TARPON HEALTH

Solution Architecture
Part 1

Agenda

- The Solution Architecture role
- Shadowing
- Process mapping for RPA

Today's goals:

- Prepare members on a set of Solution Architecture responsibilities

Solution Architect role

The RPA Solution Architect is responsible for identifying, designing, and planning automations for development

Core Responsibilities

- Use case identification
- Shadowing
- Process mapping and optimization
- Solution evaluation and design
- Process Definition Document (PDD) creation
- Automation build and testing management

Shadowing

Shadowing is the first step, but it requires knowing where you're going and what your development team will need.

Slow down with end-users

- Schedule time with SMEs and supervisors
- Set expectations
- Explain the goals

Mechanics

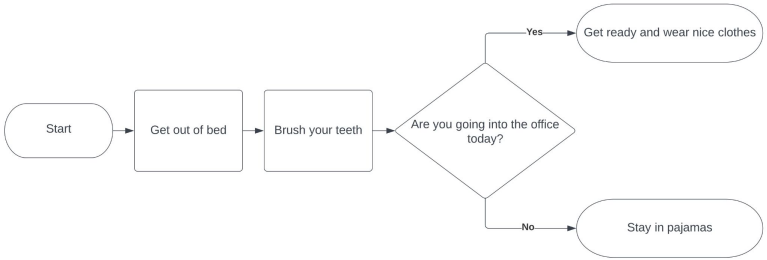
- Record the session for revisiting
- Snap screenshots of important actions or decisions
- Focus on rationale and required information

Process mapping is for turning operational requirements into technical requests, referencing during quality assurance and/or maintenance, and transferring knowledge across stakeholders.

Process mapping basics

Master the basics, especially if you've never built a process map before.

Nodes



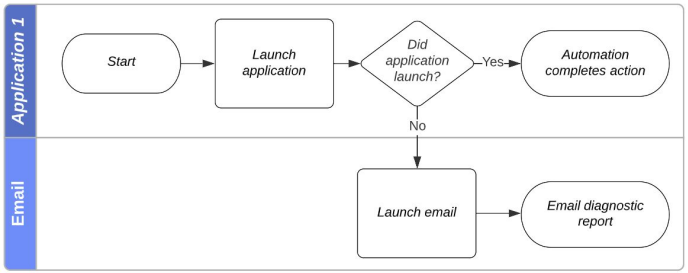
Terminal/Terminator - Indicates the beginning or end of a process

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



Process mapping for RPA

Start with mapping the process in its current state then outline the best process for the automation.

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.

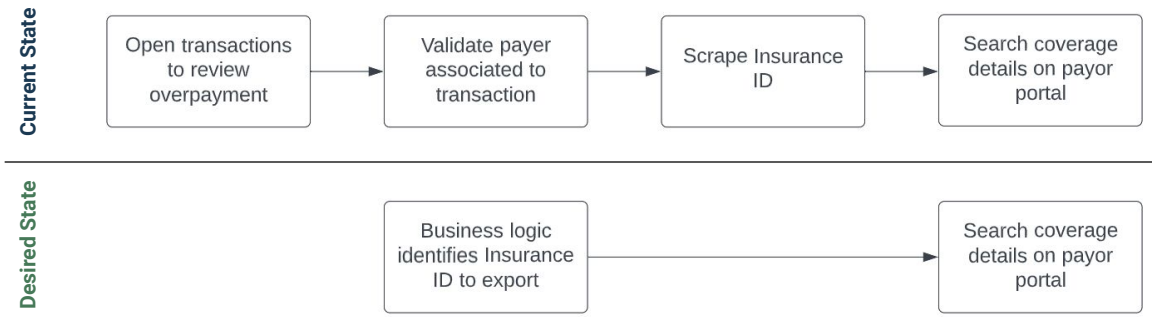
Discussion:

1. How important is it to capture both current and desired states?
2. How do you learn what is most efficient in an automation?
3. Process improvement: Don't build an automation on a poorly designed process

Standardizing upstream data

The goal is to reduce the number of decisions and steps as much as possible.

Example - Overpayment Identification



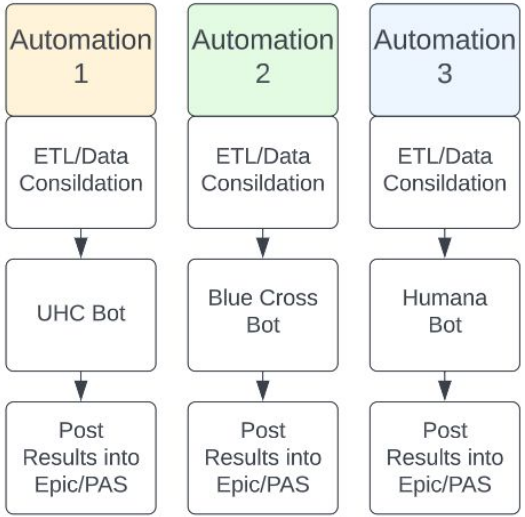
Why?

