Shanshan County Turpan, Xinjiang, China
42.834284, 89.864778

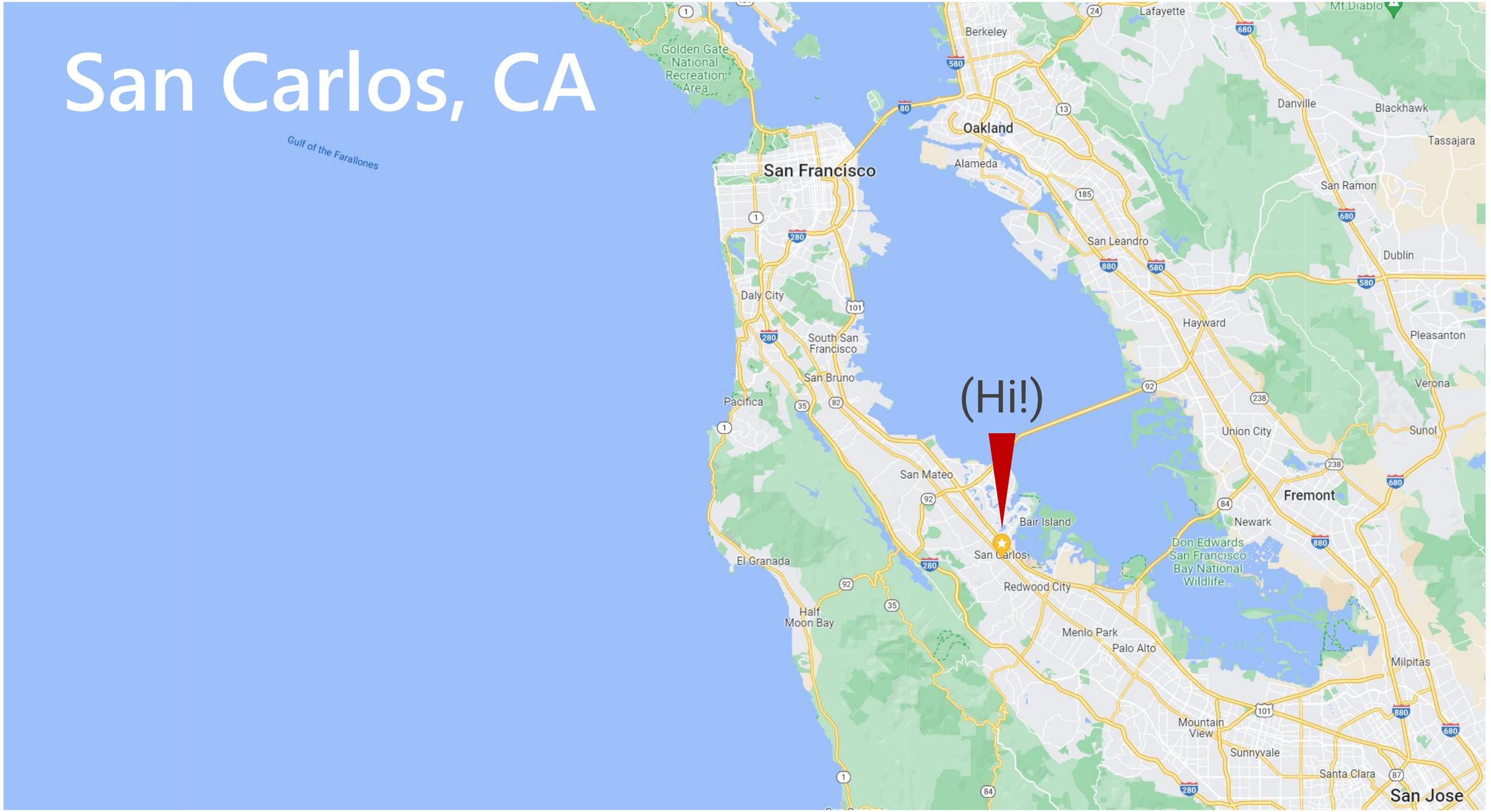
# Kevin Kell, M.A.

SF Bay Area, CA

Product / UX Designer - CX Certified

- Traveled around the world (26 countries)
- Coaches soccer for AYSO
- Volunteers for school activities (creative director of haunted house)
- Designed own house / designs and constructs backyard projects by hand
- Teaches kids about investing
- Studies soil science and permaculture

# San Carlos, CA



# About Me

- End-to-end, hands-on designer
- 20+ years experience as a UX Designer
- First job: User Researcher

## Domains

- AI/ML
- B2B Enterprise
- SaaS/Cloud Apps
- B2C Applications








# Journey Timeline

Masters of Psychology: Human Computer Interaction
Thesis published in Computers in Human Behavior

Founded a boutique design consultancy in San Francisco
(Clients: Intuit, Yahoo!, Intel Capital, DreamWorks)

Worked on large projects for Cisco and Apple

Started a UX practice at Badgeville

Redesigned legacy products and designed new SaaS/cloud products at Oracle

Lead the cloud platform UX effort

Served as Staff Product Designer at ServiceNow
(AI Search / Virtual Assistant / Platform / Portal)

Created a vision for Virtana's hybrid cloud management platform

# Highlights & Achievements

## Career achievements with impact.

Conducted over 40 user research studies as a user researcher

Lead designer for iPhone and iPad sales tool for VP of Global Sales

Redesigned the Learning site for Cisco.com, revenues grew from $150m to $400m post redesign

Redesigned Intuit.com’s homepage and navigation architecture

Helped redesign My Oracle Support, the primary tool for a $20 billion/yr. division

# Recent Projects

2020

servicenow

Workflow Management
Business Automation
Customer Service

**Staff Product Designer**

Platform UX

- Portal
- Virtual (AI) Agent
- Landing Pages / Dashboards
- AI Search

2022

virtana

AIOps
Performance Management
Cloud Cost Management

**UX Architect**

Core Design Team

- AI Cloud Platform Vision

# virtana

## Context

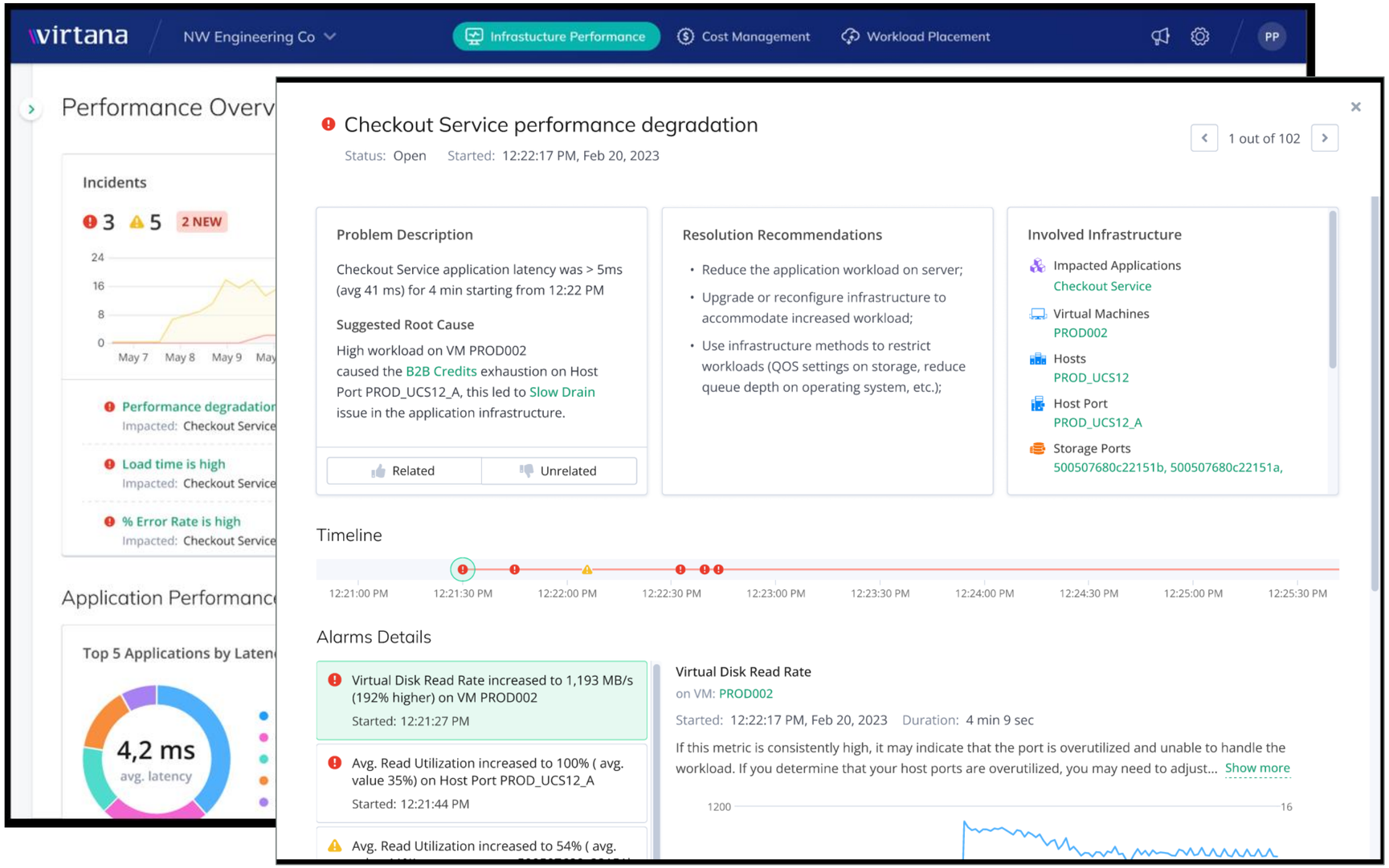
- I was hired as a UX architect to create a long-term vision for how the cloud platform would unify separate use cases around Infrastructure Performance management and Capacity Management

## Problem

- There was massive design debt since most of the On-premise software was done without formal design and was shown to be too complicated for all but the highest-level IT folks
- With AI becoming a viable technology, the risk that other companies would beat us to automated troubleshooting became increasingly likely

## Activities

- Conducted deep interviews with customers and industry users
- Created a vision that addressed the found pain points and a road map that detailed how we would get there
- Designed interactive Figma prototypes



servicenow

## Context

- ServiceNow needed a solution for an internal tool experience to monitor the outcomes of component performance testing in terms of page load performance times

## Problem

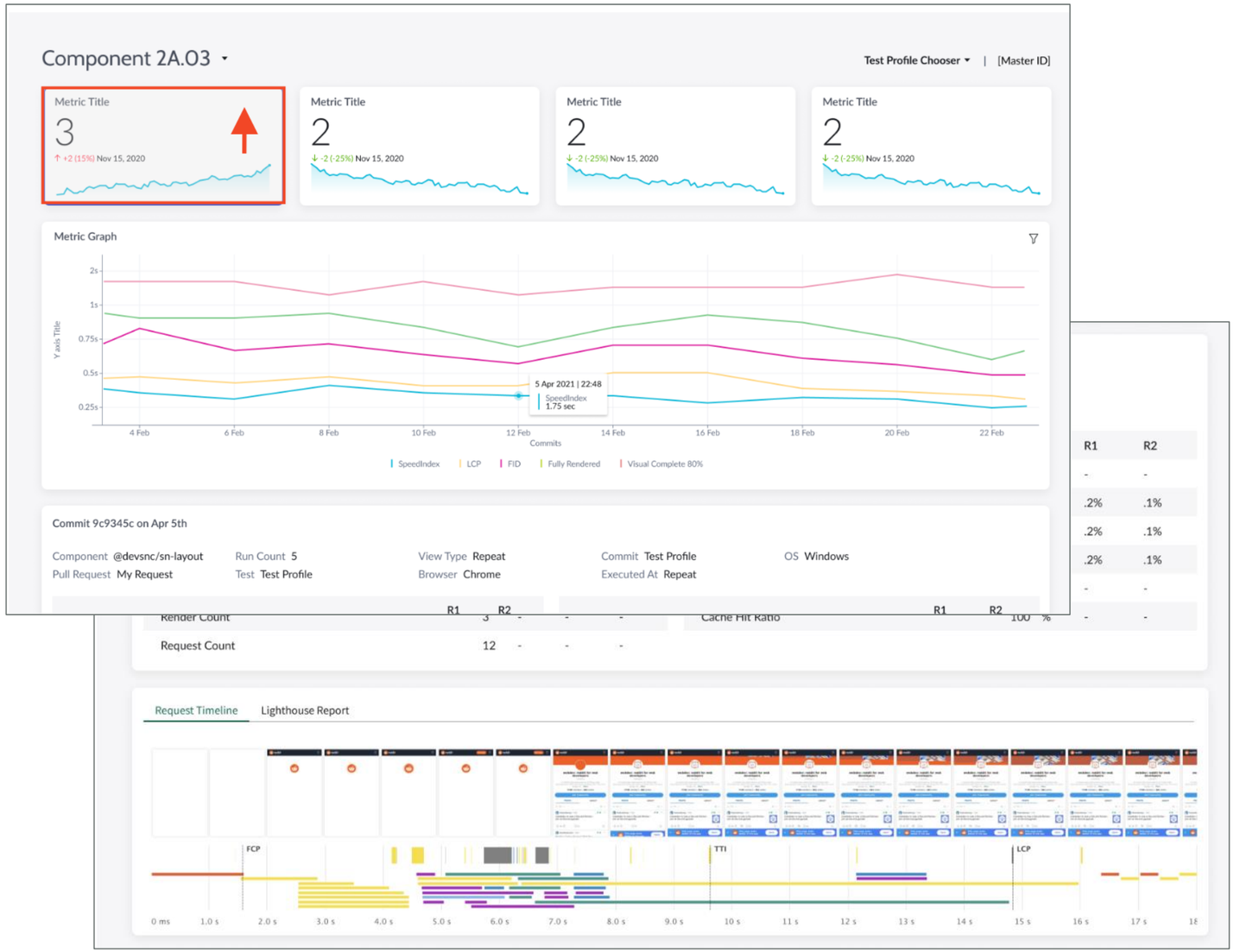
- There were many competing issues: how to onboard new users, show off the new features, and optimize and prioritize users' work

## Activities

- Worked with PM, users, engineers, and visual designers
- User research, concepting, wireframing
- Conducted research with PM, designed the initial concepts, and collaborated with visual designers on the final look and feel.

## Outcomes

- By innovating on the interaction model, I was able to create a tool that developers were far more productive with by centralizing all the information in one place, reducing page refreshes, and through the use a master-detail interaction model.



