

Process Mapping Guide

Overview

The Process Mapping Guide will help teams document a process for automation. Process mapping is for turning operational requirements into technical requests, referencing during quality assurance and/or maintenance, and transferring knowledge across stakeholders.

The main goals of process mapping are to identify:

- Systems used and the associated access requirements
- Discrete steps performed today (i.e. what URL is used, what drop down is selected, etc.)
- Rationale at important decision points (i.e. why do you select “Aetna” based on the information you are reviewing)
- Parts of the process that are misunderstood
- Inputs (i.e information needed to make decisions or complete the process)
- Outputs (i.e. information that is gathered during and at the end of the process)

Shadowing Basics

The first step in process mapping is shadowing the process. The following are important considerations before and during the shadowing session:

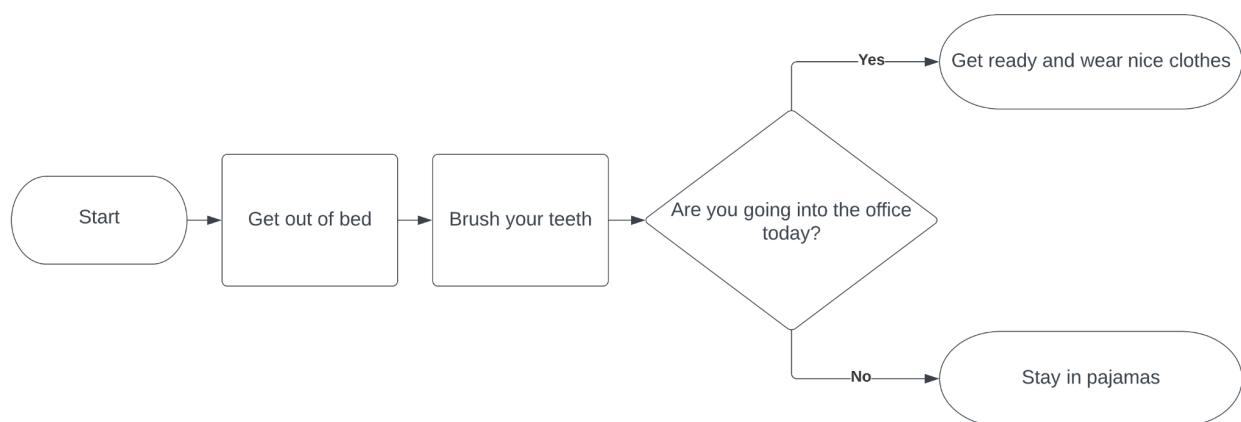
- Schedule time with the best subject matter experts in the organization
 - Typically this includes staff who perform the process and their direct supervisor
 - Email participants ahead of the shadowing call outlining the workflow name, workflow description, and an overview of the shadowing process to set expectations early and ensure all necessary attendees are included
- Explain the goals of the shadowing process (see above)
- Use recording software such as Zoom, Microsoft Teams, etc. so the process can be revisited as long as it is permitted by EHR/application compliance agreements
- Take screenshots of each distinct action, decision, etc. to reference in the process flow as long as it is permitted by EHR/application compliance agreements
- Have the subject matter expert articulate every step taken, the rationale for the step, what information is needed to perform the step, and the goal of completing the process
 - Ensure that each action taken is out of necessity, and not by preference
- Each decision and/or exception must be covered so every outcome is mapped

Process Mapping Basics

A process map is a step-by-step depiction of a process. A simple example would be starting your morning:

1. Get out of bed
2. Brush your teeth
3. Are you going into the office today?
4. If yes, get ready and wear nice clothes
5. If no, stay in pajamas

Visually, this would be represented left to right (or top down) with connecting arrows to show steps and decisions. The example below is a process map of the steps outlined above. As you can see, the end result is a clear guide of the steps, decisions, and end-points.



