



Solution Architecture

Part 3

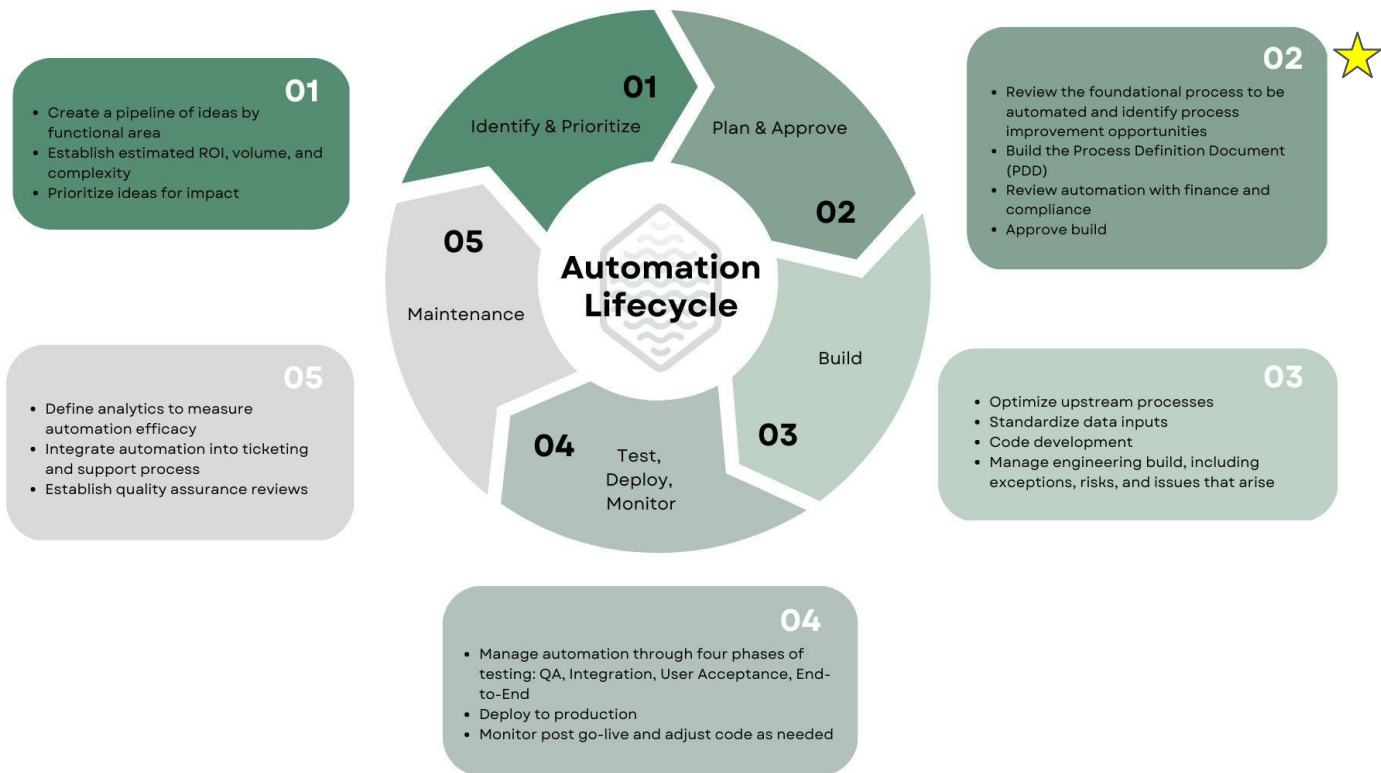
Agenda

- Solution Architecture recap
- Toolkits in action - examples
- Process Definition Document (PDD)

Today's goals:

- Articulate how all the toolkits work together when designing an automation
- Define how to use the Process Definition Document (PDD)

The Tarpon Health Automation Lifecycle



Solution Architect role

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

Core Responsibilities

- Use case identification - *Automation Use Case Webinar (3/16)*
- Shadowing - *Solution Architecture, Part 1 Webinar (3/30)*
- Process mapping - *Solution Architecture, Part 1 Webinar (3/30)*
- Solution design - *Solution Architecture, Part 2 Webinar (4/14)*
- Process Definition Document (PDD) creation - **Today**
- Automation build and testing management - *Future*

Toolkits in action



We will be sharing examples of the important steps needed to move an automation from idea to 'ready to build'.