# Context

- Oracle did not have a Field Sales solution and many salespeople weren't using the Sales Cloud platform to manage their sales deals

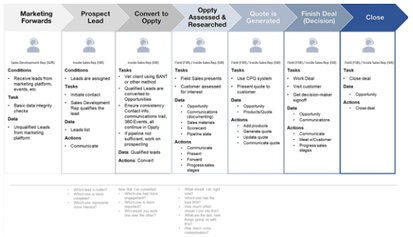
# Problem

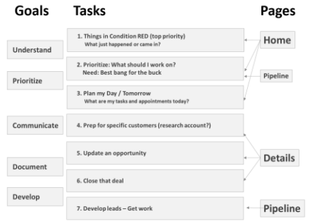
- Not enough was known about Field Sales' work to adequately build features and solutions that would make our sales force more productive
- The main competitors here was Outlook (email and calendaring) – it just did a competent job fulfilling users' needs

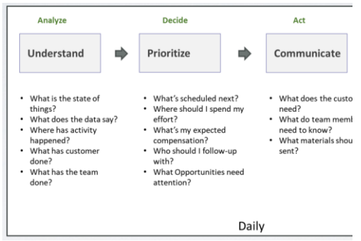
# Activities

- Worked with and interviewed the Field Sales team
- Analyzed the sales lifecycle
- Updated the Sales persona to include Field Sales
- Created an architecture that represented users' problems and goals. Then I deepened the model to include the tasks and data that users wanted to use.
- Create a page level architecture
- Designed proofs of concept to represent different ideas and alternative design possibilities

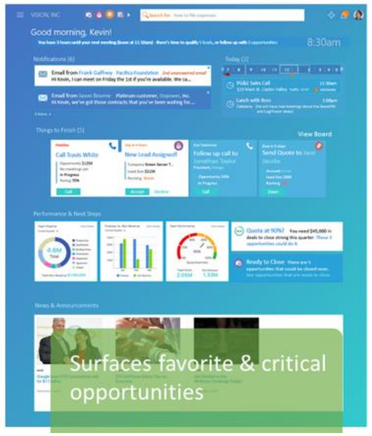
# Solution



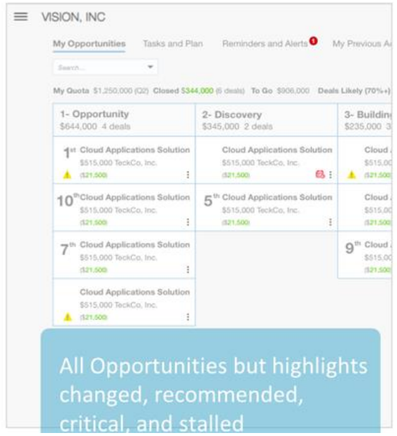



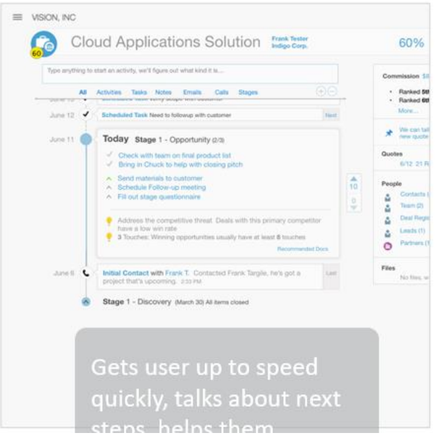

## Home Page




## Pipeline (Kanban)




## Detailed Opportunity



# Badgeville
The Behavior Platform™

## Context

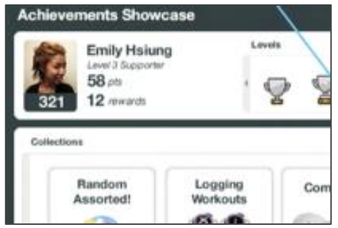
- I left Apple to start a UX practice at Badgeville (an enterprise gamification platform, the founder and CEO tasked me with leading a ‘radical revision’ to the usability of the current gamification platform.

## Problem

- V.1 version constructed with no professional design talent
- Little formal documentation of the customers’ pain points or needs existed

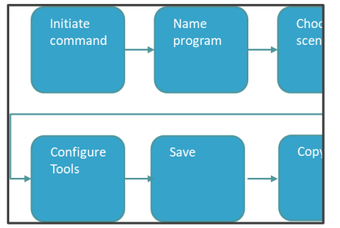
## Activities

- Worked with **Account Management**, **Product Management**, **Support** and even **Sales** to quickly understand who our users were
- UX methods: user research, task flows, wireframes, prototyping
- Worked with customers to collect pain points, tasks, and other inputs
- Hired and managed a visual designer
- Worked with PMs on new requirements and with Engineers to get the new designs implemented



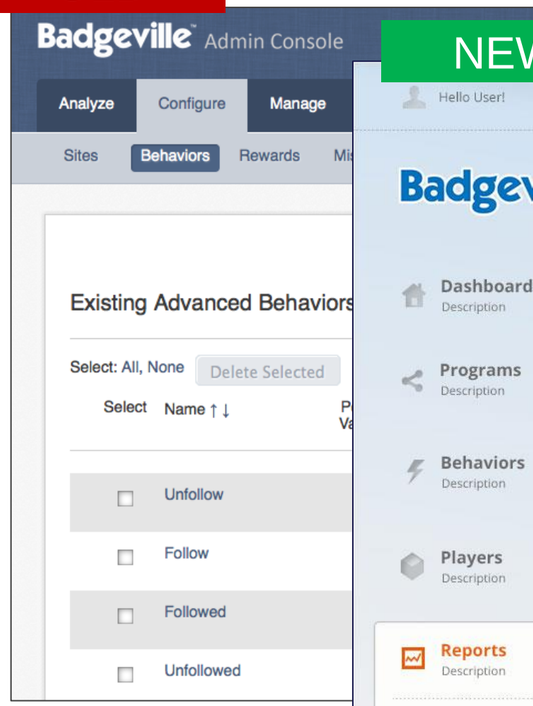

Understand the domain


Empathize with the user

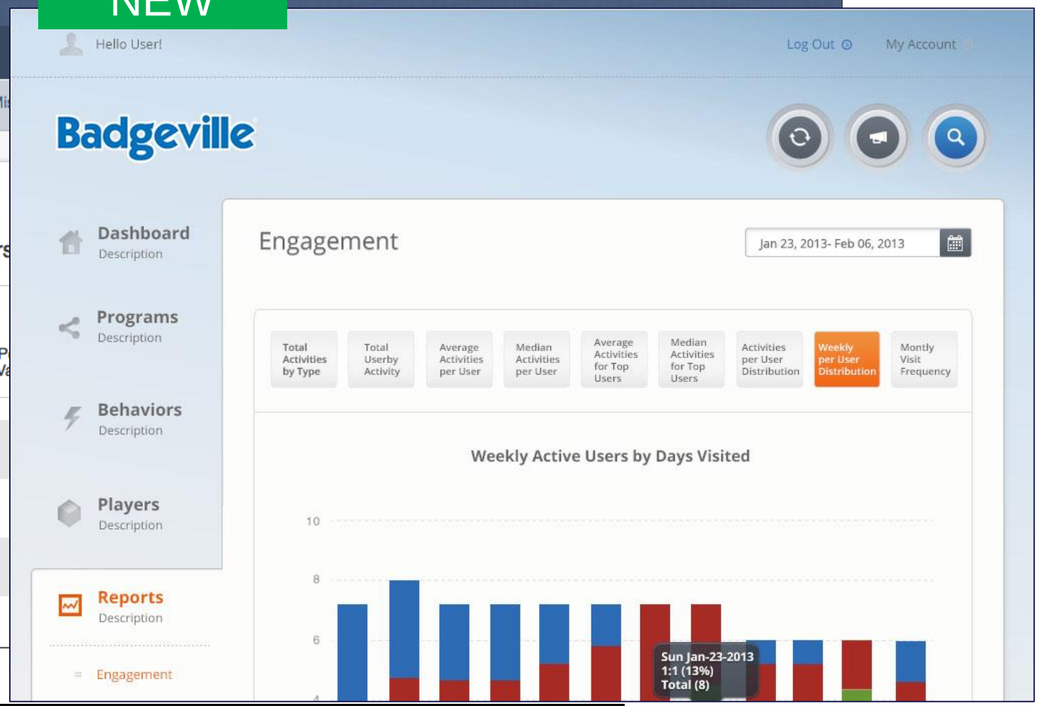



Map key user flows

OLD




NEW




**Customers and stakeholders were thrilled with the new version!**

# Apple, Inc.

## Context

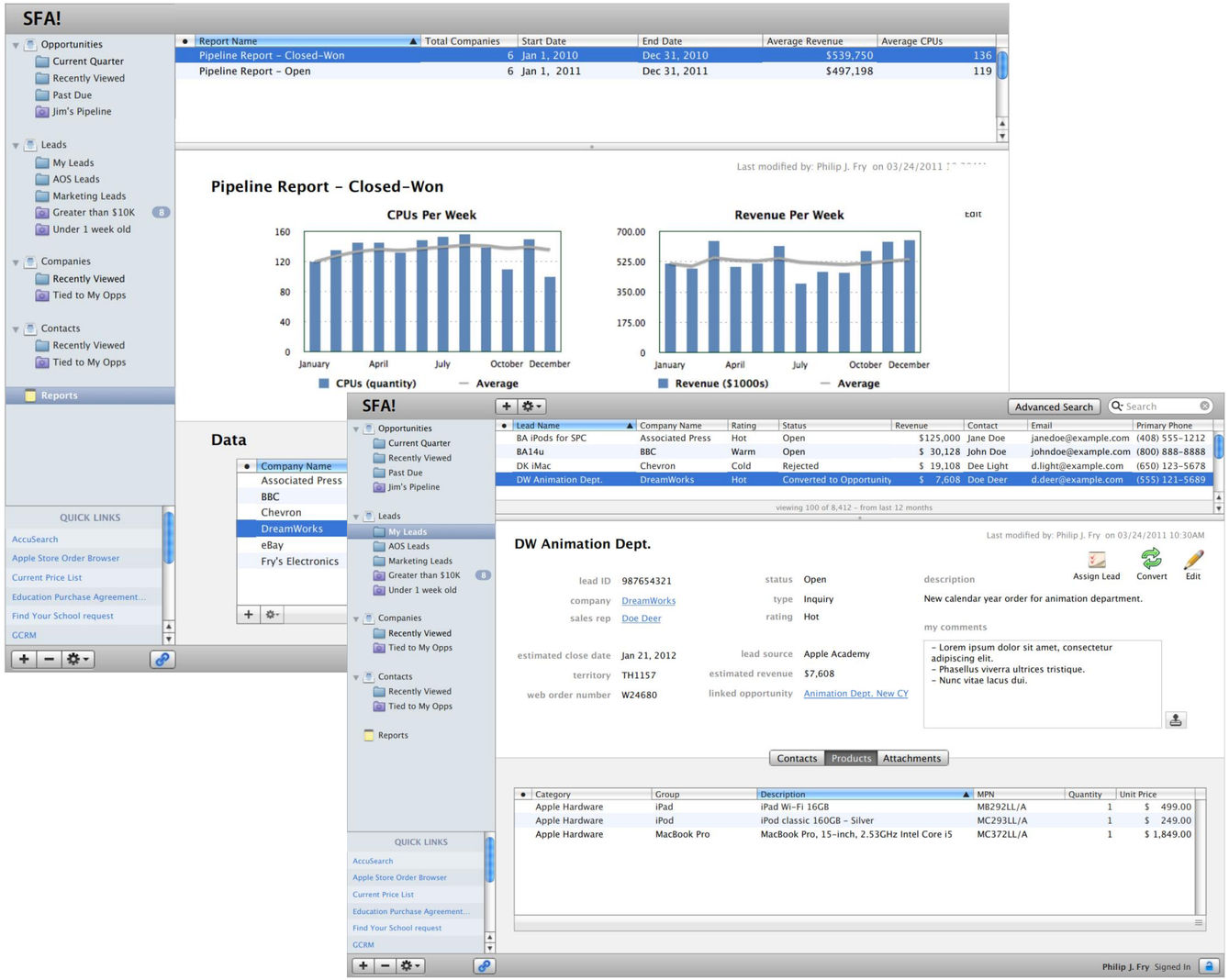
- At Apple I helped design a tool for the VP of iPhone/iPad sales and his sales force to manage large account sales and the overall sales lifecycle and process.

## Problem

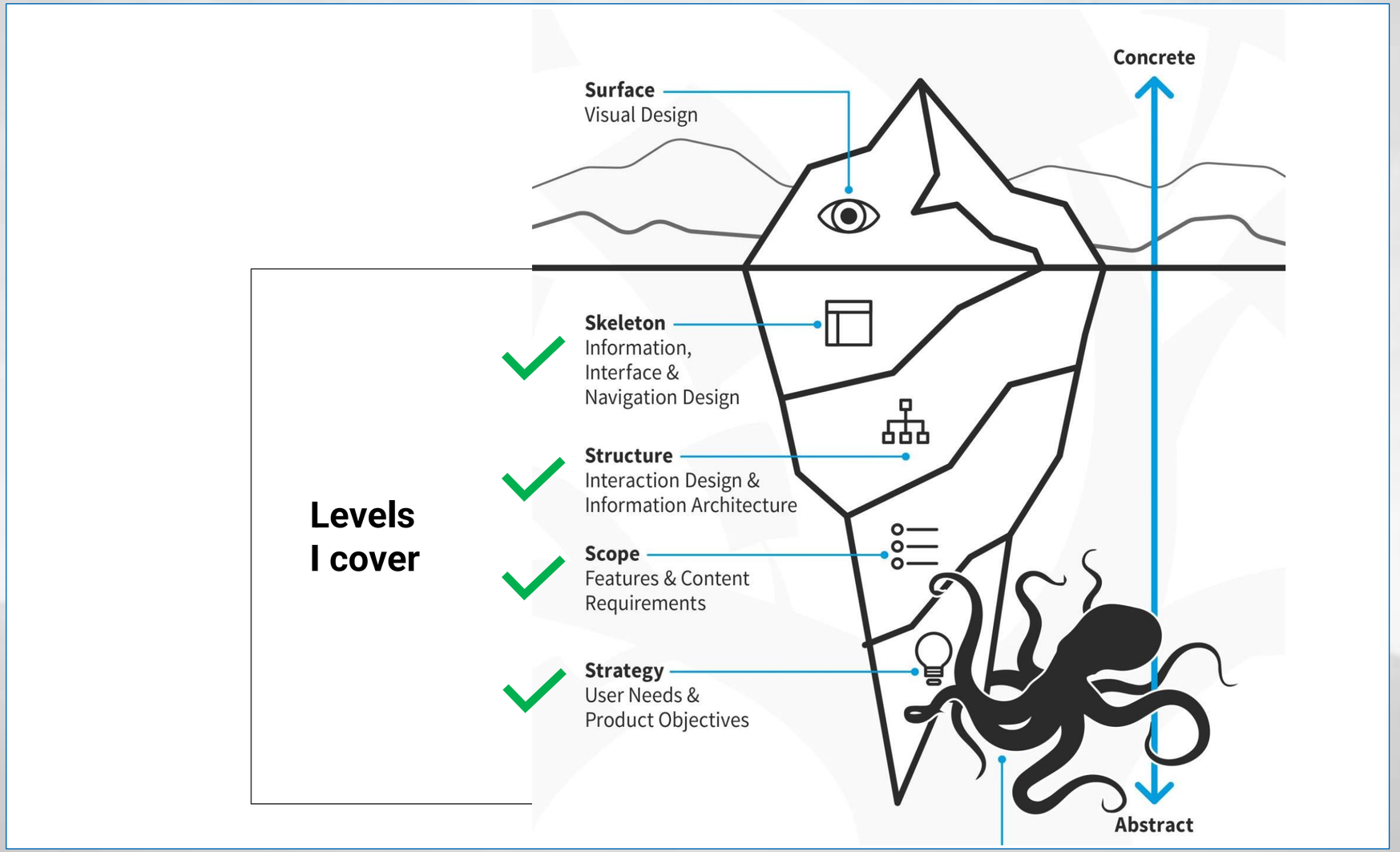
- Apple salespeople needed ways to handle common sales objects like Opportunities and Leads

## Activities

- Worked with users to understand their problems and needs
- Designed ways to handle opportunities, leads, companies, and contacts all in a single tool



