Analysis –
Optimization/Effort Reduction

TF Web Application – New Architecture

Option 1

- Replicate Functionality as-is.
- No Design changes
- All UI/Services implemented as currently designed using React/Redux & Spring MVC
- Estimated Effort - ~6000 hrs.
- Estimated Completion Date – 4/6/2021

Option 1

- Concerns
  - TimeLine/Effort
  - No optimization/automation opportunities explored.
  - Project structure
    - Cascading impact of unknowns.
    - Positive impact of code reuse not captured/quantified.

Analysis

- Design Changes
  - Present all CRUD forms as Modals (from grid pages).
  - Present all Lookup grids as Modals (from grid pages).
  - Present all child pages (with grids) as modals.
  - Eliminate the need for related links.
    - Group pages/links together as modules.
      - ** Requires input from TF team.*
  - Eliminate recent usage (if possible)
    - Constraint - All entities have specific API for recent usage.
  - Build reusable pages and grid components per screen type*
    - Side effect – increases API complexity.
  - Potentially reuse filter components. (generic filters)
    - Use grid filters to reduce filter complexity.
  - Potentially use a form builder to build simple CURD screens.
    - Side effect – increases API complexity.

Analysis

- Design/Implementation - Optimization opportunities
  - **Type 1 Screens** (#15 areas)
    - Have a single grid and form(s) for Create/Update/Deletion of a single entity.
    - Grid complexity ranges from Low to Medium (based on # columns)
    - CRUD complexity ranges from Low to Complex (2 fields to 58 fields)
    - Design/Implementation Approach
      - Create components to reuse the following
        - Read only Grid with a link to CRUD form(s)
        - Build support for multiple data loading endpoints/APIs.
        - Build support for CRUD operations based on entity types.

Analysis

- Design/Implementation - Optimization opportunities
  - **Type 2 Screens** (#11 areas)
    - Lookup parent entity first & have a single grid and form(s) for Create/Update/Deletion of a single entity.
    - Grid complexity ranges from Low to Medium (based on # columns)
    - CRUD complexity ranges from Low to Complex (4 fields to 20 fields)
    - Design/Implementation Approach
      - Create components to reuse the following
        - Modal to pick parent entity.
        - Read only Grid with a link to CRUD form(s)
        - Build support for multiple data loading endpoints/APIs for
          - Lookup parent entity grids.
          - Child entity grids.
        - Build support for CRUD operations based on entity types.

Analysis

- Design/Implementation - Optimization opportunities
  - **User Data Queries (reporting)** (#34 reports)
    - Eliminate filters where applicable. (use JQ grid filters)
    - Use common read-only grid component.
    - Common page with filters/without filters.
      - Impacts API complexity.
    - Use common filter components/lookups.
    - Design/Implementation Approach
      - Create components to reuse the following
        - Modal for filters.
        - Read only Grid with filters to load data based on grid metadata.

Analysis

- Design/Implementation - Optimization opportunities
  - All other pages/areas
    - Potential re-use of read-only grids.
    - Potential res-use of filter/lookup components

Option 2

- Estimated effort reduction approx. 25-30%
  - Focus on code reuse/automation.
  - Build project infrastructure first.
  - Break application into modules based on related functionality.
    - Deliver to product team as and when done with each module.
    - Do certification in parallel.
  - Build APIs along with UI (no mock data/stub APIs)
- Project Structure (proposed)
  - Establish a goal/end date for project.
  - Drive towards goal via milestones/phases.
    - Use bi-weekly iterations.
  - Do JIT planning (project plan) for next milestone/phase.
    - First milestone – Infrastructure implementation (UI/API/Security/MAC)
      - Includes re-usable UI components/widgets etc.
      - Implement one area per screens type with APIs
      - Approximate Duration: 6-8 weeks.
    - Next Milestones – Implement each module in succession.