functions
* Removes barriers
* Shields team from external noise
* Coach and Cheerleader
* Facilitates the process (No coercion or dictating)
* Participates and facilitates scrum ceremonies

***Scrum Master’s Service to the Product Owner:***

* Help find ways to effectively manage Product Backlog in order to maximum value of work to be done
* Understanding product planning in an empirical environment
* Helping Scrum Team understand the need for clear and concise Product Backlog items

***Scrum Master’s Service to the Development Team:***

* Coaching the Development Team in self-management and cross-functionality
* Helping the Development Team to create high-value products
* Help remove impediments to the Development Team’s progress
* Facilitating Scrum events as needed or requested

***Scrum Master’s Service to the Organization***:

* Leading and coaching the organization in its Scrum adoption
* Planning Scrum implementations within the organization
* Helping employees and stakeholders understand and enact Scrum
* Working with other Scrum Masters to increase the effectiveness of the practice of Scrum in the organization
  1. **Artifacts**
     1. **Solutions Design Document (SDD)**

A detailed process created by the Business Analyst after Discovery stage of the process nominated for RPA. It contains the technical definition to be developed by the development team in order to achieve the To-Be process functionally defined in the PDD. The developers and the lead developer/s are the ones responsible for the creation of the SDD.

The necessary information that should be in the SDD are the following:

* Master project details

This includes name of the project, the type of robot is automated for front-office robot (FOR) or back-office robot (BOR), if the process uses Orchestrator and if the process can be run in parallel by multiple robots.

* Process flow of the solution

Process flow how each workflows are connected and how the data flows in the whole process.

* Packages that will be used to develop the workflows

List of the packages and the versions of the packages

* All the workflows in the whole project (with the reusable workflows identified)

The list of workflows (.xaml) that are developed and data type of the inputs and outputs of each workflows.

* + 1. **UAT Test Cases**

The Business Product Owner and the RPA Tester are the ones responsible in the creation of this document. This document contains list of scenarios that robots have to perform. There is an expected outcome for each scenarios that the robot must meet. If the robot meets the expected outcome for a test scenario, then the test scenario will be a ‘Passed’, if not, the test scenario is marked as ‘Failed’. Only when all the test scenarios pass and the Product Owner and/or Business SME have signed-off this document will the process be deployed to the robot in Production.

* + 1. **Operations and Deployment Handbook (ODH)**

This document is created by the Developers during the Development. The ODH contains information on how to operate and support the process. This document is handed over during deployment to Production and will serve as a guide for the Support.

The Schedule of when the process will be executed by robot should appear in the ODH.

* + 1. **Release Notes**

This document is created by RPA Developer for every deployment to Production and must contain general information inclusive of the package to be released. This document should include new or updated feature/process of a package. It provides traceability of the changes in production, and provides the operations the information of components to be deployed in order to execute the changes per component has described in the ODH.

The release notes should include the following:

* Release date
* Version of the release
* List of enhancements / new features in the release
* List of fixes in the release
  + 1. **Product Backlog**

It is a list of features, functions, requirements, and/or fixes for the product (Robotic process). This document dynamically changes as the product (Robotic process) and the environment where it works evolves. It also contains attributes and description of the process, order of its priorities, estimations and value. Product backlog items are also called User Stories.

* + 1. **Agile Journal**

The scrum master is the one responsible and accountable for this document. The definition of Ready, Definition of Done, Sprint Backlogs and etc. should be included in this document.

**SLAs:**

Paste this table here as editable

* 1. **Scrum Lifecycle – Sprint**



The core of the Scrum is the Sprint, a time-box of one month or less (preferably at least 2 weeks) during which a potentially feature or product Increment is created. Sprints best have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

* + 1. **Sprint 0 / Release Planning**

**Pre-requisite/s:** A signed Process Definition Document (PDD) is required before proceeding with Sprint 0.

During the start of a new product or project, it is highly recommended to create a Definition of Ready which involves a series of activities and/or preparations as a foundation needed by the Scrum team to proceed with the actual development work. This can be referred to as “Sprint 0”. The most common activities being done during Sprint 0 include but not limited to:

* Product Planning: Meeting with the Scrum Team and Business Unit to define the following:
  + Robot Vision: provides a coherent direction to the people working on the project; The value of the process being automated is also discussed in this planning.
  + High-level Product Backlog: high-level product features to be done by the Robotic process
  + Product roadmap: features and functionality may be laid out in different releases/timeline
* Establishment of the Orchestrator and the environment where the robot will work on upon deployment.
  + 1. **Sprint Planning**

The work to be performed in the Sprint is planned during Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

A simple flow of how Sprint Planning happens:

1. Product Owner presents the Product Backlog with list of upcoming features/items that the Development Team can work on the upcoming Sprint
2. Product Owner explains per Product Backlog Item (PBI) all necessary details and information (e.g. *What feature should be built or enhanced? What is the purpose/value of this feature to customers?*)
3. Discussion between Product Owner and Development Team if there are any questions, clarifications, or concerns on a Product Backlog Item
4. Accesses are requested, Test Data preparation and Environments readiness

Development Team members discusses among themselves how many Product Backlog items (PBIs) they can work on and commit to during the Sprint. Together, they create the Sprint Backlog by identifying all tasks and activities needed to complete the selected PBIs (in accordance with the Scrum Team’s Definition of Done)

Duration of Sprint Planning is usually 2 hours per week in a Sprint (e.g. 2-week Sprint may have 4 hours of Sprint Planning).

* + 1. **Daily Scrum**

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is done by inspecting the work since the last Daily Scrum and forecasting the work that can be done before the next Daily Scrum.

During the meeting, the Development Team members answer the questions:

* What work did I complete yesterday that helped the team accomplish the sprint plan?
* What work will I do today to help the team accomplish the sprint plan?
* Are there any problems or issues that prevents me from accomplishing the sprint plan?
  + 1. **Backlog Refinement**

Product Backlog Refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog Items which can be worked on for future Sprints.

This usually takes up 5-10% of a Sprint’s duration (e.g. 2-week Sprint may take .5 - 1 day of Backlog Refinement)

Things that can happen during Product Backlog Refinement:

* Detailed requirements analysis
* Splitting large items into smaller workable items
* Estimation of new items
* Re-estimation of existing items
  + 1. **Definition of Ready (DoR)**

It contains the clear criteria that a user story must meet before being accepted into an upcoming iteration. These are the list of things, documents or environments to prepare before we start the development of the Robotic process. Includes any required enabling specs, wireframes, etc. And fully meet INVEST criteria for user stories. Lastly, it should be free from external dependencies.

* + 1. **Sprint Review**

A Sprint Review is held at the end of the Sprint to inspect the Product Increment and update the Product Backlog if needed. The Scrum Team and stakeholders collaborate about what was done in the Sprint. This is an informal meeting and the presentation of the Product Increment is intended to elicit feedback and foster collaboration

The Sprint Review may include the following elements:

* Attendees include Scrum Team and key stakeholders invited by the Product Owner
* Product Owner presents which Product Backlog Items (PBIs) have been “Done” or accomplished
* Development Team demonstrates the work the they have “Done” and answers any questions or clarifications
* Attendees collaborate on what to do next, so that the Sprint Review provides valuable insight/input to subsequent Sprint Planning

Duration of Sprint Review is usually 1 hour per week in a Sprint (e.g. 2-week Sprint may have 2 hours of Sprint Review)

* + 1. **Definition of Done (DoD)**

Scrum Team members (Product Owner, Development Team, and Scrum Master) must have a shared understanding of what it means for work to be complete, to ensure transparency and quality. This is the definition of “Done” and is used to assess when work is complete on the Product Increment. Simply put, Definition of Done is a criteria checklist of activities that should be accomplished before considering a Product Backlog item to be finished or completed. When a Development Team completes Product Backlog items, it is expected that each of these items have conformed to Definition of Done.

As Scrum Teams mature, it is expected that their Definition of Done will expand to include more stringent criteria for higher quality (e.g. Passed Automation Testing, Passed Regression Testing).

* + 1. **Sprint Retrospective**

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

* Inspect how the last Sprint went with regards to people, relationships, process, and tools
* Identify major items that went well and potential improvements
* Create a plan for implementing improvements to the way the Scrum Team does its work

Duration is usually 45 minutes per week in a Sprint (e.g. 2-week Sprint may have 1.5 hours of Sprint Retrospective)

* 1. **RPA Delivery**
     1. **Process Map Creation**

Before the start of the project, a process from the Business unit should be mapped by the Process Analyst in detail in order for the developer to have a reference on what the robot should do or what process it would follow.