# Where do I contribute?



# Best Career Lessons

Focus on user pain points and what value means to them

Don't build human intervention into the process

Good ideas can come from anywhere

Design fast and iterate quickly

# Questions I Like to Ask

**How is this going to be used?**

**What pain point does this address?**

**How do we provide value to the user?**

**What is our design center or north star?**

**How can we fulfill the company’s goals through design**

# What Differentiates Me?

20+ years in UX

**Research background** – academic and professional

Strengths in **data** and **problem** analysis

**Pragmatic** – get things done quickly and get feedback

**Range of experiences** in shipping complex consumer and enterprise products

# What Do I Like Doing the Most?

**Blue-sky designs from concept to implementation**

**Owning the user experience whether a feature, product or platform**

**Providing design leadership, innovative ideas and processes**

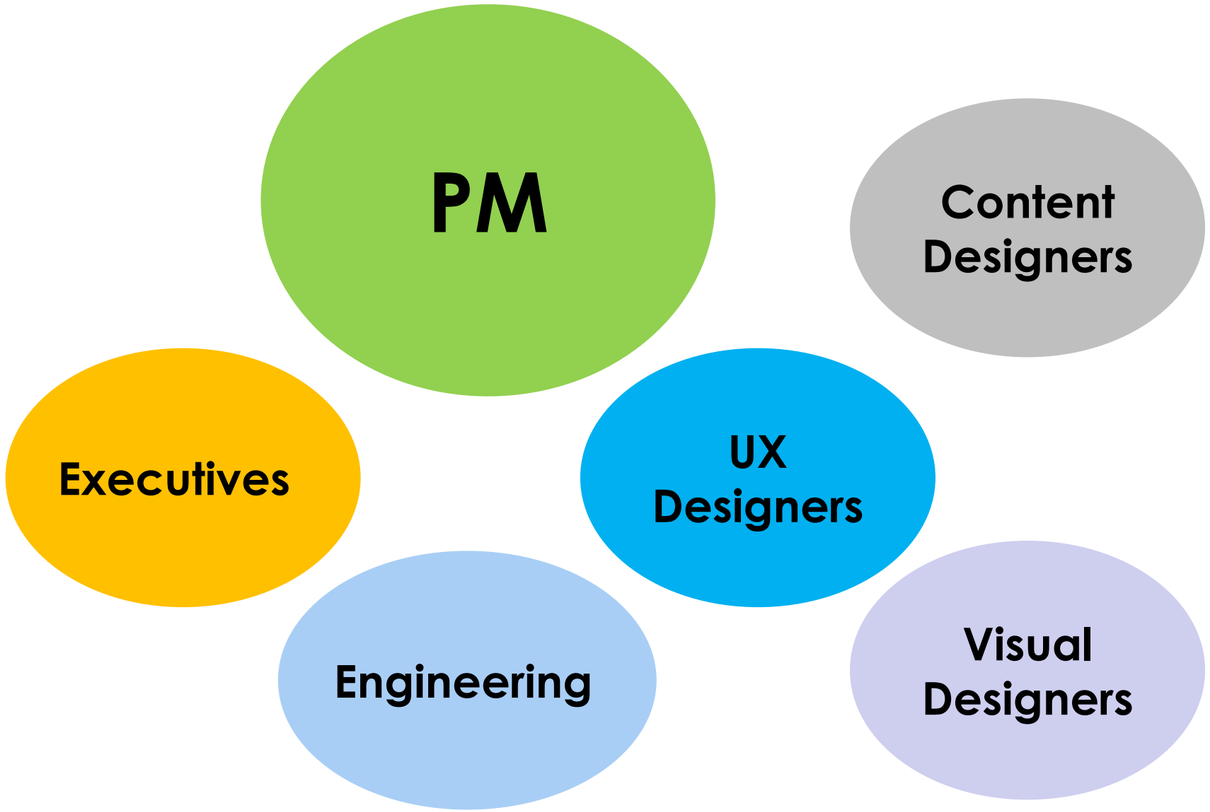
# Philosophy


Get quick design wins to build organizational credibility
Find what can be reused, made more consistent and systematized
Think consistently about value for the user

# Collaboration

## Who do I team with?

- PMs
- Engineering
- Executives
- Content designers
- Visual designers
- Other UX designers




### Collaboration Tools

Miro Figma JIRA Confluence

# My Typical Design flow

**PROBLEM** → **SOLUTION**


### Research

Understand the problem space (use cases), the personas, the business' goals, PM's goals. Work with customers and industry members.

### Design center

What's going to constitute an appropriate design? How will we measure it and know that we've reached our goals?

### Concepting

Now is the time to create alternate designs and spiral towards a single proposal, getting feedback and validation along the way.

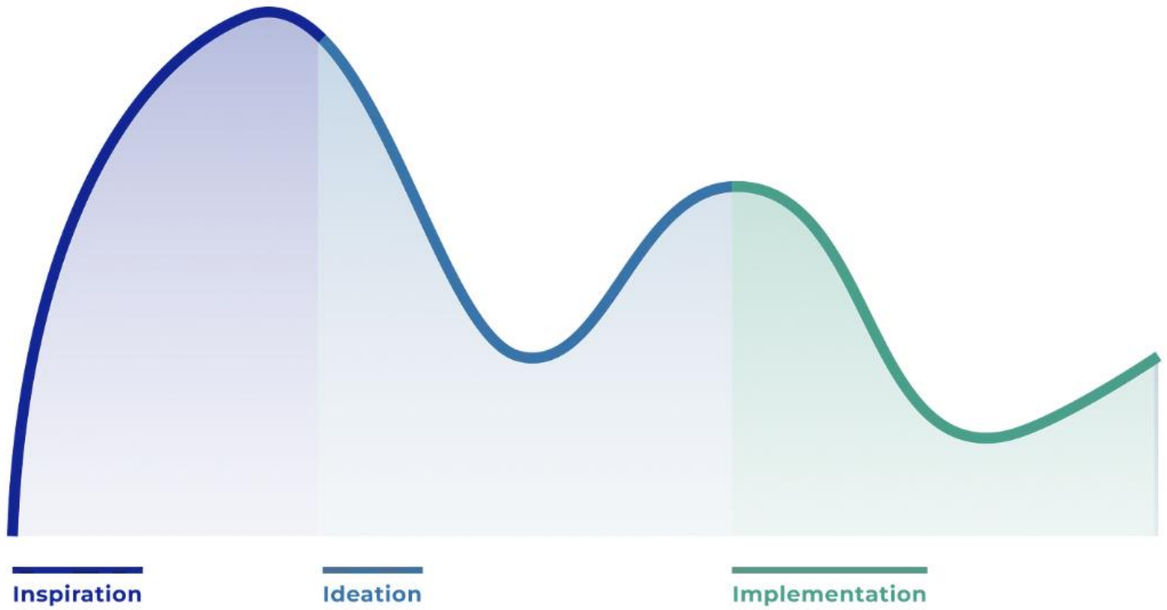
### Refinement

Understand implementability, reflect incoming feedback, iterate and work in the details. Select UI components and patterns.

### Handoff

Create specs, document the components, apply visual system, follow the implementation and validate the specs

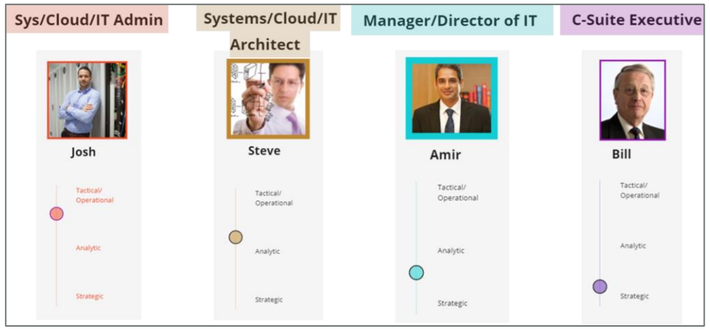
# Process – Steps to Bring a Design to Life



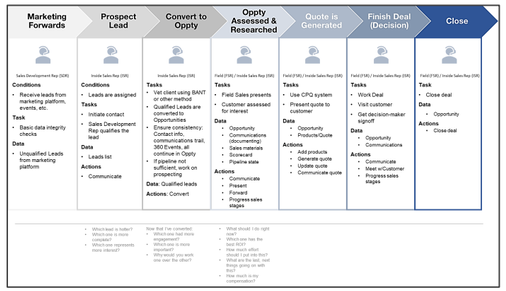

1. Research to understand your users and the competition
2. Build Personas, Scenarios, Journeys
3. Organize using Story Maps, Use Cases, and User Stories
4. Decide on design centers or north stars
5. Find inspiration
6. Start concepting and ideate
7. Create clickable wireframes
8. Research to get feedback
9. Iterate on your designs
10. Narrow focus down to a winning solution
11. Create a system that’s consistent and can be extended to other situations
12. Document it all in a spec
13. Work with engineers to validate against your spec

# Methods

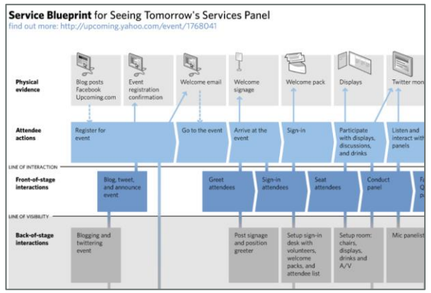
## Personas



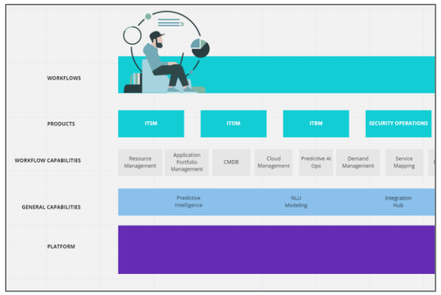

## User Journeys




## Service Blueprint

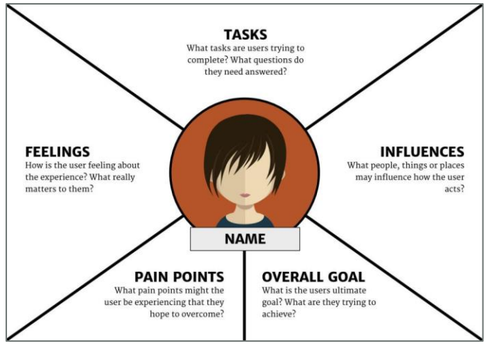



## Taxonomy/Mapping

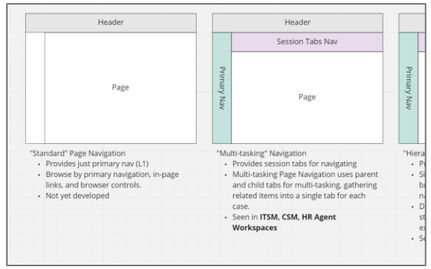



## Competitive Research

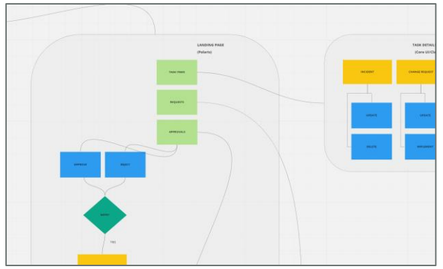

## Empathy Maps




## Navigational Models




## End-to-End Flows



# What is a Modern User Experience?

Based on patterns and user needs I identified, I formulated several major principles of what it means to have a modern user experience

## Principles of Modern UX

**Prescriptive - AI (Wisdom + Insights that are actionable)**

- Organize and Prioritize
- Wisdom not data
- Dynamic not Static
- Automation of simple tasks
- Reactive

**Info Meaning: Comparison & Baselining**

- Comparison
- Gamification
- Understand relative performance

**Convenient**

- Don't make me interact
- Minimize Navigation
- Visual UI
- Streamlined / Simple First
- Rapid scaling
- Social / Collaborative

**Knowledge: Contextual and Relevant Content**

- Recency / Frecency
- Curated - Role-specific or showing users where activity is
- Display choice
- Knowing where user in their journey
- Role-driven
- User-driven Recommended Content

**Personalized & Customized Experiences**

- Personalized & Customized Experiences
- Customization = Flexibility
- Resume: Pick up where you left off
- Use data to personalize
- Remember how you were using something

# What is a Modern User Experience?

## Concepts to Drive Productivity



### Contextual Knowledge

- Raise awareness in the user of their own actions, of what others are currently doing, and also of what the system is doing outside of automation
- Show present users, their status, and their actions
- Allow real-time collaboration



### Personalization

- UX fits the way I want to work, either automatically or through my manual intervention
- I want to tell the system what notifications, what tickets, what data are important



### Prescriptive AI

- A prescriptive approach that relies on intelligence and recognition to convey insights to the user.
- Accomplished by creating smart algorithms means that the system reacts, has a POV, and is actively assisting the user.

**Concepts map neatly to the 3 important questions:**

**What's Happened?**
- History (Me)
- Activity (Me + Others)

**What's Happening?**
- Current Activity
- Personalized Alerts

**What's Next?**
- Recommendations
- Prioritization

I believe these three concepts enable more productive user experiences

# Tools

All day.

Every day.

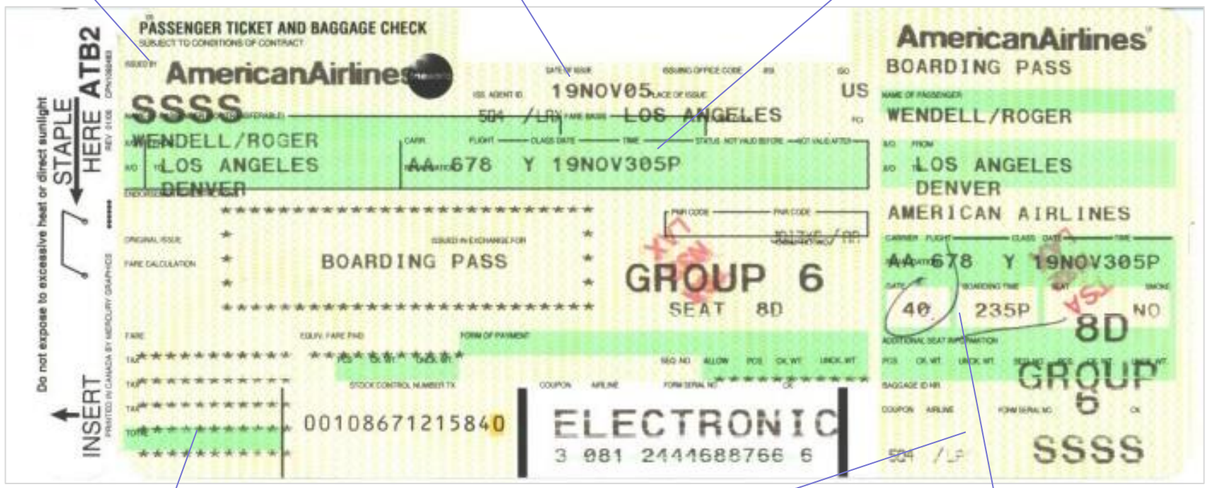
# How I'd Redesign an Airline Ticket in 10 minutes

Airline name and initials repeated 5x

It's barely possible to find the important information on this

Time and date is in an unfamiliar and hard to read format

**Remember the old ticket formats? Let's fix this!**




Information relevancy actually shifts depending on when and where you are

Information hierarchy is not effective and inconsistent

You can barely find the gate and departure time

## To create a great design, start with the questions that users have:

1. What is my **destination**?
2. What is my **gate**?
3. What is my **seat**?
4. What **boarding group** am I in?
5. What **airline** am I flying?
6. What is my **departure time**?
7. What's my **flight number**?
8. Is this **my ticket** and not another family member's?

# The Information Hierarchy is Key


## Next, create an information hierarchy:

1. **Preboarding**
   a) My Name
   b) Airline
      1. Flight Number
   c) Destination City
      1. Origination City
   d) Departure Time and Date

Need to know first, then second, third, and so on.

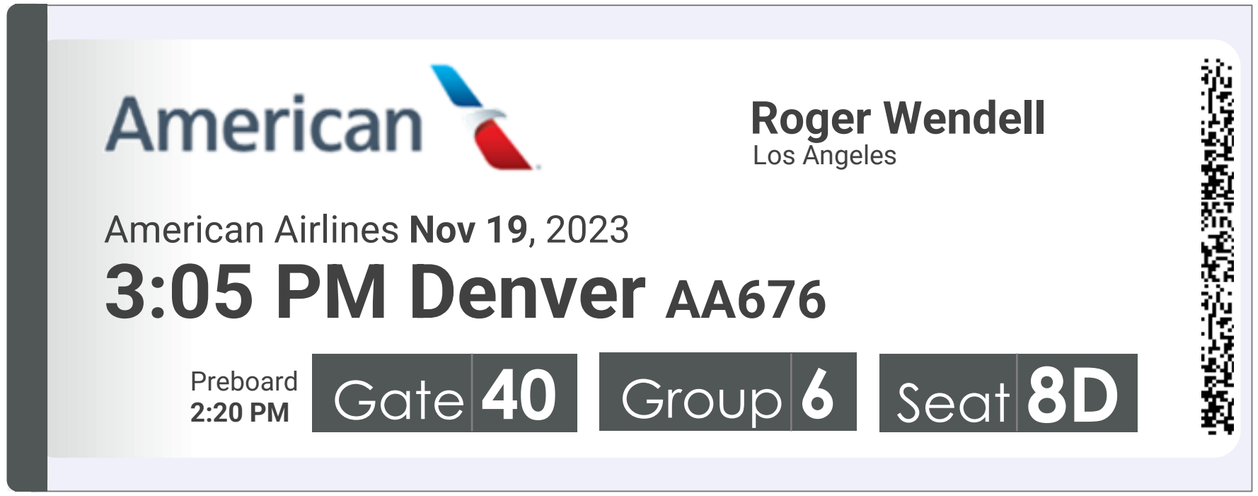
2. **Boarding**
   a) Gate
   b) Boarding Group
   c) Seat

Get me to the gate, then tell me how to board and where to sit.