Nodes

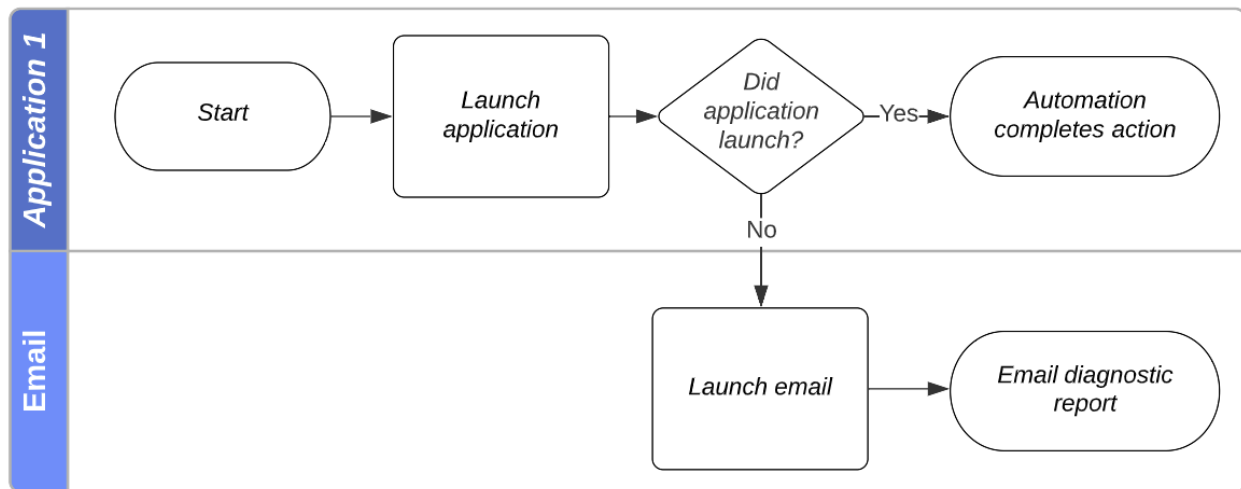
Different “nodes”, or shapes, indicate particular actions within the process map. These nodes are common across all industries. You can build a process map with common nodes such as:

-  **Terminal/Terminator** - Indicates the beginning or end of a process
-  **Process** - Assigns a specific action within the process
-  **Decision** - Indicates a choice that results in more than one possible action

For more examples of shapes and their intended use, see [here](#).

Swimlanes

Swimlanes should be used whenever the automation is moving from one system to another. This allows stakeholders to more easily identify what application the automation is using for each step or decision. The delineation helps with automation development as well. For example, access to each system may require its own set of credentials.



Tools

Ask the Information Technology department if you have access to any process mapping tools in your organization. [Microsoft Visio](#), [Lucid Charts](#), [Creately](#) are commonly used. Excel, Word, or other Microsoft products can also be used, although they have limited functionality. Lucid Charts has a great [tutorial](#) on creating process maps for more information.

Process Mapping for RPA

Process maps are easily applied to automation initiatives, however current state process maps are typically not ideal for building RPA. The process should be split into two phases to ensure the automation runs efficiently, requires less maintenance, and has a low failure rate.

- ☐ Current State: The process is mapped as an exact representation of the manual process performed today. Current state process maps are shown with no predetermined or upstream improvements.
- ☐ Desired State: The process is mapped as a streamlined representation to better suit RPA implementation and sustainability. The process may have fewer steps or show improvements.

Consider the following factors when building a process map for automation.

