

THE
Art & Science

OF DIGITAL PROJECT MANAGEMENT

CASE STUDY: AMERICAN EXPRESS

After many years of business as usual for the @Work Product, I was brought on to help manage the Design Program on a complete overhaul of the platform.



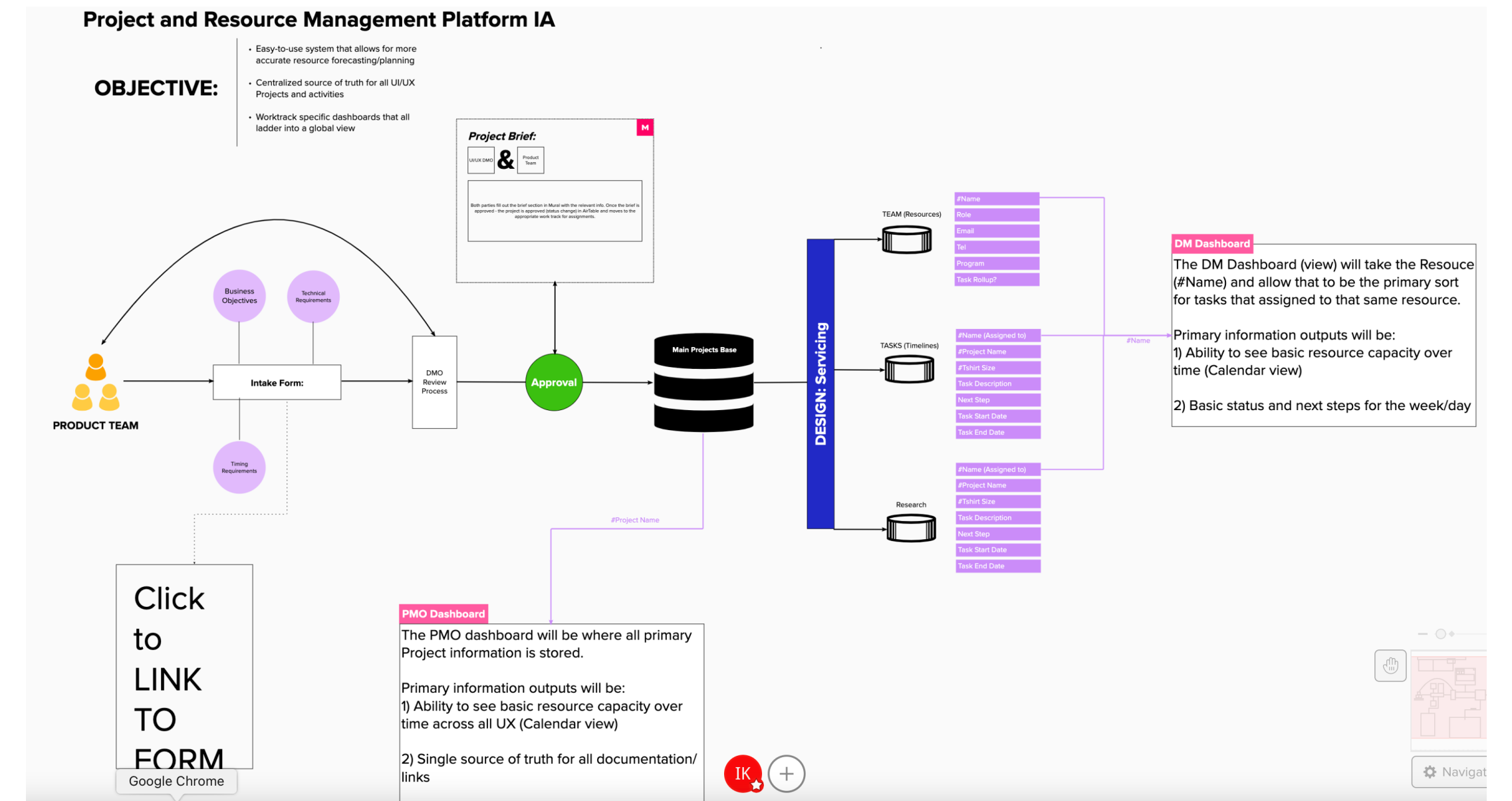
THE CHALLENGE

- 1) The @Work Design team consisted of about three UI/UX Designers and Project Managers who had spent the last 5 years simply maintaining the system via a couple of small projects per year.
- 2) Now all of the sudden there was both the intent and budget to update the entire platform, which required the team to expand rapidly, and get systems in place to support that growth.
- 3) Neither the Design or Product teams had any real system in place, and gotten by with some old templates in JIRA/Confluence. Any sort of resource planning was done in Mural (a white-boarding tool).
- 4) As a result there was no way to even accurately gauge the long term need(s), or subsequently make informed decisions around prioritization.
- 5) The product team was also new (and mostly comprised of SMEs with no formal product experience) and was struggling with articulating the program into something actionable.