# A 10-minute Design is Born

**Highlight the most important information by building an information hierarchy using visual design cues**


**Operations First Design**




**User First Design**

**It's the exact same information minus extraneous text**

# Case Studies

## New AI Experience Vision for Infrastructure Performance  Management

Helping an enterprise platform become AI-driven

## Component Testing Console

A tool for internal developers to improve UI component performance

## Field Sales Proof of Concept

for Sales Cloud

## Badgeville Platform Redesign

Enterprise gamification platform


## Project and Goal

- When I was hired as a UX Architect, Virtana had a pressing need to start to simplify their product offerings and have an integrated experience model that would unite two strategic areas: Infrastructure Performance Management and Capacity Management.
- AI has raised new possibilities for helping with problem diagnoses, interpretation, and solution recommendations for infrastructure problems.


New AI Experience Vision for Infrastructure Performance Management
for Virtana Platform

Helping an enterprise platform become AI-driven

## Challenges

- Virtana had been struggling to sell on-premise software and needed to quickly transition to a cloud-based web platform to cover infrastructure troubleshooting, monitoring, and capacity management use cases.
- With AI (ChatGPT) a viable option for data interpretation and explanation, I made that a central part of my vision for the new version of the Virtana Platform.
- Performance PM: *“Our customers don’t understand their own data, let alone the performance problems that they have”*

New AI Experience Vision for Infrastructure Performance Management
for Virtana Platform

Helping an enterprise platform become AI-driven

## Activities

- Worked with experienced users and customers who are domain experts to collect pain points, personas, use cases, and new ideas through interviews, design sprints, and card sorts
- Teamed with PMs, Professional Services, current customers, and other designers
  - *“Reduce the data shown to users”*
- Presented vision to key stakeholders to get buy-in
- User research, data analysis, concepting, wireframing and Figma prototypes


New AI Experience Vision for Infrastructure Performance  Management
for Virtana Platform

Helping an enterprise platform become AI-driven

## Outcomes

- Getting quick wins was essential, so focusing on specific areas (performance and capacity management) and producing designs early helped demonstrate the feasibility of my vision
- Produced a prototype (infrastructure) of the vision that centered around a **troubleshooting** use case which demonstrated the power of an **AI-based** experience


New AI Experience Vision for Infrastructure Performance  Management
for Virtana Platform

Helping an enterprise platform become AI-driven

## Process

- Understand where we’re at with the **product**
- Fully document the business’ **goals** and **strategy**
- **Understand the user** as well as possible from many angles
- Create **themes** that describe what the user is trying to do
- Author a **model** of the relevant concepts (objects, tasks, data) interaction model
- Translate all that into a coherent interaction model
- Produce artifacts that allow for gathering **feedback** and eventually **specifications**

# METHODOLOGY

**Research**

- Methodology
- Pain points
- Analysis

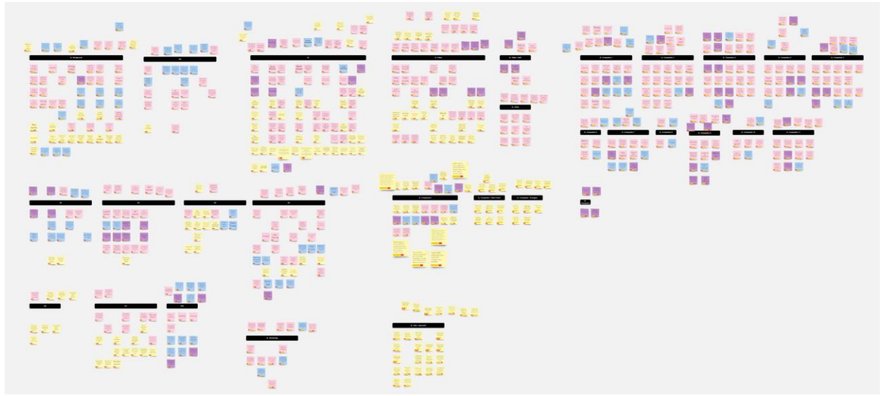
Goals & strategy

Artifacts

Design

Outcomes

- Talked to 9 IPM professionals
- Asked users questions about pain points, what their needs were, and showed competitor screens
- 560 unique data points (Miro sticky notes)
- Tagged all 560 notes by interviewee and question
- Clustered the data notes into study question groupings
- Assembled the notes into conceptual categories
- Created an analysis framework – Persona – Product – Principal Themes – Design – Features
- Summarized Findings
- Articulated major themes



# PAIN POINTS

**Research**

- Methodology
- Pain points
- Analysis

Goals & strategy

Artifacts

Design

Outcomes

**Users get too many alarms, in addition to being overloaded with data**

- It’s not clear who owns what error source

**The big picture is hard to get**

- Platforms don’t effectively support creating a clear mental model of users’ infrastructure
- Users don’t want to find problems, they should find users – they want the product to be proactive even before problems arise
- It’s hard to pull together log file information into a coherent picture
- Users like to think in terms of ‘what’s connected to what’

**Poor UX design is a reality for users of IPM platforms**

- UI’s are too complicated, require training, or are hard to share with other teams and personas
- Graphs or visuals often don’t add value to operational problem solving, but are more suited for executives
- Most visual or page designs are not clearly solving a problem
- Users want to go to related objects but frequently cannot get there easily

**Users don’t want to be in reactive mode continuously**

**Data is generally poorly represented, even in cloud platforms**

- Millions of data points are generated a day; there is too much going on and too much to keep track of
- A lot of work goes into filtering down data to an addressable set of problems

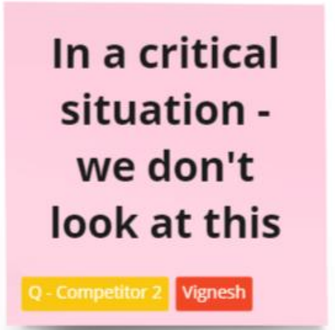
**Hard to correlate infrastructure changes with performance issues**

# ANALYSIS



## 1. UI for all - simplicity and learnability

- Design of IPM tools are a problem
- Complexity / Information Overload
- Customers are drowning in information
- **Heat map visualizations** are noisy and not useful



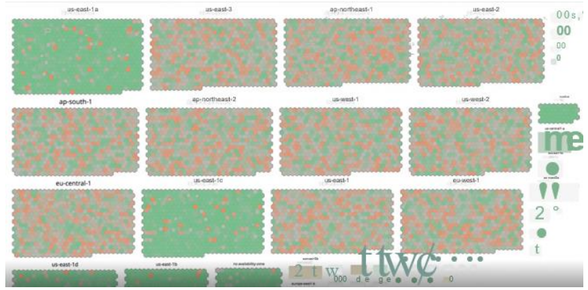


## 2. Users want solutions not data

- Actionable insights + Manage by exception
- UI must be task-oriented and focused on the problems at hand
- Users think in terms of **applications** not individual pieces of equipment, metrics, or specific problems


## 3. Holistic / agnostic platform

- Show me the connections and the causal chain
- Contextualizing problems with recent changes
- We make it very hard to connect the dots and we don’t align with customer’s mental model of their environment



# ANALYSIS - CLOSEUP

**Research**

- Methodology
- Pain points
- Analysis

Goals & strategy

Artifacts

Design

Outcomes

## #2: Users want **solutions not data**

- Multiple users (the most articulate ones) voiced that they cared about solutions not the data

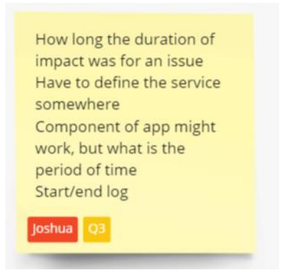
**Solve my problem**

- Users care about their services, business processes, applications, or critical endpoints

**Show me the things I care about**

- Users want a product that understands their wants, needs, and problems and helps them solve issues quickly in a focused way.

**Get me answers**





# GOALS & STRATEGY

Research

**Goals & strategy**

Artifacts

Design

Outcomes

- **Capacity and Performance were silos.** We needed to unite the experience
  - **Application centric:** Infrastructure exists to support applications. Infrastructure issues are most relevant in the context of application performance.
- Surface errors in a '**manage by exception**' paradigm
- Create a simpler, more learnable UI **for less technical personas** and other groups and vendors to drive adoption
  - Make the product more **organizationally aware** of who owns what
- AI-driven: Leverage ChatGPT to provide **natural language** diagnoses, recommendations, and concept explanations

# GOALS & STRATEGY

Research

**Goals & strategy**

Artifacts

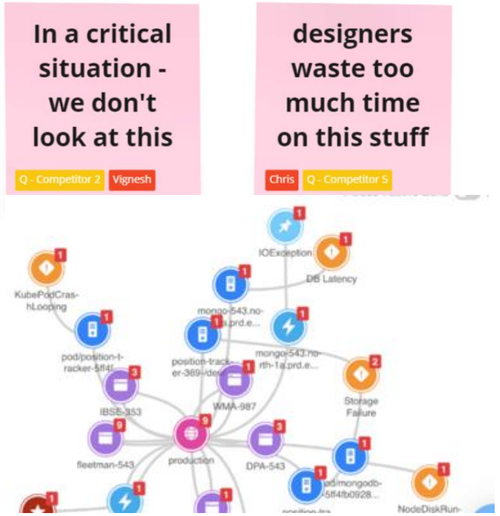
Design

Outcomes

## What Users Really Want

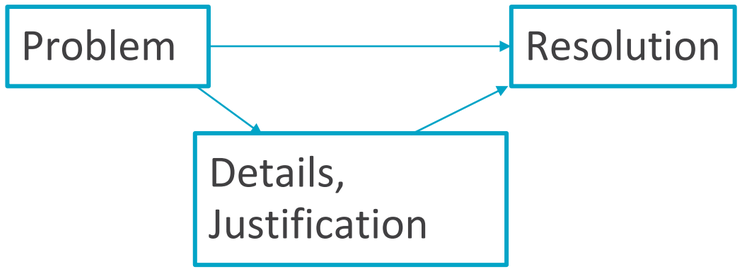
***"Solve my problem in 2-3 clicks"***

***"Tell me what the problem is and how to fix it"***




**Users want solution-oriented experiences**

## The Model



# PERSONAS

- Research
- Goals & strategy
- **Artifacts**
  - Personas
    - Scenarios
    - Use cases & Taskflows
    - Card Sort
    - Storymap
    - Architecture
- Design
- Outcomes

Instead of many personas or many permutations, I simplified the user set down to four canonical positions that would touch the use cases in varying ways

**Sys/Cloud/IT Admin**

**Josh**


**Systems/Cloud/IT Architect**

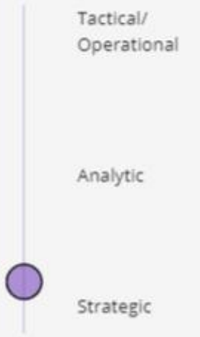
**Steve**


**Manager/Director of IT**

**Amir**


**C-Suite Executive**

**Bill**



# SCENARIOS



Deepening the personas with realistic scenarios helps you understand and evaluate the context of usage.

**Josh**

**Scenario: Troubleshooting**

- Josh the admin gets a slack message that a particular device is triggering an alarm. He clicks on the link in the message and opens an airy, clean, and simple platform page that details the alarm (the trigger, the conditions that triggered it) and the performance metrics for that device. There's a recommendation that he adjust the configuration of the device to allow more bandwidth. After looking at the visuals, he decides to adjust the device's configuration by clicking on the link to the administration page for that device.

**Signature Moment**

- Direct access to the detailed alarm and out of band data
- Recommendations on the immediate course of action
- Visuals that support the recommendation

**Steve**

**Scenario: Complex Troubleshooting**

- Steve the architect gets a more complex alarm notification. It seems that several devices that are connected to each other are in an alarm state. Steve also goes to a case page that shows all of the affected devices that are related. There isn't a specific fix recommended but he is advised to look upstream. He browses the topology of the connected devices to see the one that is most upstream. He sees that the device's **change history** is highlighted for the most recent changes. Since there was a recent change he drills into the change and realizes that the adjustment wasn't the right thing to do (thanks Josh) so Steve clicks on the access link in Platform to go to the device's configuration page and fix things.

Support reverting the change

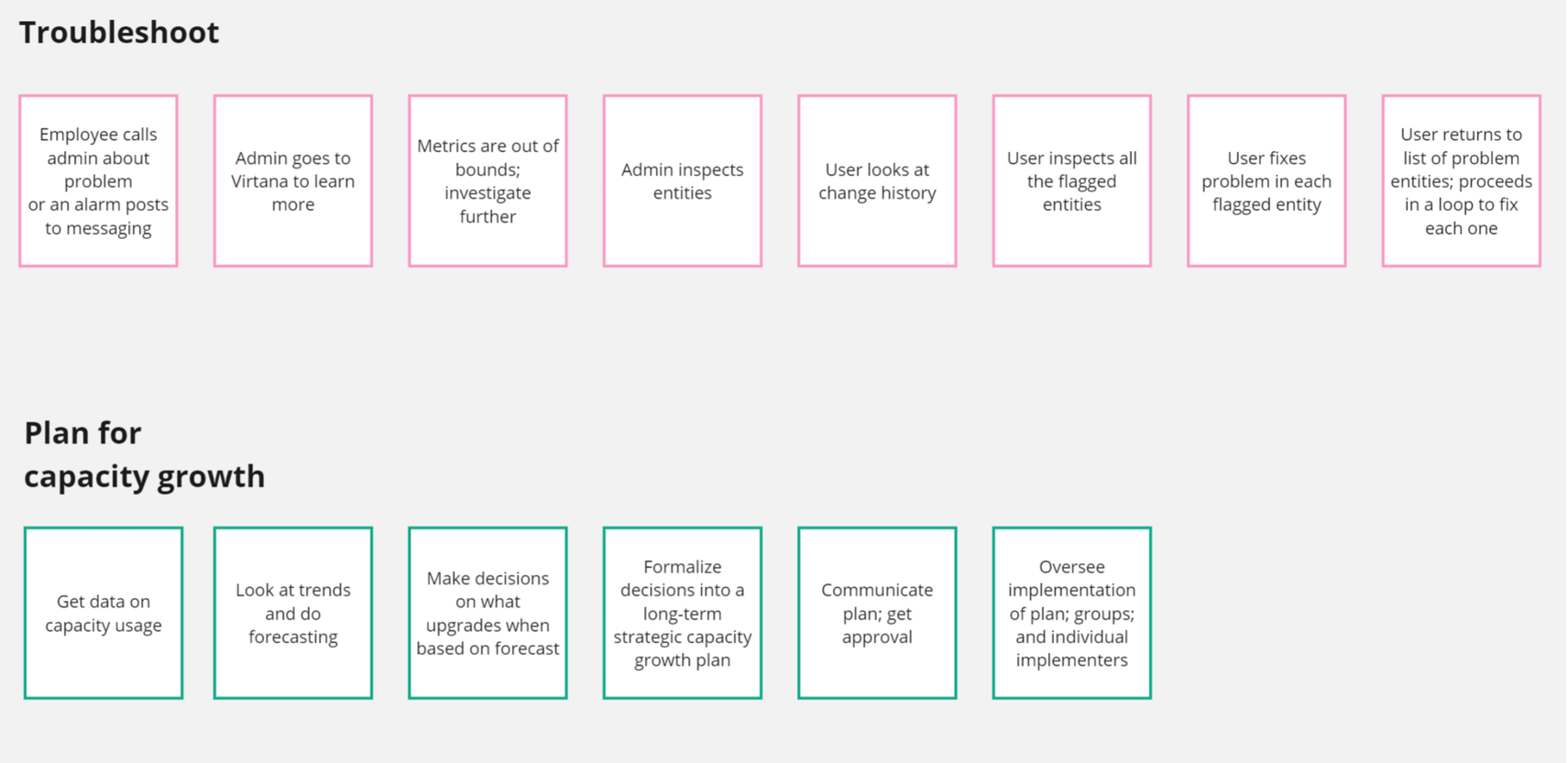
**Signature Moment**

- Multiple-entity alarm notifications
- Change history shows why the performance problem happened
- In-line access to device configuration pages

# USE CASES AND TASKFLOWS

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes

I also deepened the use cases with taskflows that we planned to support in the design.