* + 1. **Creation of DS/US or Product backlog**

A series of User Stories should be created, valued and prioritized by the Product Owner for the Development team to estimate the difficulty of the User Stories and what User Stories will be delivered for the current Sprint.

* + 1. **Definition of Ready**

Environment where the robotic process will be developed and tested should be prior to the development stage. User stories should also be checked for readiness of the web application or the Desktop application or tools the Robotic process will be using.

* + 1. **Development**

Here the process of Development of the robotic process is performed. This includes Unit testing per workflow, Integration testing of the workflows, QA testing before proceeding with the UAT. Duration of development depends on the effort it takes to develop and the complexity of a process.

The table below shows the level of complexity and effort per type of process, whether it will be fast track, medium track or complete track.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Complexity** | | | |
| **Effort** |  | **Low** | **Medium** | **High** |
| **Low** | 1 to 5-step Processes. (Clicks and Inputs)  Process Enhancements | Transfer of Data in more than 2-5 screens (Including Clicks and inputs)  Involves 1 Web Application or Desktop Application | Requires Programming Skills (.NET) in string manipulation, working with arrays, data tables, collections, data formatting and exception handling  Transfer of Data in 5 or more screens (Web applications and Desktop Applications) |
| **Medium** | Web Scraping / Data Scraping (1 Web Application or Desktop Application) | Automation in SAP,  Mainframe Applications (Using Mainframe UiPath Activities) | Automation in Complex SAP Application (complex = UiPath unable to inspect elements)  Mainframe Applications (string manipulation) |
| **High** | Involves 2 Web applications or Desktop Applications  Transfer of data from Standard Inputs to other application | Automation in Citrix Environment  Transfer of data from Standard Inputs to other application  Java Applications (3 Web or Desktop Applications) | Automation in complex application running in Citrix Environment, Automation in Terminal Emulators (complex = UiPath unable to inspect elements)  Processing of inputs containing Free Texts (4 or more Web or desktop Applications) |

**Fast Track.** Processes with LOW Complexity and LOW Effort.

**Medium Track.** Processes with MEDIUM Effort and Complexity, LOW Effort and MEDIUM Complexity and LOW Complexity and MEDIUM Effort.

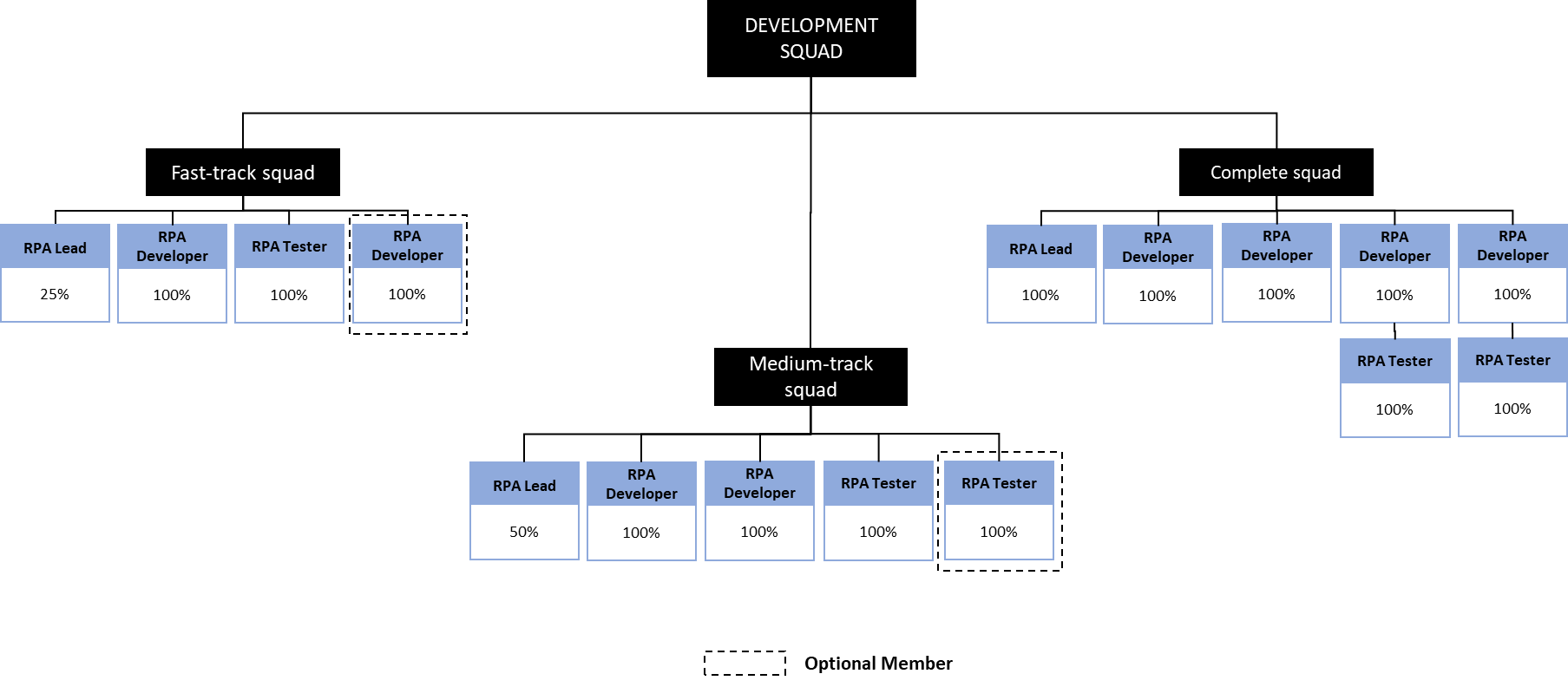
**Complete Track.** All processes with HIGH Complexity or HIGH Effort.