THE SOLUTION

- 1) Introduced myself and setup time with the PMs, Designers, and Product Leaders in order to get as up to speed as quickly as possible on all the ways both communication and work happened
- 2) Outlined an Information Architecture map of the current state and identified gaps/areas where we could improve
- 3) Created a future state workflow in order to get buy-in from all the Directors

Built a custom dynamic Program/Resource Management solution (AirTable), including an Intake Form/Process that linked to our JIRA tickets/epics



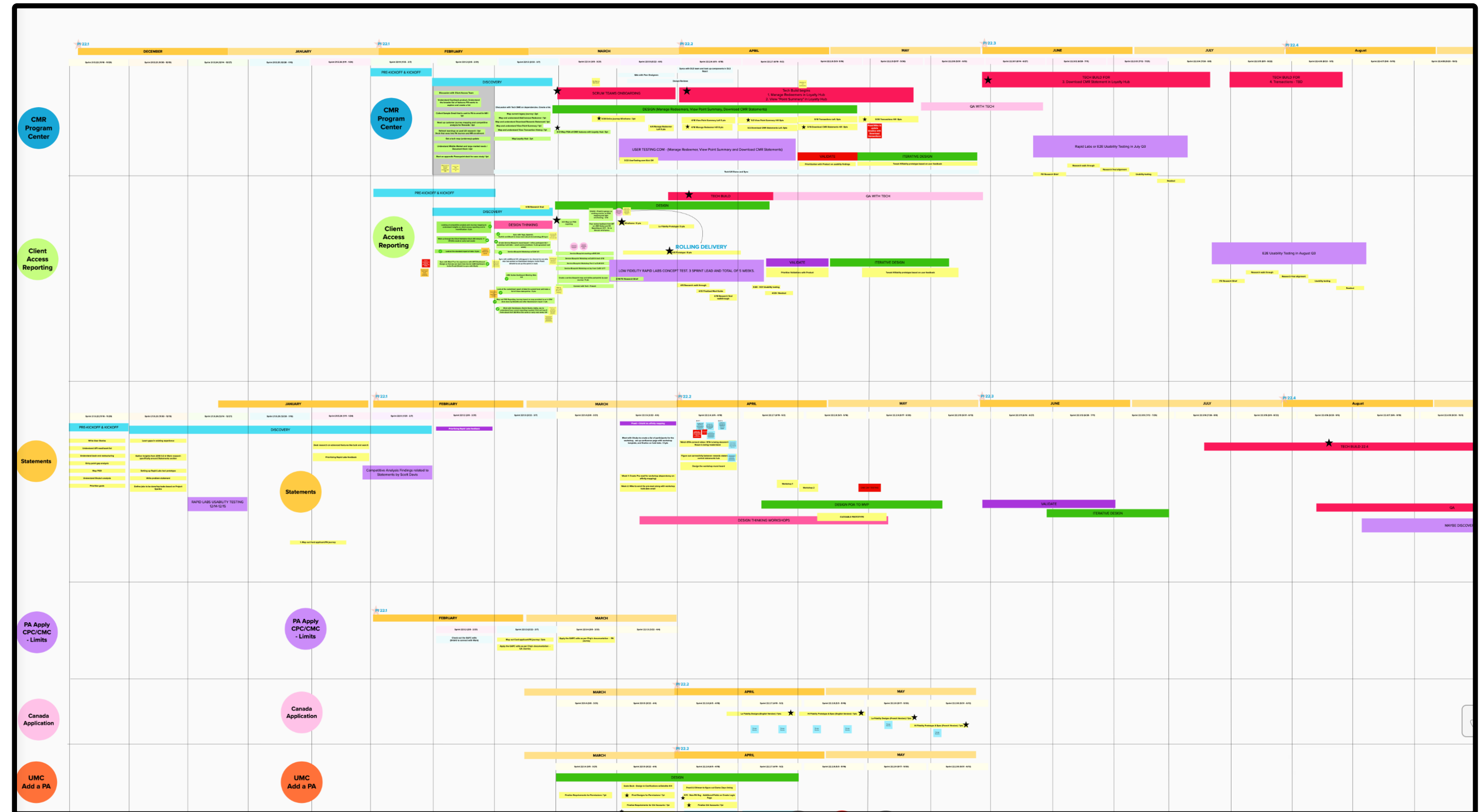
THE RESULT

- A custom/dynamic Program in AirTable that:
 - Connected all stakeholders across all functions
 - Increased (accurate) visibility into resource workload
 - Reduced the amount of handwork required to keep the system up to date
 - Significantly reduced churn in our overall process
- Performed an audit of the entire Design Team's meeting cadance to reduce/optimize time on our calendars
- The time we got back helped us focus on quality recruitment so that we were able to hire x5 designers, x1 UX researcher, and another PM in about 2 months
- The Design Program/Portfolio was so well organized and clear, that the PMO and Product Teams adopted our system as the single source of truth for the entire product roadmap