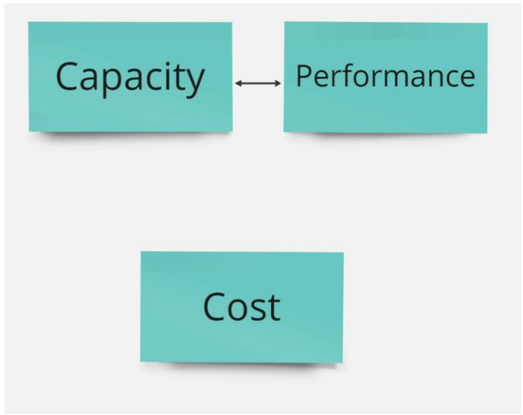


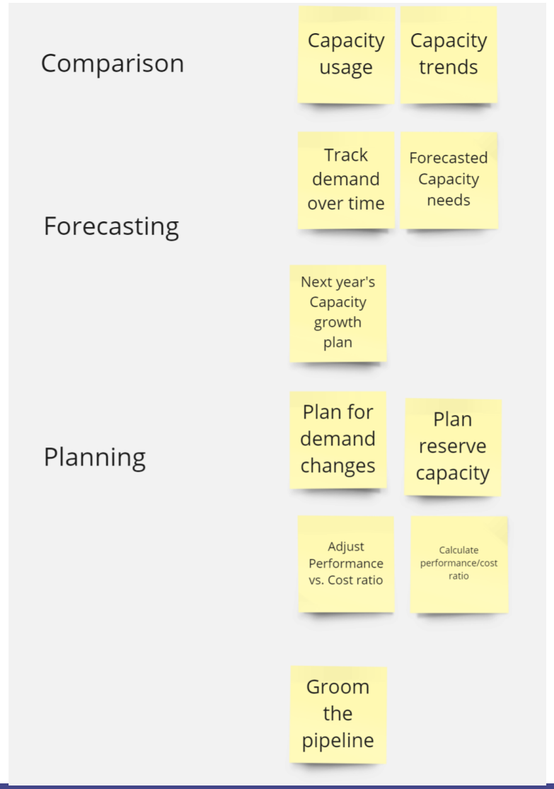
# CARD SORT

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes

I asked users to arrange cards detailing capacity concepts and cost into a flow.
An interesting outcome was that users showed consistent agreement on clusters that the cards fell into.

Users also indicated that capacity was more related to performance.

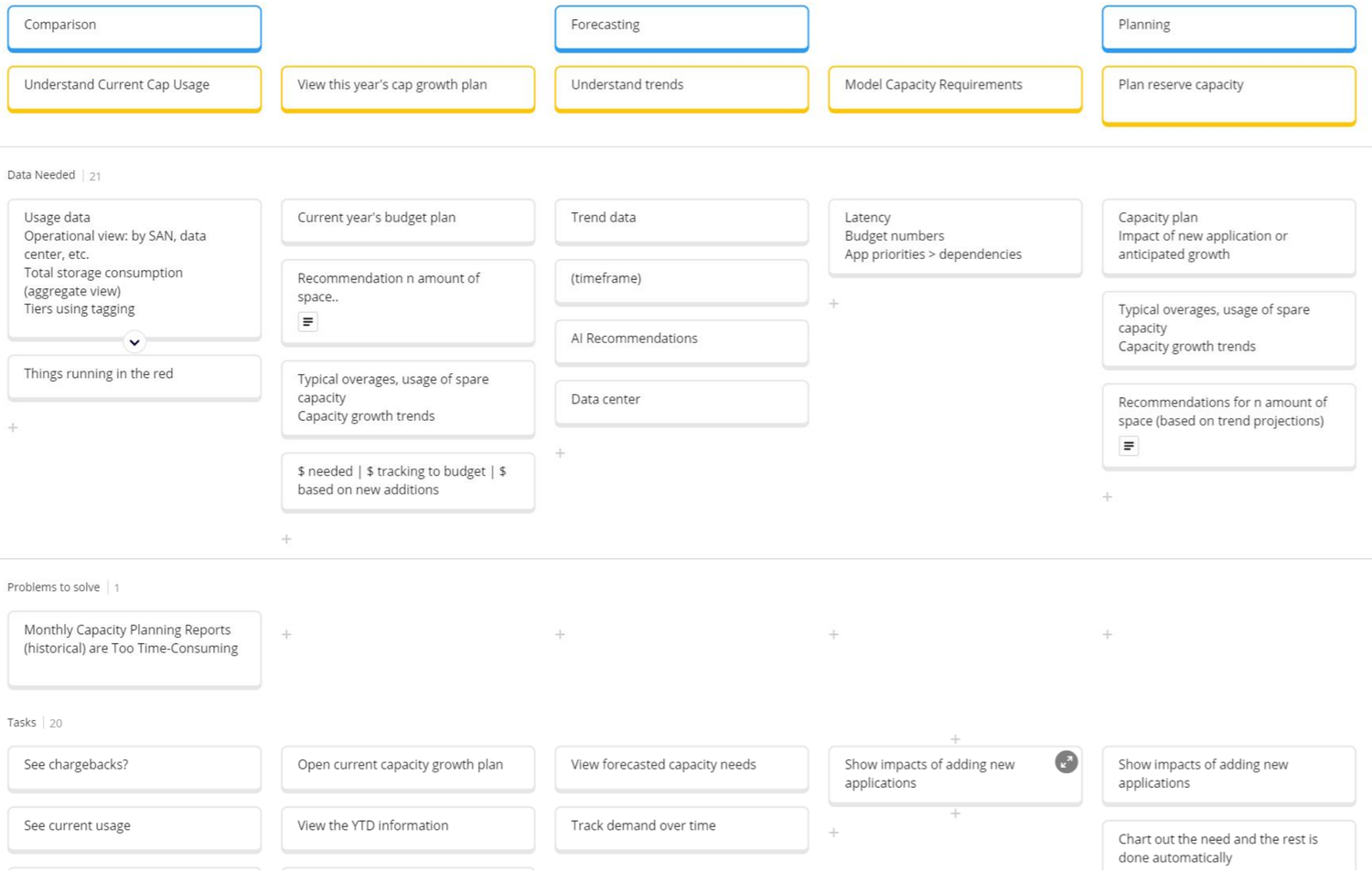


# STORY MAPs

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes

Story maps show how the entire user model fit together.



# ARCHITECTURE GOALS

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes

## Goal 1

Surface performance errors to the top of the interface

## Goal 2

Make the UX application-centric

## Goal 3

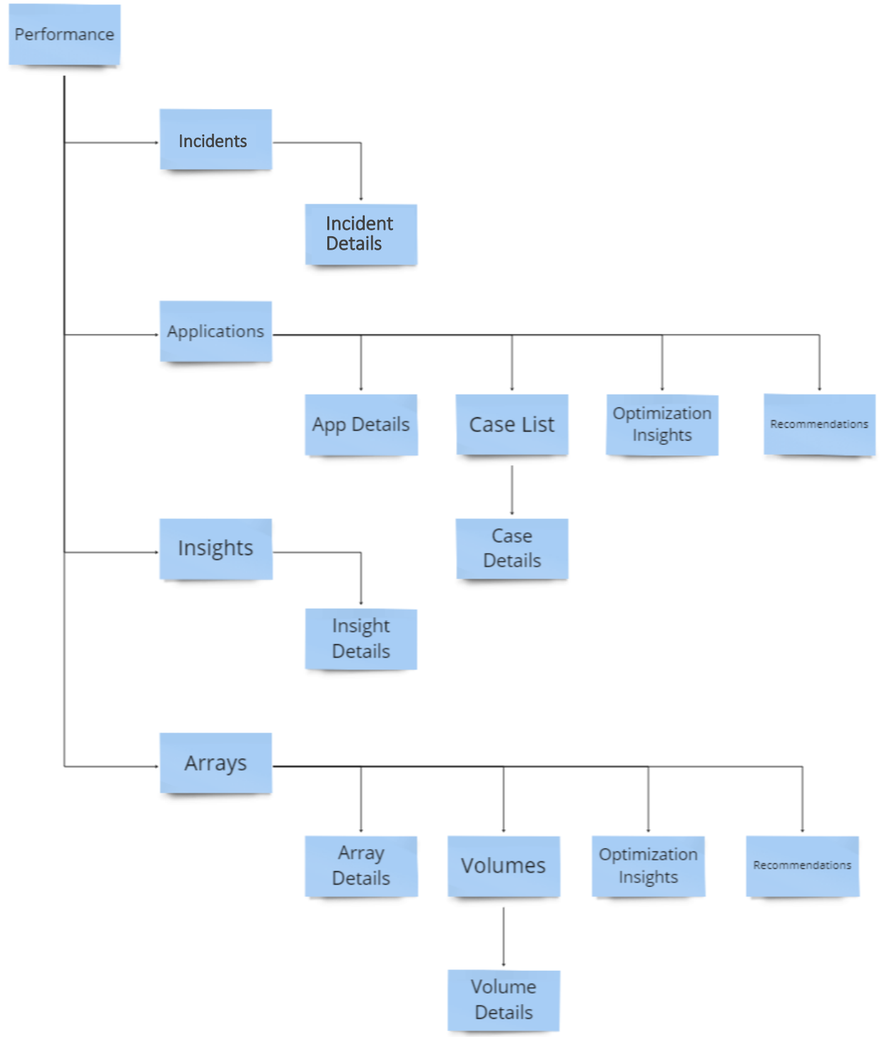
Incorporate ChatGPT to serve up natural language diagnoses and solution recommendations

## Goal 4

Bring together Capacity (storage) management into the Performance area

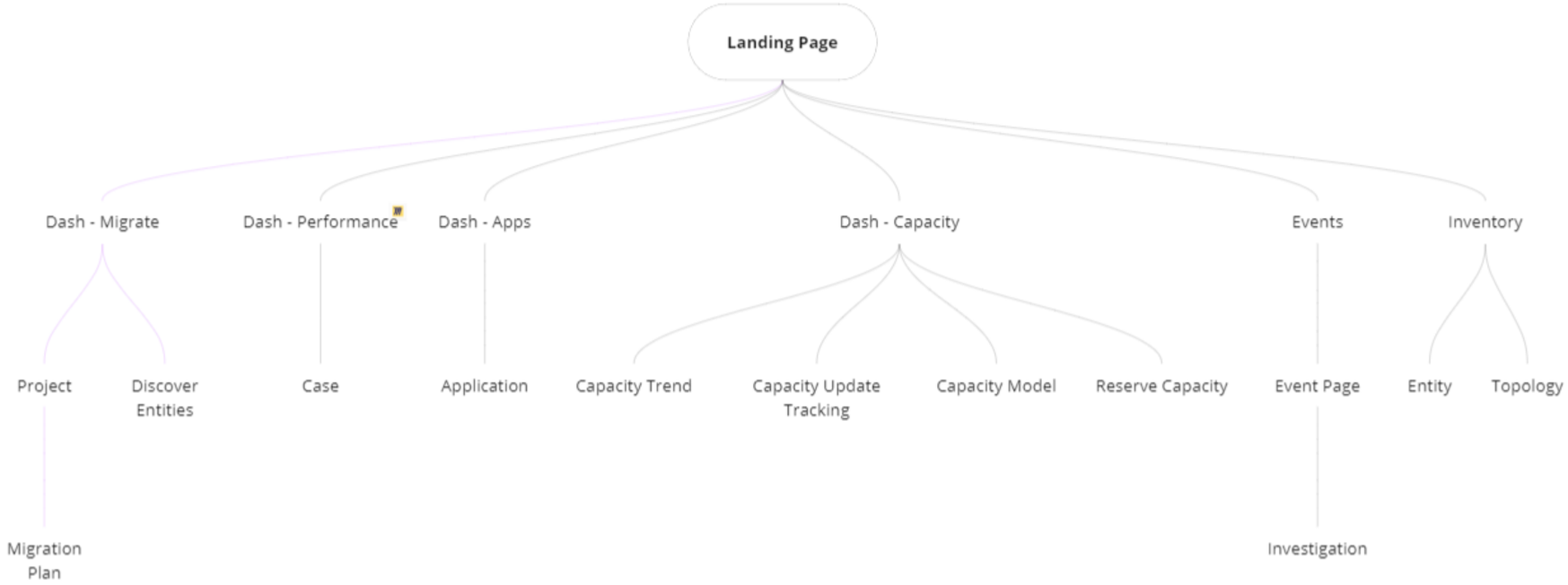
# CONCEPT HIERARCHY

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes



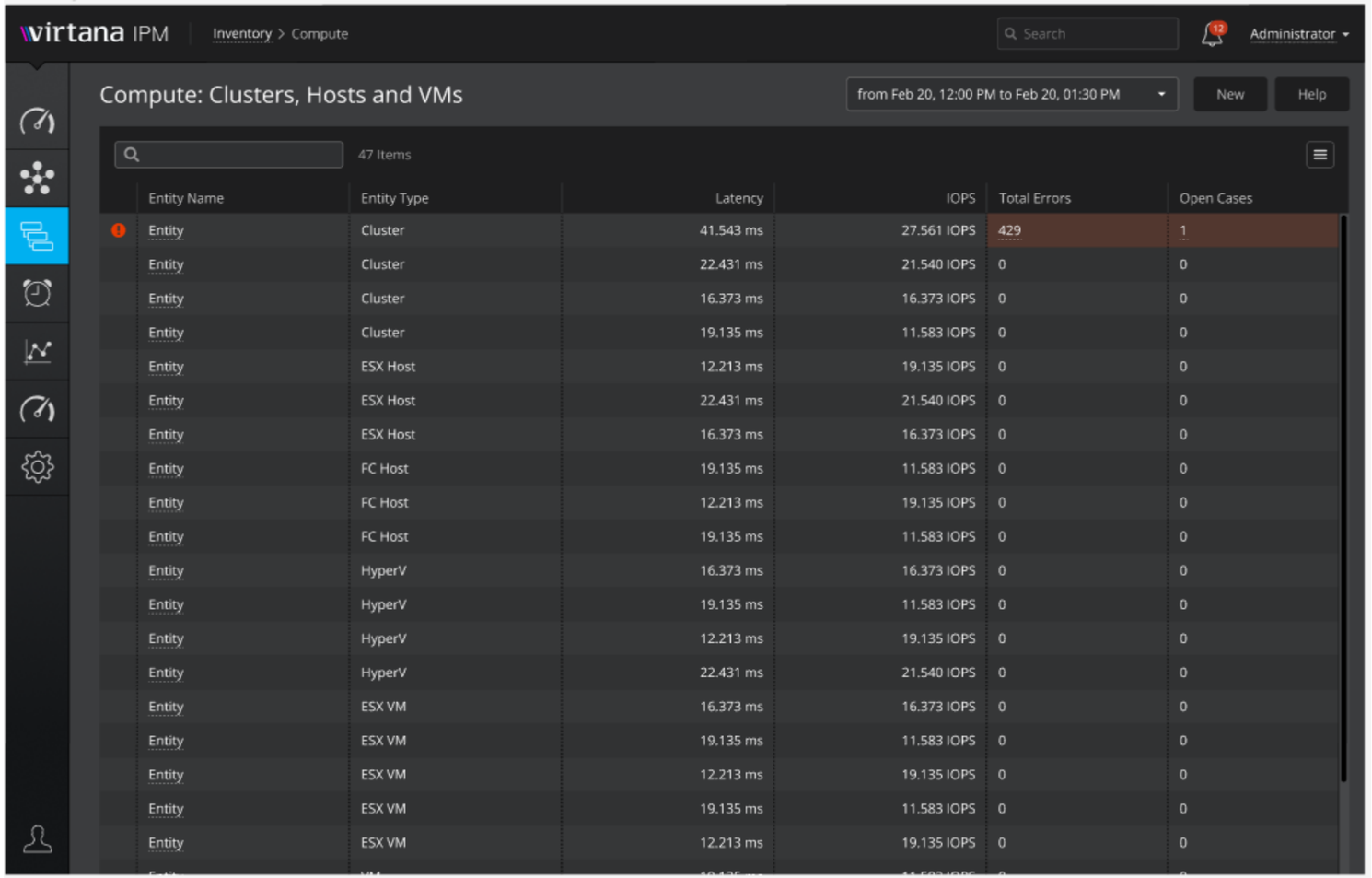
# ARCHITECTURE - SITEMAP

- Research
- Goals & strategy
- **Artifacts**
  - Personas
  - Scenarios
  - Use cases & Taskflows
  - Card Sort
  - Storymap
  - Architecture
- Design
- Outcomes