Reduced steps and decisions improves build time, complexity, and maintenance.

Modular design

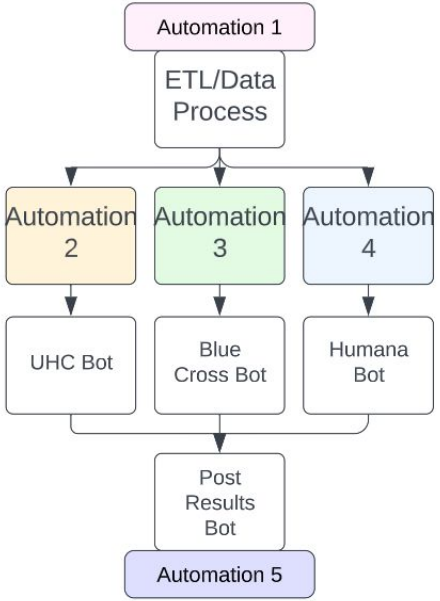
Splitting a very complex process into multiple, simpler processes will save you time and effort down the road.

Non-modular example



Versus

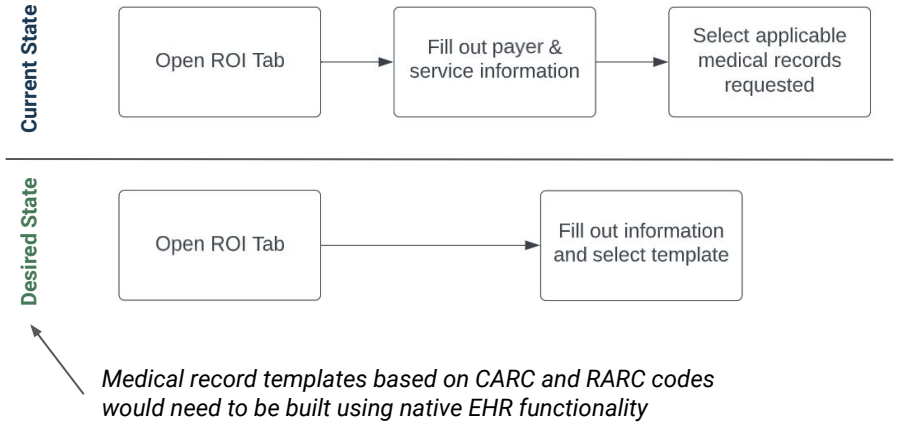
Modular example



Reducing clicks and actions

Any reduction in clicks and actions is beneficial in the long run.

Example - Medical Records



Discussion:

1. What are other examples?
2. How do you account for the prep work needed?

Resolution and format of screens

As they say, “the devil is in the details”.

Images	OCR engines	Output text of OCR engines
	*Teseract*	English is a West Germanic language
	Asprise	English is a West Germanic language
	*Teseract*	Eng sh ‘S a wax Grrmamc lnnguzge
	Asprise	Eng Iish is g W t CnmgniL Iq e

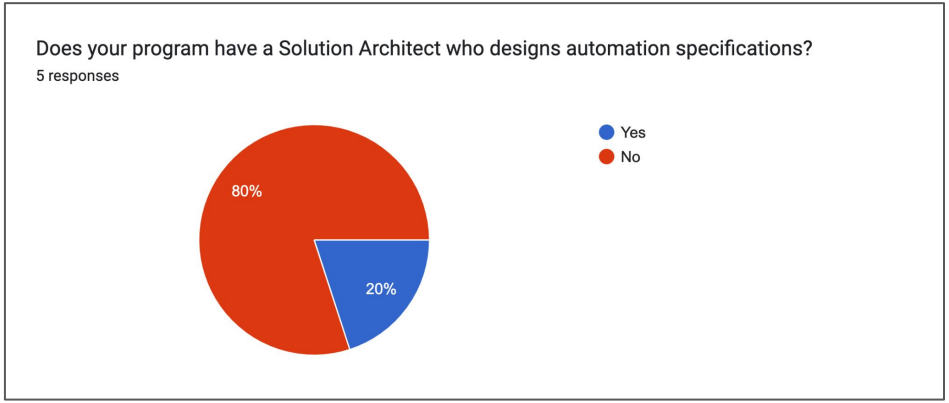
Caution:
If using OCR, screen resolution, color, and formatting can cause issues

Key concept: Make sure what you are process mapping will be consistent with the interface the automation is using

Tarpon Community

Only one organization in the Community has dedicated a position to Solution Architecture.

Question 7: Tarpon Health Governance Survey



Who's performing these functions in your organization?

Tell us how we did

1 - Least valuable

10 - Most valuable

We value your feedback!

TARPON HEALTH

Benjamin Reigle | benjamin@tarpon.health

James Stanford | james@tarpon.health