Standardization and consolidation of upstream data

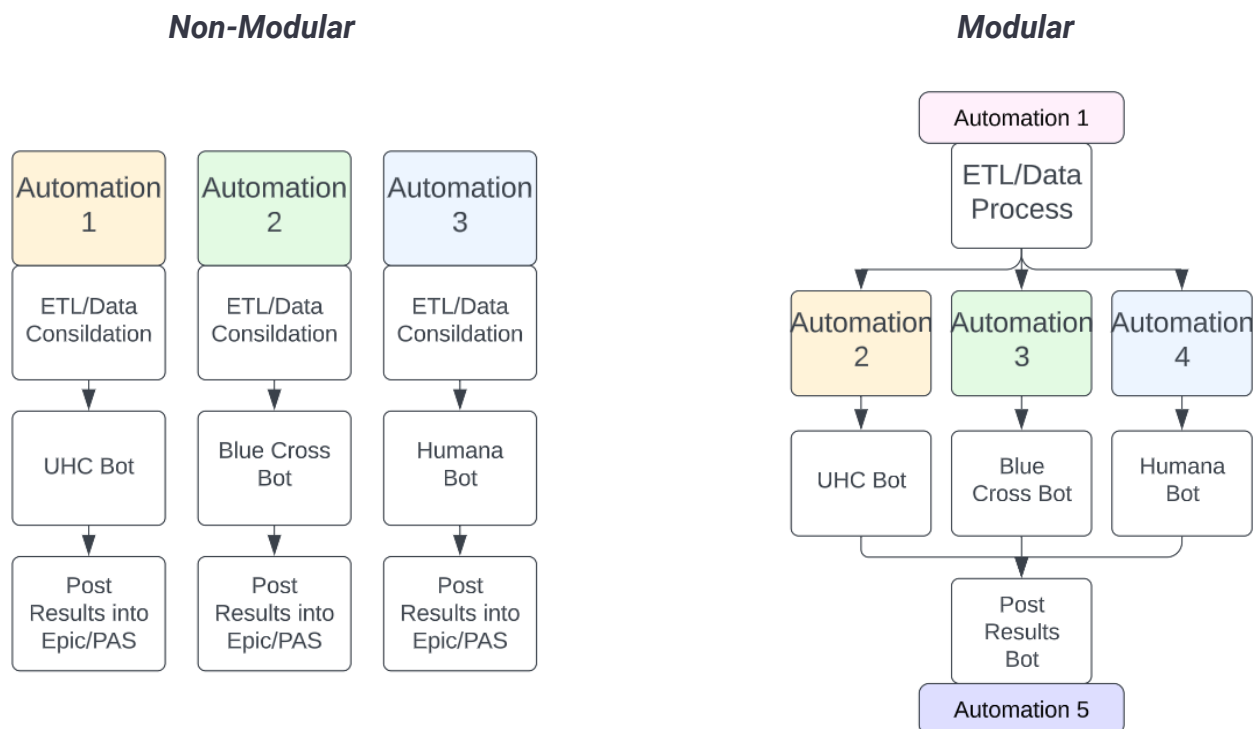
The benefit of standardizing and consolidating particular data points is to reduce the number of decisions an automation has to make. Although decisions can be coded, the goal is to reduce the decisions and steps as much as possible. This will improve build time, complexity, and maintenance. When decisions, data, or processes cannot be standardized or consolidated,

organizations should implement a modular design. Below is an example of standardizing upstream processes:

- ☐ Current state process: Process begins in EMR/EHR reviewing coverages. Associate reviews which payer is involved to determine which portal to visit.
- ☐ Desired RPA process: Automation starts at payor portal, not in EMR/EHR. Payor details are available via export to determine which portal needs to be accessed.

Modular design

RPA uses the term modular design to describe the action of splitting one complex process into multiple less complex processes. Reducing the number of inputs, variances, decisions, outputs, etc. will make the automation more efficient over its lifetime. RPA can also leverage a modular design to consolidate shared functions across related automations. The purpose of doing this would be the reduction in maintenance times/cost as well as improved scalability. See below for a visualization of modular design:



The non-modular example is the typical approach to building an automation. The modular example is an improved approach, because we are gaining efficiencies by using shared functions across similar automation types. The efficiencies are the consolidation of the “ETL/Data Process” and “Post Results”. These improvements result in the following:

- ☐ Maintenance Cost Reduction: If the “ETL/Data Process” or “Post Results” function breaks in the non-modular example it will break across all automations, requiring multiple fixes. If it breaks in the modular example, it only needs to be fixed once.
- ☐ Improved Scalability: The automation can be built faster and expanded to additional payers because the “ETL/Data” function and the “Post Results” functions are being handled in separate automations. Additional payers can also be handled by other solutions such as an API. In our example above, UHC could use an API and Blue Cross could use RPA.

Reducing clicks and actions

User interfaces are decided for human users who typically view, then select a desired action/process. Automations can be built following the same steps as a human so long as there is sufficient granularity. That said, automations benefit from shortcuts like Macros, Key Binds, and templates because it limits the number of details to view, then select. As a rule of thumb, reducing the amount of clicks and actions will result in improved efficiency and maintenance. An example would be:

- ☐ Current State Process: An employee selects specific medical records in the EHR’s ROI tab based upon CARC/RARC code indication from a payer.
- ☐ Desired RPA Process: The organization’s IT team creates a pre-populated template in the ROI tab for the automation. The automation selects the template in the ROI tab based on CARC/RARC codes from the payer, thereby avoiding a per record decision/selection by an automation.

Resolution and format of native screens

An overlooked aspect of process mapping is capturing screen resolution (including zoom levels) as well as application color and format. If the mapped process is built on screenshots from a user who customized their user interface, then the automations instance of the application (e.g. the service account) will need to be identical. If they do not match, the automation will fail or take longer to determine what action to take. Make note of resolutions and user interface customization when process mapping to ensure a seamless transition to production.