# Old On-Premise Page Style

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



# LAYOUT

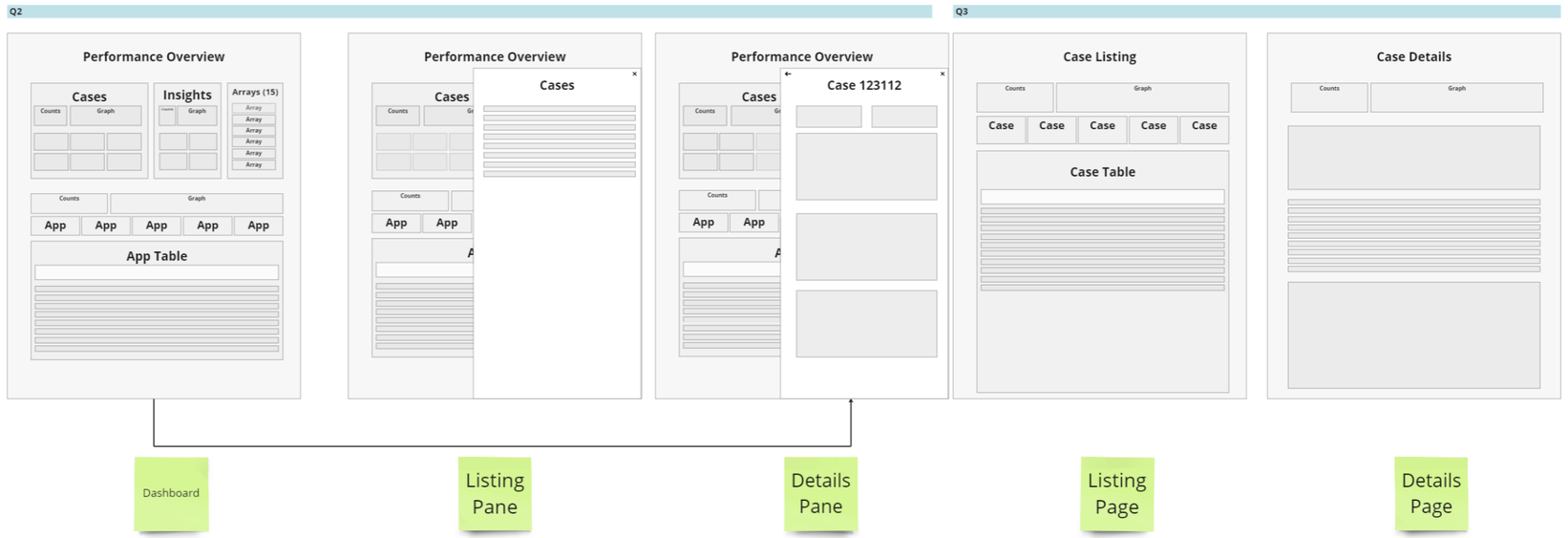
- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes

View partition


Global single pane of glass


Card-driven single pane of glass

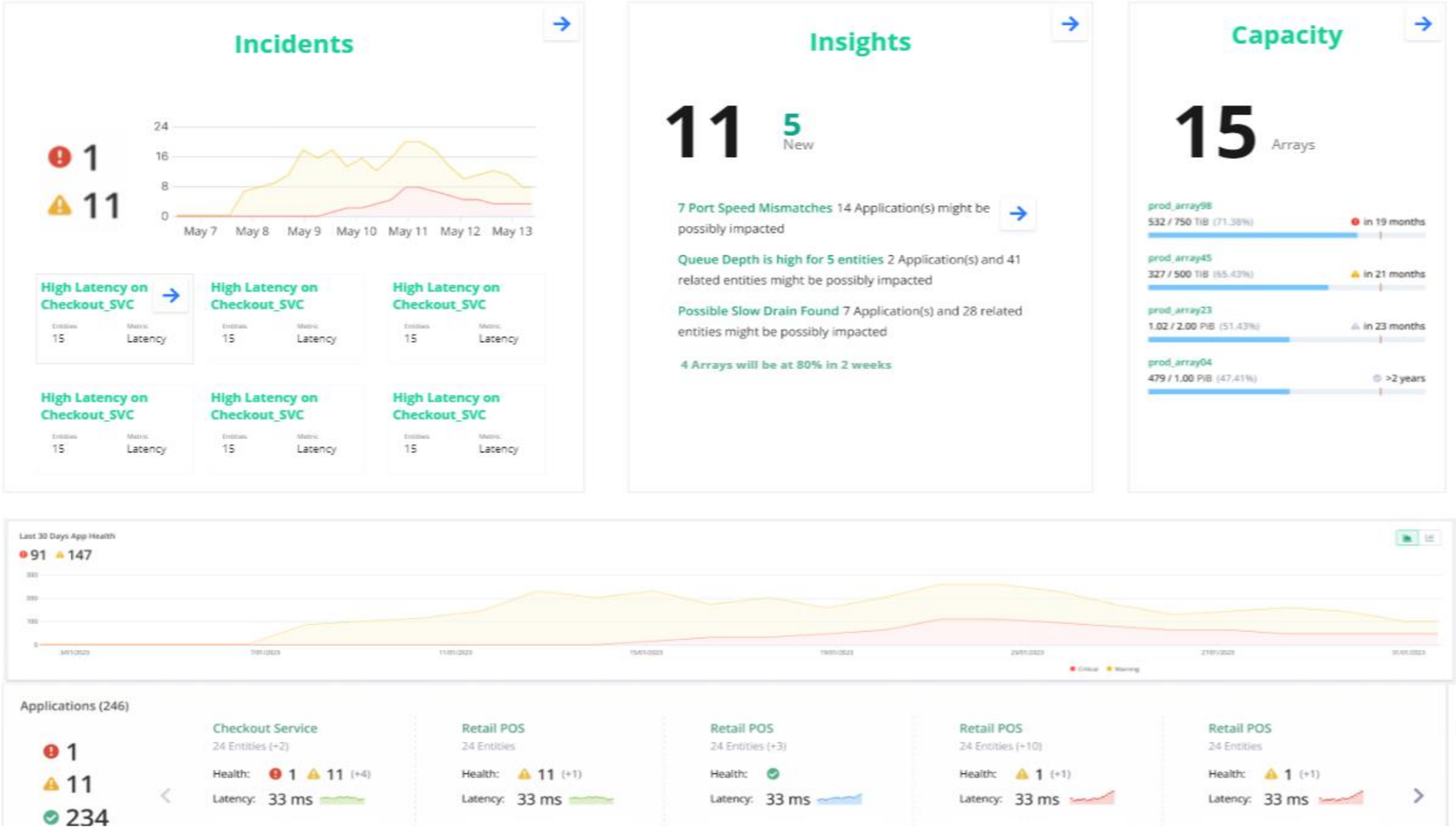

Some of the layout possibilities were whether to be card driven or to use a hybrid card + table layout to drawers



# CONCEPT DESIGN – Landing Page

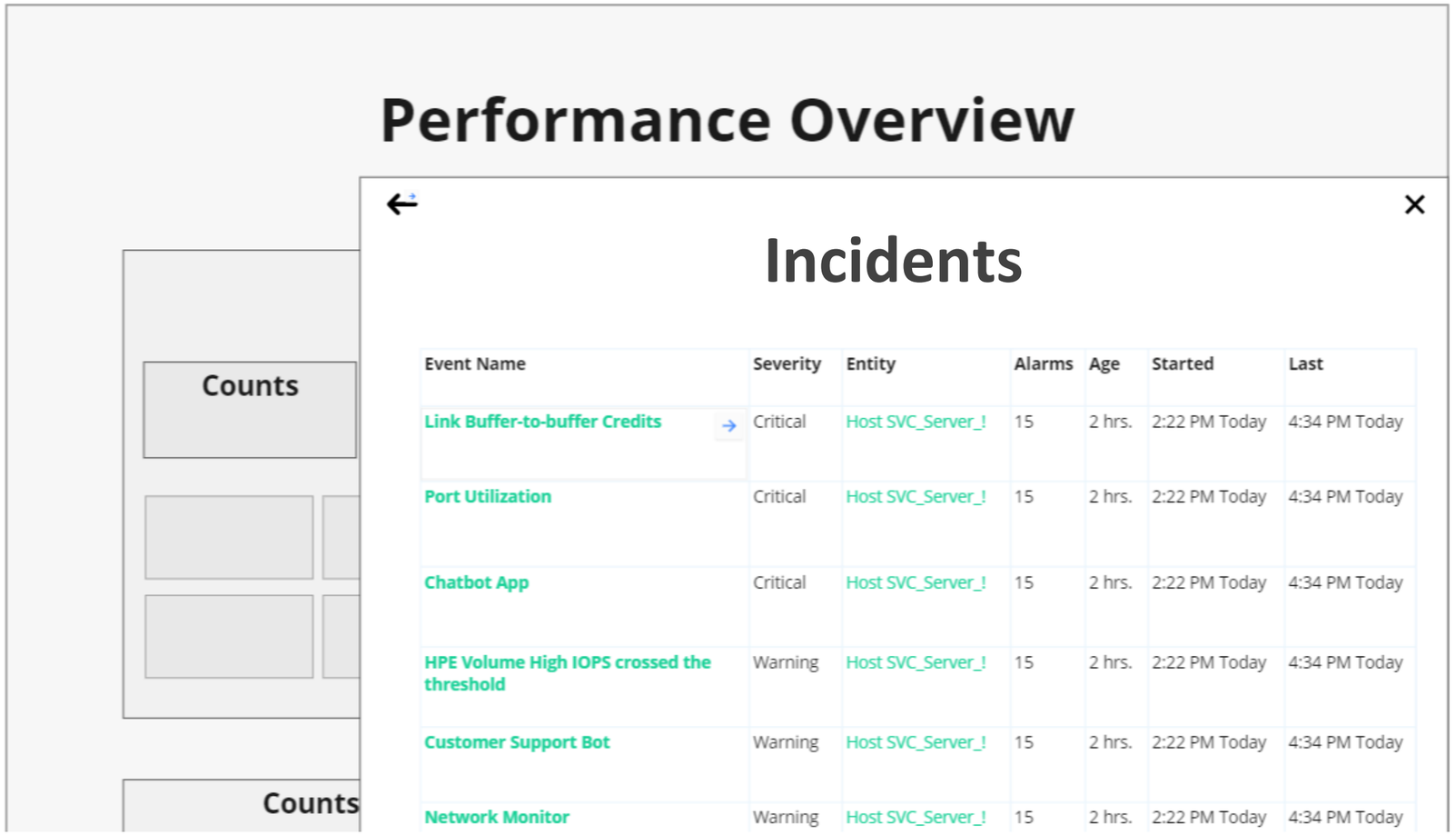
- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes

## Performance Overview



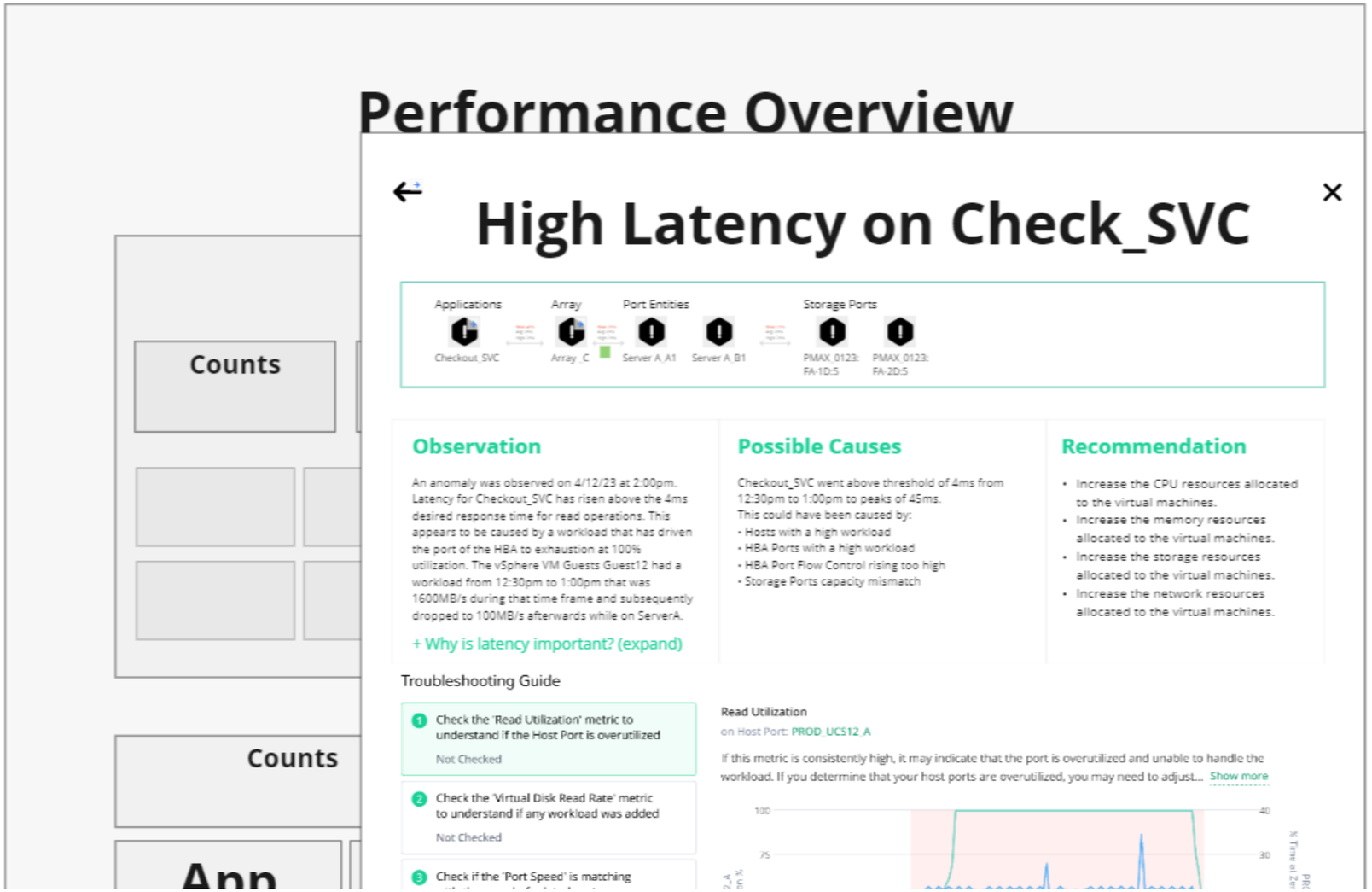
# CONCEPT DESIGN – Incidents List

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



# CONCEPT DESIGN – Incident Details

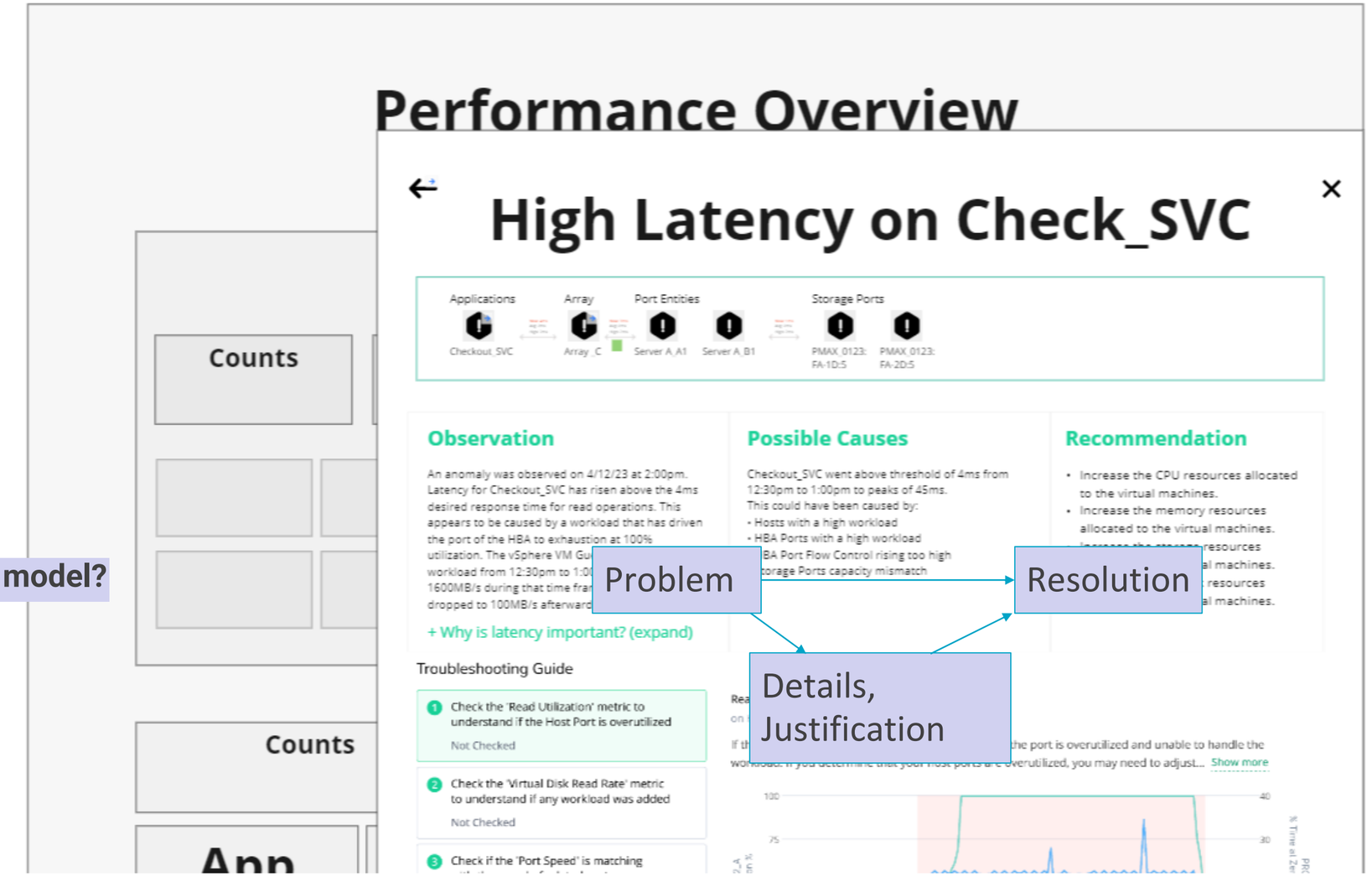
- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