Identify + Prioritize

Establish estimated ROI, volume, and complexity and prioritize for impact

Shadowing + Process Map

Review foundational process, map the process in its current state and then for the automation

Design

Transform a process into a technical document for engineering

PDD

Capture all the details required for engineering and sign-off

Step 1: Identify and prioritize

The first step in the Solution Architecture process is identifying and prioritizing the automation. This gives the organization confidence the idea is worth the investment.

Identify + Prioritize

Must be completed before moving to Process Mapping:

- Volume validated by export
- Positive year 3 ROI using the identify toolkit
- Automations are prioritized based on organizational goals

Key Takeaways

- Two approaches: bottom-up and top down
- Validate volume through an export
- Prioritize against other ideas based on what your organization is trying to achieve
- Fail fast mentality

Step 2: Shadow and process map

The second step in the Solution Architecture process is shadowing and process mapping. Process mapping allows the organization to understand feasibility and accurately design.



Shadowing + Process Map

Must be completed before moving to Plan:

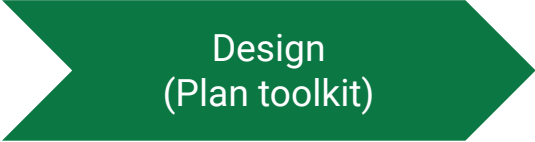
- SME and manager sign-off, which includes a manual account review
- Desired process reviewed with engineering for feasibility
- Process optimization discussion

Key Takeaways

- Spend time up front identifying the right SME and manager
- Compare your process map to the steps taken by the SME for validation
- Align data elements to the process for engineering

Step 3: Design the process

The third step in the Solution Architecture process is evaluating and designing the process, with an emphasis on data mapping and exception handling.



Design
(Plan toolkit)

Must be completed before moving to PDD:

- Export, file location, and timing
- Queuing and logging details are documented thoroughly
- Exception handling validated with engineering

Key Takeaways

- This document supports the PDD
 - The format is more useful and functional for development
- Takes time, iteration, and multiple stakeholders to accurately document the exceptions

Step 4: Capture it all in the PDD

The fourth step in the Solution Architecture process is creating a Process Definition Document (PDD) that captures the requirements for sign-off and build.



PDD

Must be completed before handing off to engineering:

- Review and sign-off from SMEs, engineering, and Governance

Key Takeaways

- Some information is duplicative, pull directly from previous work
- This is an administrative and a solution architecture tool
- The PDD is important for change control

Elements of the PDD

The PDD contains administrative and practical elements to help manage the automation build and beyond.

Change Control (New) • Directions on managing the document, revision history, and future initiatives	Introduction • Purpose, objectives, contacts and prerequisite checklist
Process • Applications being automated, Data inputs, process map, exception management, out-of scope	Metrics (New) • Automation and business performance ○ Automation: Run time ○ Business: Labor hours
Approval (New) • Sign-off before build	

How to use the PDD

Treat the PDD and plan templates as a living breathing document.

Signature: Getting sign-off on the PDD is an important milestone as it signals authority to begin build and make major updates to the design.

Design source of truth: The PDD (and Plan templates) document should be revised anytime a fix, design change, or code change is needed.

Expectation setting: The metrics section outlines the automation performance expectations, which should be tied to a performance dashboard.

Discussion:

1. Who is accountable if the automation doesn't perform as expected?
2. Why is tracking changes over time important?
3. Should the PDD be updated if there is a minor design change made?

Tell us how we did



1 - Least valuable

10 - Most valuable

We value your feedback!



TARPON
H E A L T H

Benjamin Reigle | benjamin@tarpon.health

James Stanford | james@tarpon.health