Pre-Project Phase

Roadmap

Define a Roadmap (Integration 1, 2 etc) and keep the different steps digestible in small iterations. Make sure to set clear expectations for these iterations.

Here, it is advisable to focus on Content rather on Timeline.

Separation of concerns

Clearly define who does what.
The Project manager should not be the account manager, similarly the Solution Architect should not act as the Developer.

Define 1 Lead Developer for Code Review - Code refactoring if needed. Also, define a QA person for ensuring quality. QA should be an independent person for spotting errors and disruption testing. Avoid Handover of people.

This applies for Client as well - each person must have a specific function to fill that is documented in the project kickoff. Clear Project Team definition on Clients side. Business Owner, Project Manager. We need 1 person that gives us the orders and makes decisions.

Need a format that's easy to create and consume? Consider using a RACI chart

User Stories

Be precise when writing User Stories by setting limitations/exclusions of the solution.

This means including positive user stories (as a User, I want to be able to view my current Time Entries summary, so I know how many hours I still need to time), as well as negative assumptions (the REST application will not authenticate against Azure directly but go through the client backend, as detailed in User Story UA-15)

This allows to set expectations correctly and helps the customer to amend requests early on in case something is not according to his wishes. Think about what we cannot do rather than solely on what we can.

Data Management

When handling existing organizations or integrations with other systems, it is important to include a Master Data Management Design in your project.

An MDM's main purpose is to define the single source of truth - per table, per field, per integration flow. A MDM allows for accurate and trusted business information of the right quality level, in the right form, at the right time.

Key elements are to define what controls the data at which point, and what happens if the data is not accurately synced.

Project History and Accountability

While deploying bigger projects, it is important to be able to track who deployed what, and in which version each element is.

In order to do so, the recommended approach is to deny any modification by changeset, instead relying on SFDX deployment chains via Git - including admin-only deployments like fields, Flows, or Process Builder. This means that a single person should be identified who will be in charge of scheduling deployments. With that said, have a back-up person so that your main person can take vacation!

Deployments themselves should be scheduled ahead of time, including for bug fixes. They should not number more than one per week, in order to allow for UAT and regression testing.

Assumptions

Are YOUR FRIENDS.

Make assumptions BUT WRITE THEM DOWN AND SHARE THEM WIDELY!!!!. Clearly defined what are the risks if they change the Business owner and what is the consequence. Same for technical assumptions, etc.

Revision #4
Created 7 July 2019 19:18:25 by Windyo
Updated 7 March 2023 00:47:38 by thejamesjames