PORTFOLIO MANAGEMENT: BEFORE

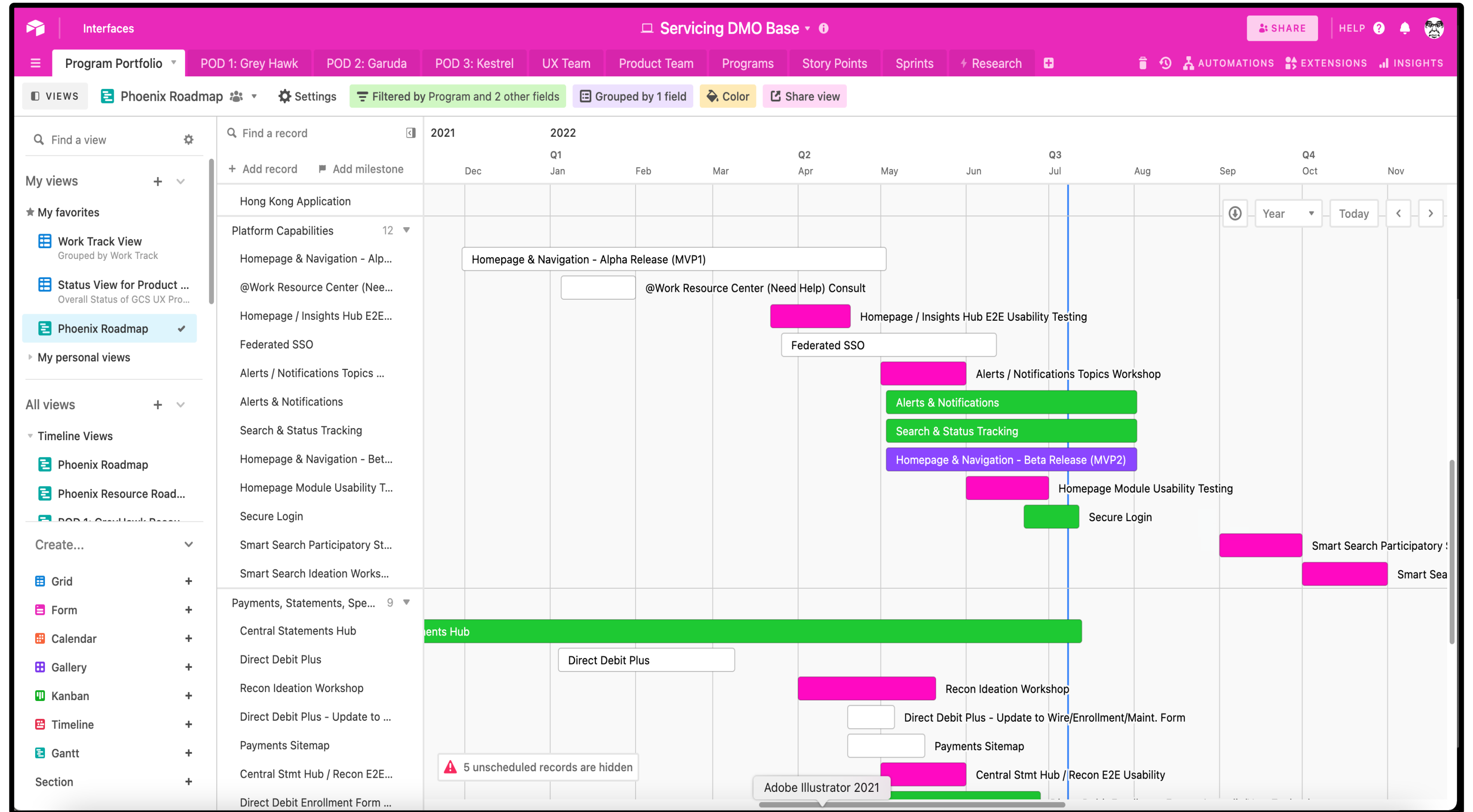
To left is how the team put together and tracked the Design program in Mural.