For SAP Automation, Setting for SAP GUI Scripting is required to Get the UI elements of the Application.

For Java Applications (Desktop), you need to install Java Bridge Plugins for Uipath. Visit this Link: <https://docs.uipath.com/studio/docs/java-extension>

The Squad for the Development team for each of the tracks are described below:



* + 1. **Definition of Done**

There should be a checklist for the definition of Done from the development team. This will ensure that the development have undergone proper testing, peer reviewed and integration tested.

* + 1. **UAT**

This activity involves a UAT tester from the Business Unit, probably an SME of the process, that will check and validate the process the Robotic process is working on. The Development team should fix every bug identified in UAT. Upon accomplishing all the test cases, a UAT sign off from the UAT tester and finally from the Product Owner is requested for documentation purposes, which will trigger the deployment preparation process.

* + 1. **Regression Testing**

Usually an optional testing phase. This is done for:

1. Robots working on Multiple or Overlapping processes to make sure that the currently developed process for the robot is working properly in conjunction with the current process.
2. Robotic process enhancement to ensure that the new feature enabled for the robot together with the previous features are working perfectly.

Pre-requisites:

* Test data and new schedule, for case 1.
* Test data for the previous process and the current enhancements, for case 2.

There should be enough number of runs for the robotic process in the production environment covery all the cases under regression testing to ensure that issues will be captured, if there is, in the production environment.

* + 1. **Deployment**

This involves the deployment of the Robot in the Orchestrator, on premises or in cloud. The environment of the production should ideally be the same as the testing environment.

The Releaser from the RPA CoE Squad will be the ones uploading the package to the Orchestrator. The Development team will be the support team until hypercare or stabilization phase ends.

* + 1. **Stabilization / Hypercare**

Upon deployment, stabilization phase for the robotic process is established. It could be a support for any bugs or errors the Robot will encounter during the live run of the Robotic process in production. A bug is any feature required in the process functional definition (PDD) and proposed in the process technical solution (SDD) that doesn’t work as expected. Any feature that prevents the robot to execute the business process correctly but has not been identified during the definition phase of the process is not to be considered a Bug, but a New Feature, and should follow a diferent procedure then the Bugs. The timeline of hypercare will depend on the frequency of the automated process.

|  |  |
| --- | --- |
| Process Frequency | Hypercare |
| 24/7 | 1 week |
| 8/7 | 2 weeks |
| > 4 / 7 | 3 weeks |
| 2-3 times a week | 1 month |