- Time consuming to maintain
- Static info made updates occur in a vacuum
- Difficult to read (or process information) and navigate



PORTFOLIO MANAGEMENT: AFTER

- Dynamic timing made updates easy and interdependent
- Able to organize info by stakeholder group/work track so that Product/Dev teams got only the info they needed, when they needed it, and didn't have to hunt for it
- Better organization made this way easier to understand and process relevant program/project information in order to make decisions



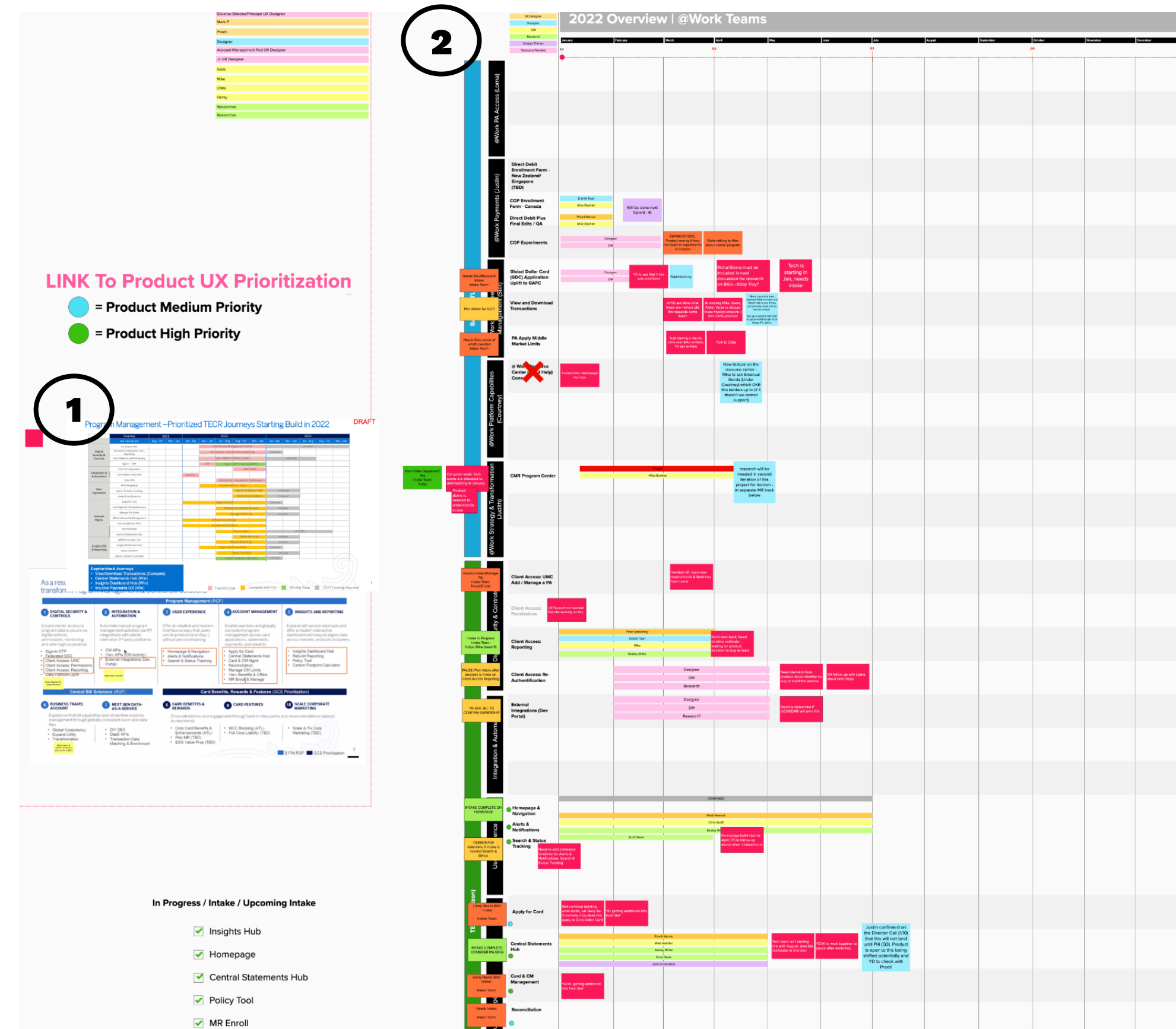
PROGRAM AND RESOURCE MANAGEMENT: BEFORE

To left is how the team approached prioritization/planning for the coming year, in order to identify resourcing needs in Mural.

- 1 Screen shot of the program from Product (done in PPT)
- 2 The Design Teams interpretation and breakout by resource

So there were two complicated visuals, both trying to say the same thing(s), while at the same time, each being time consuming to maintain (and technically redundant)

- Static info made updates occur in a vacuum. If the Product Team/PMO changed the roadmap on their PPT, they'd have to let us know... we'd have to manually update, etc.
- Difficult to read (or process information) and navigate



PROGRAM AND RESOURCE MANAGEMENT: AFTER

The changes resulting in the new system also vastly improved the way we tracked and managed resources.

- Dynamic/interconnected information made updates easy/dependent.
- Able to organize info by resource and by program. Design, Research, Product and Dev all had access to just the info they needed, when they needed it.
- Ultimately became the single source of truth for the entire @Work program, eliminating redundant systems. Everything was in one place and everybody was looking at the same thing.