# CONCEPT DESIGN – Incident Details

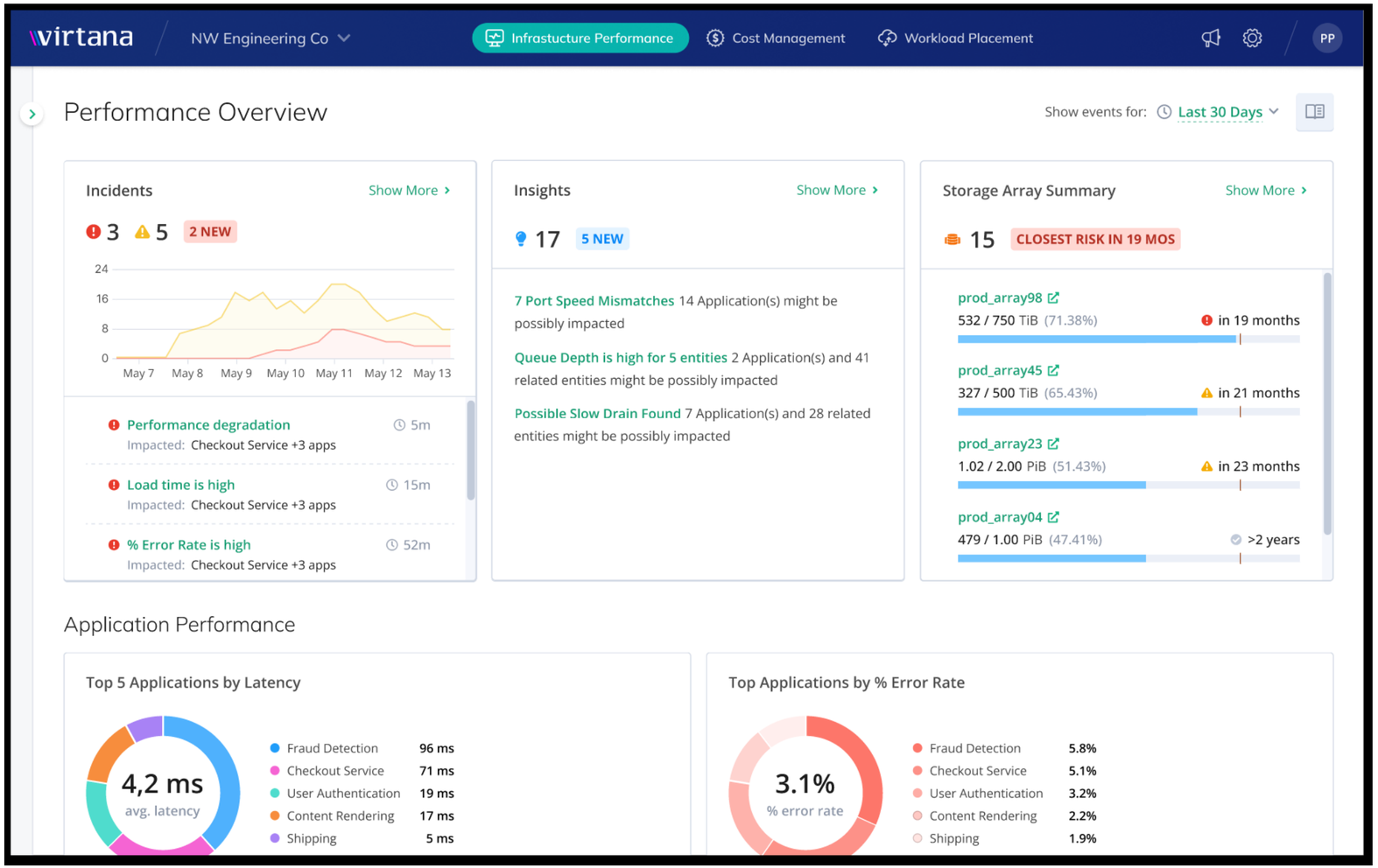
- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes

**Remember this model?**



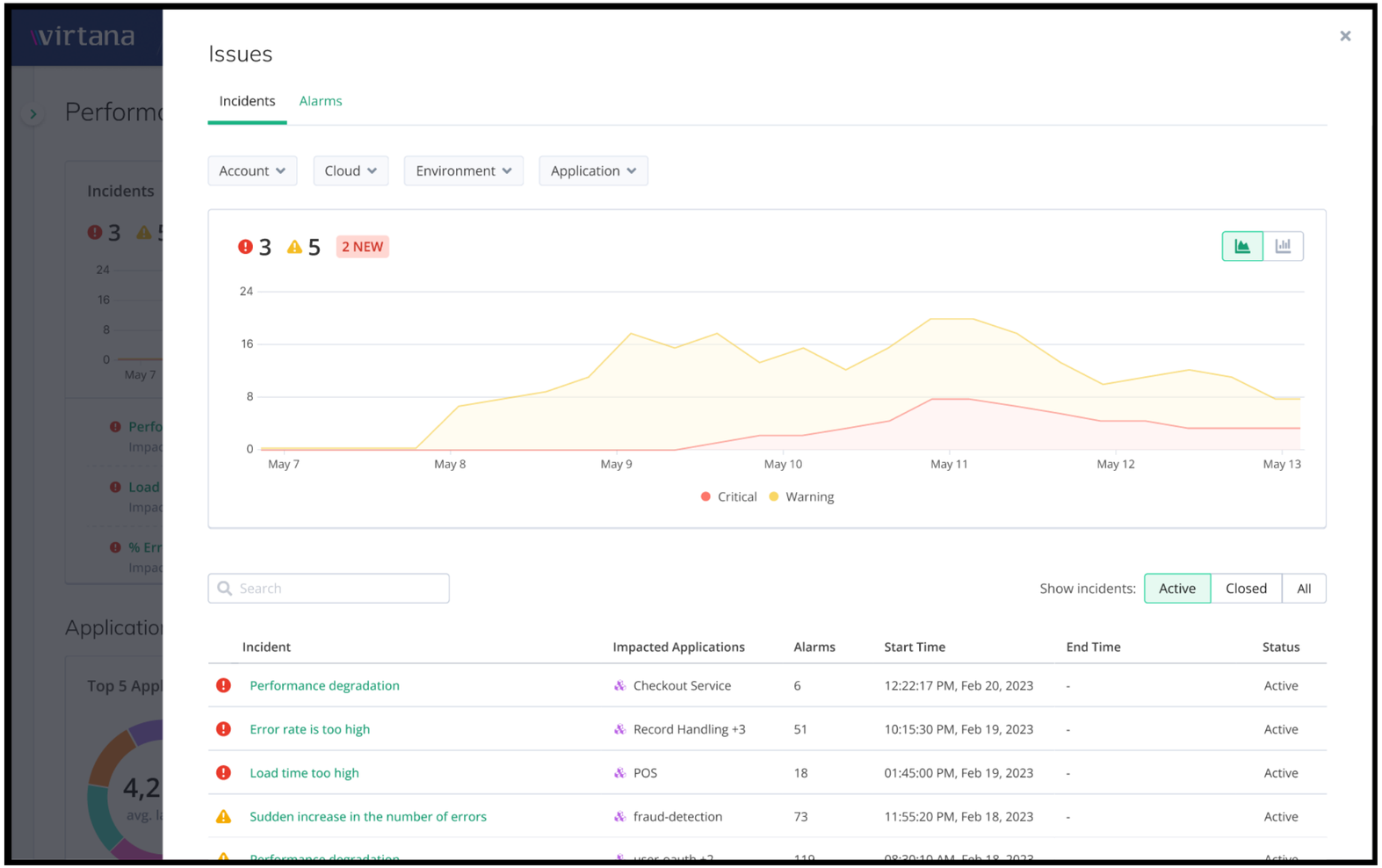
# FIGMA HIGH FIDELITY: Landing page

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



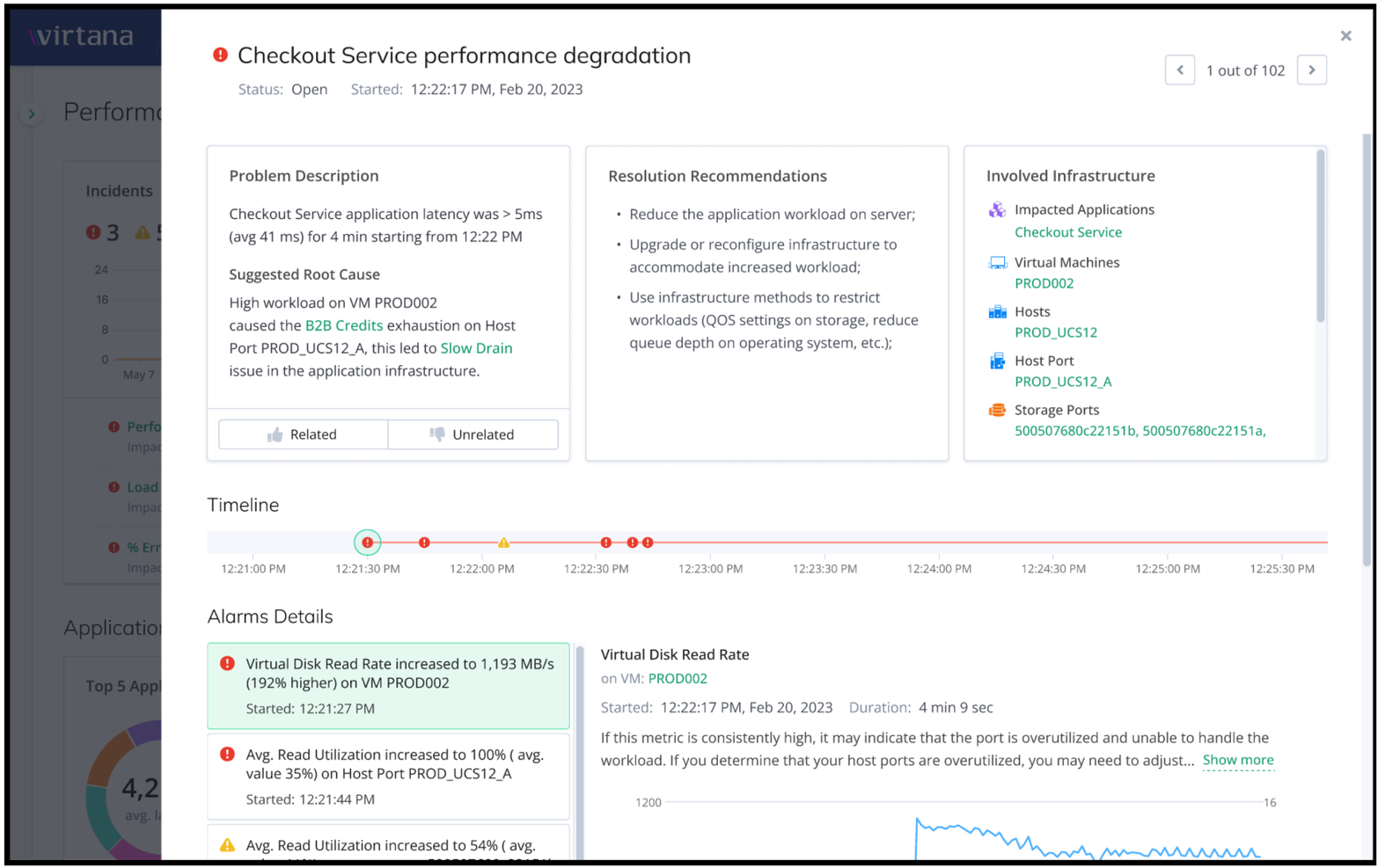
# FIGMA HIGH FIDELITY: Issues list

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



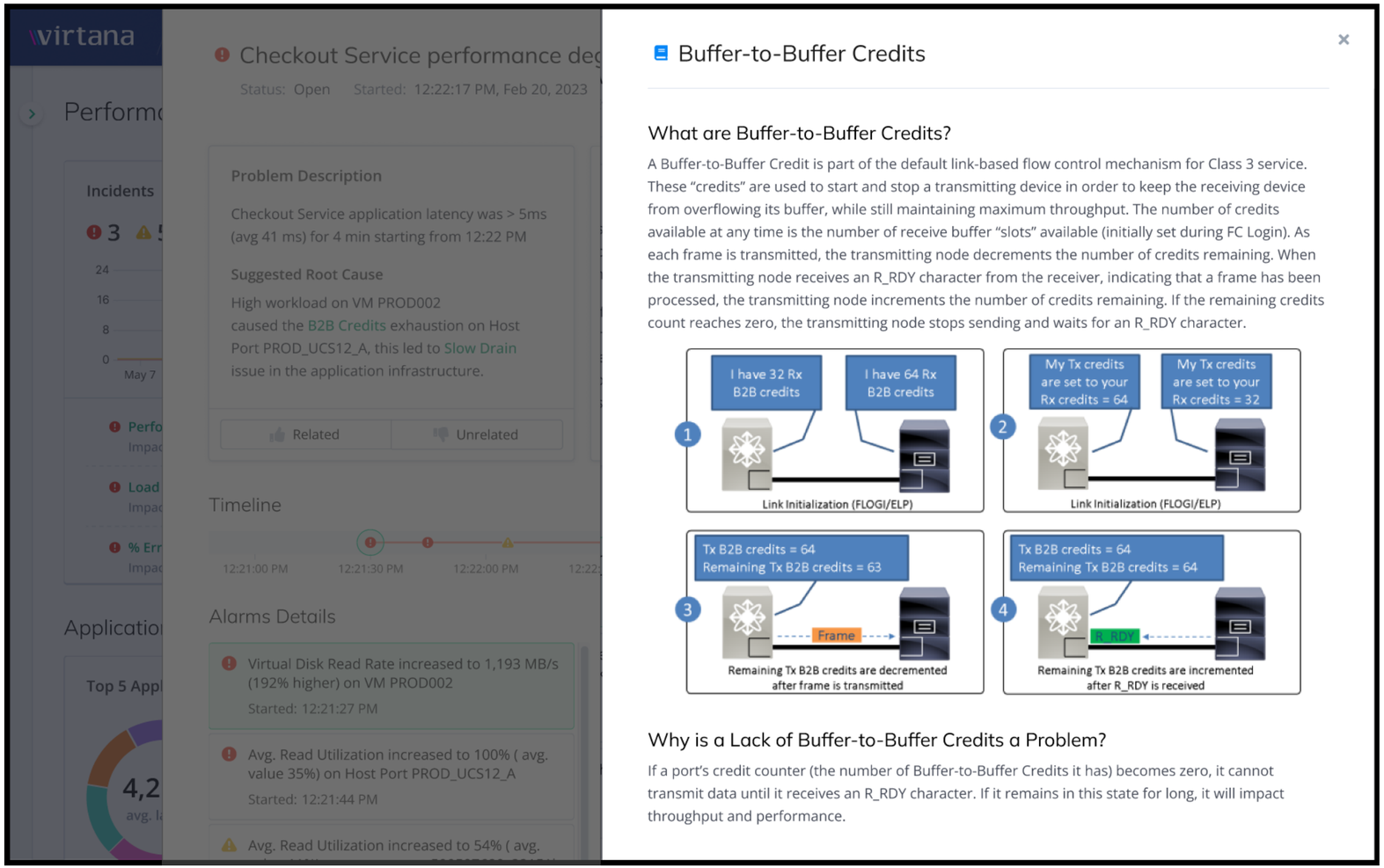
# FIGMA HIGH FIDELITY: Issue details

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



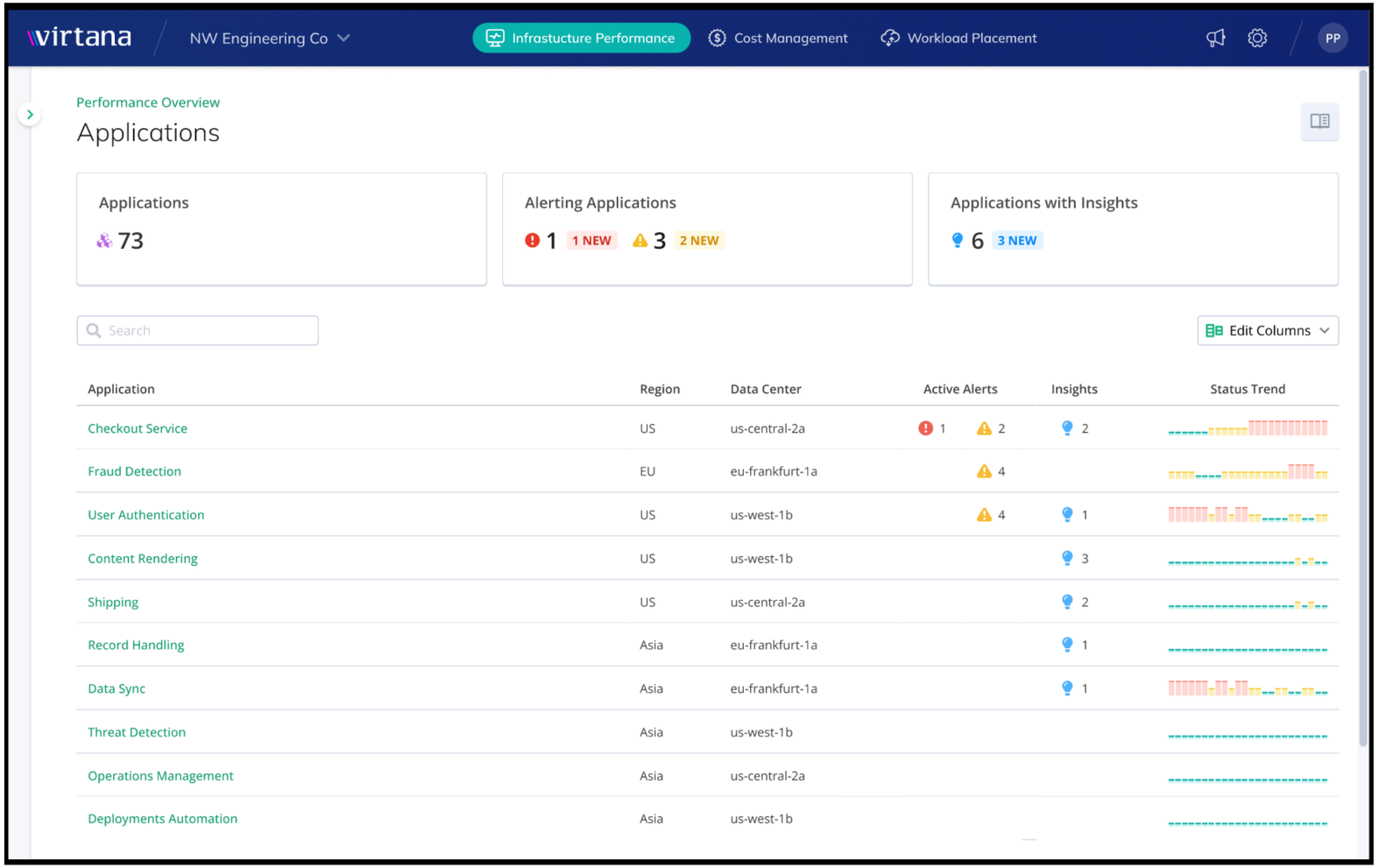
# FIGMA HIGH FIDELITY: Concept Explanation

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



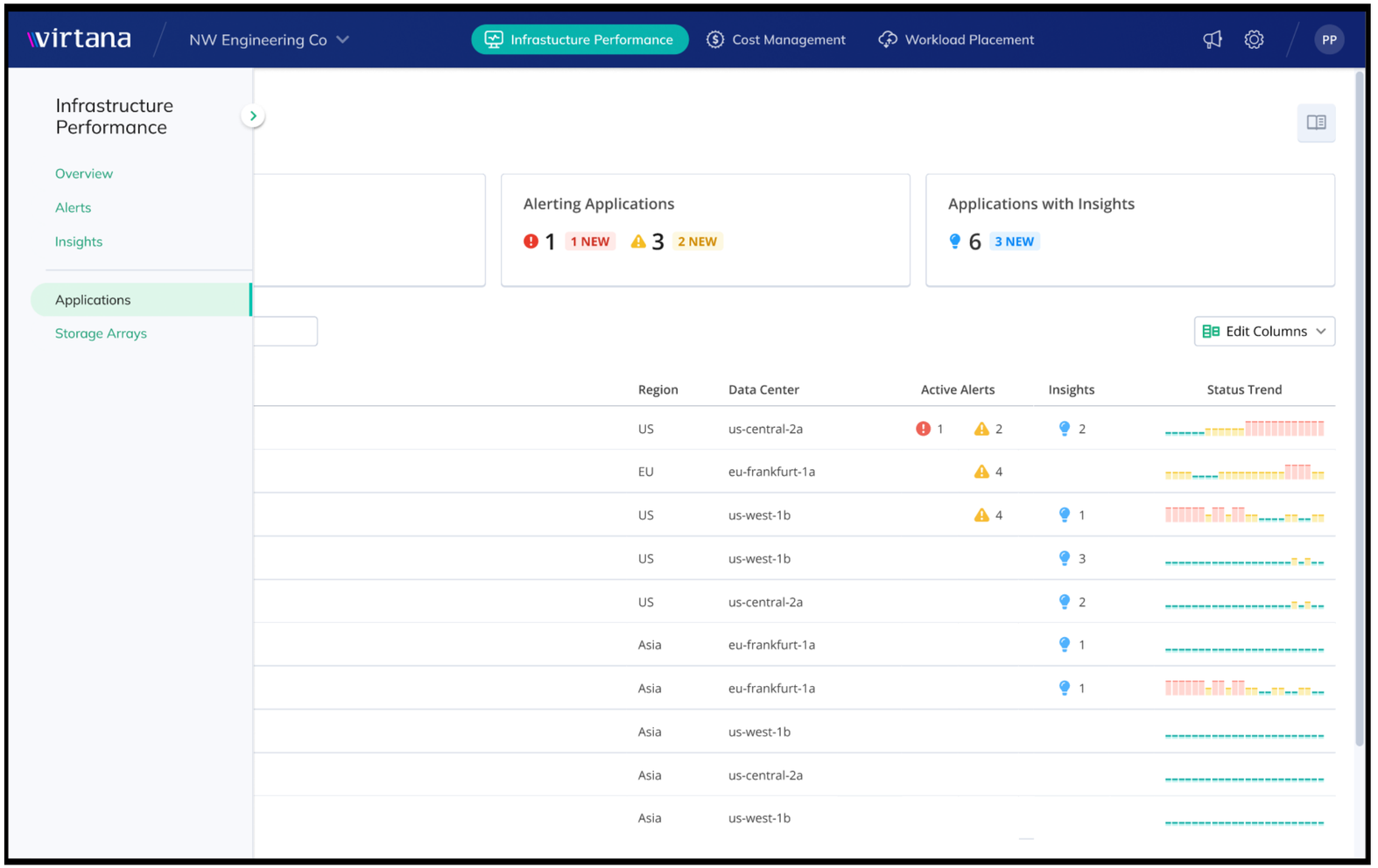
# FIGMA HIGH FIDELITY: Applications List

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



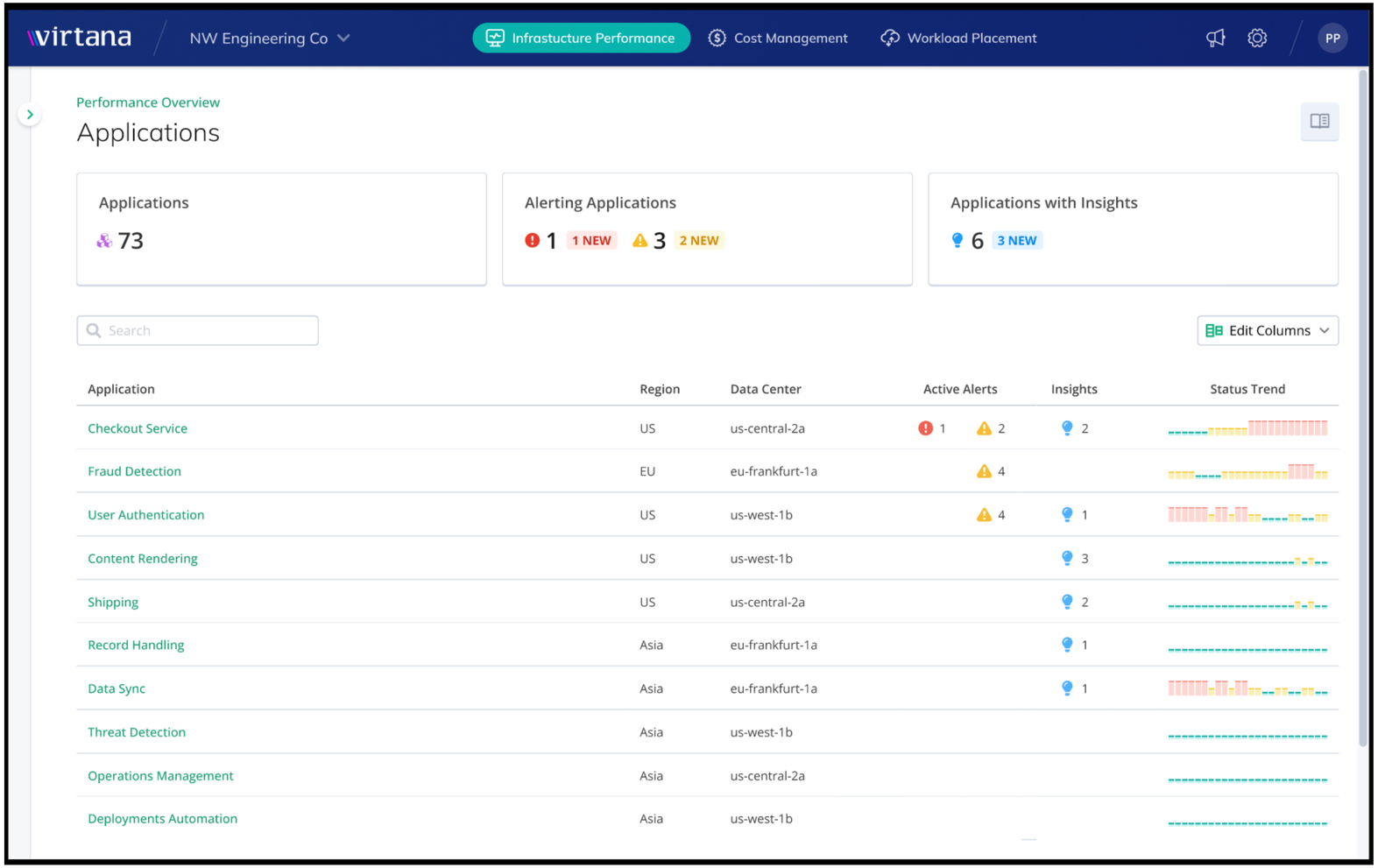
# FIGMA HIGH FIDELITY: Expandable Left Nav

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



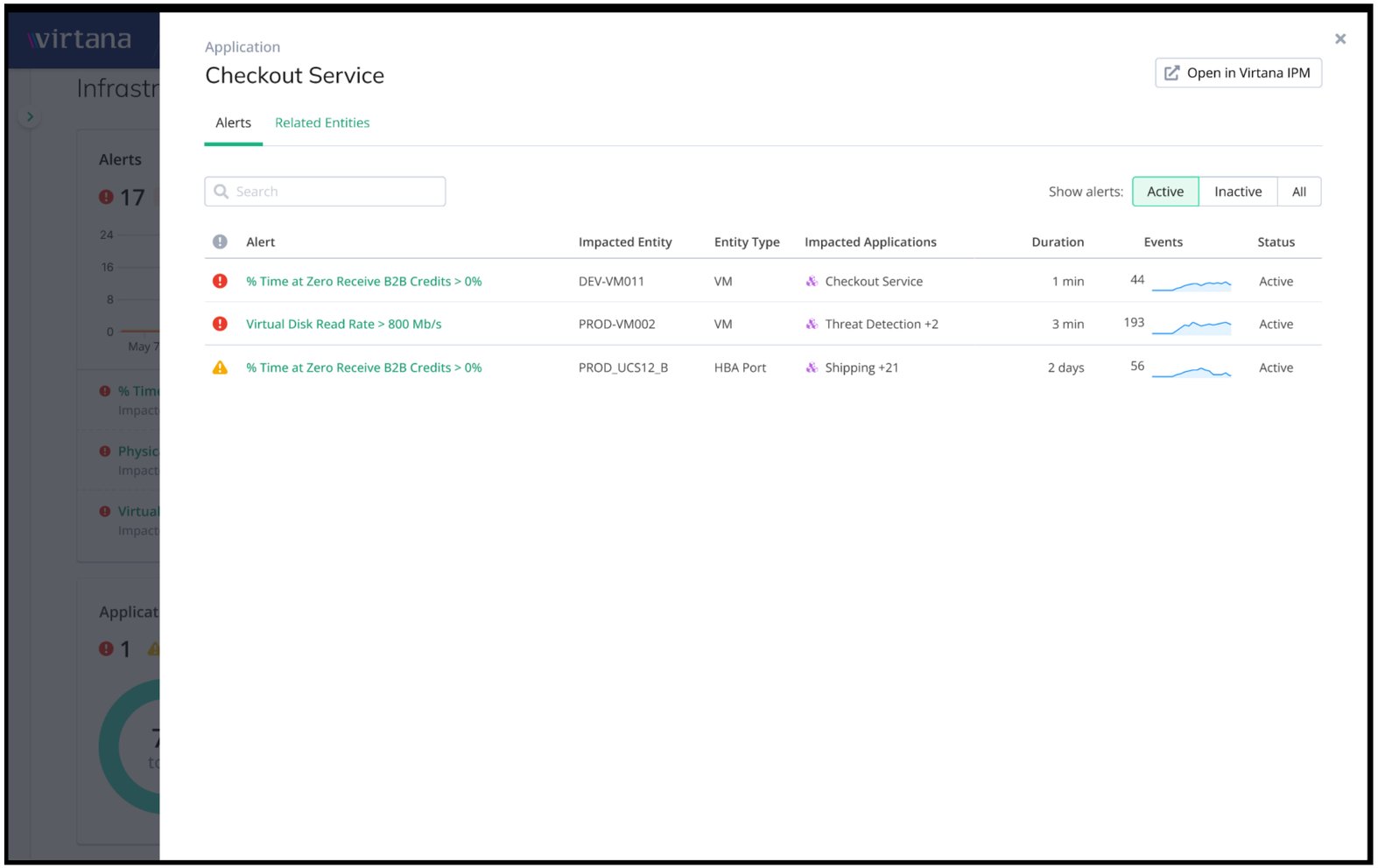
# FIGMA HIGH FIDELITY: Applications List

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



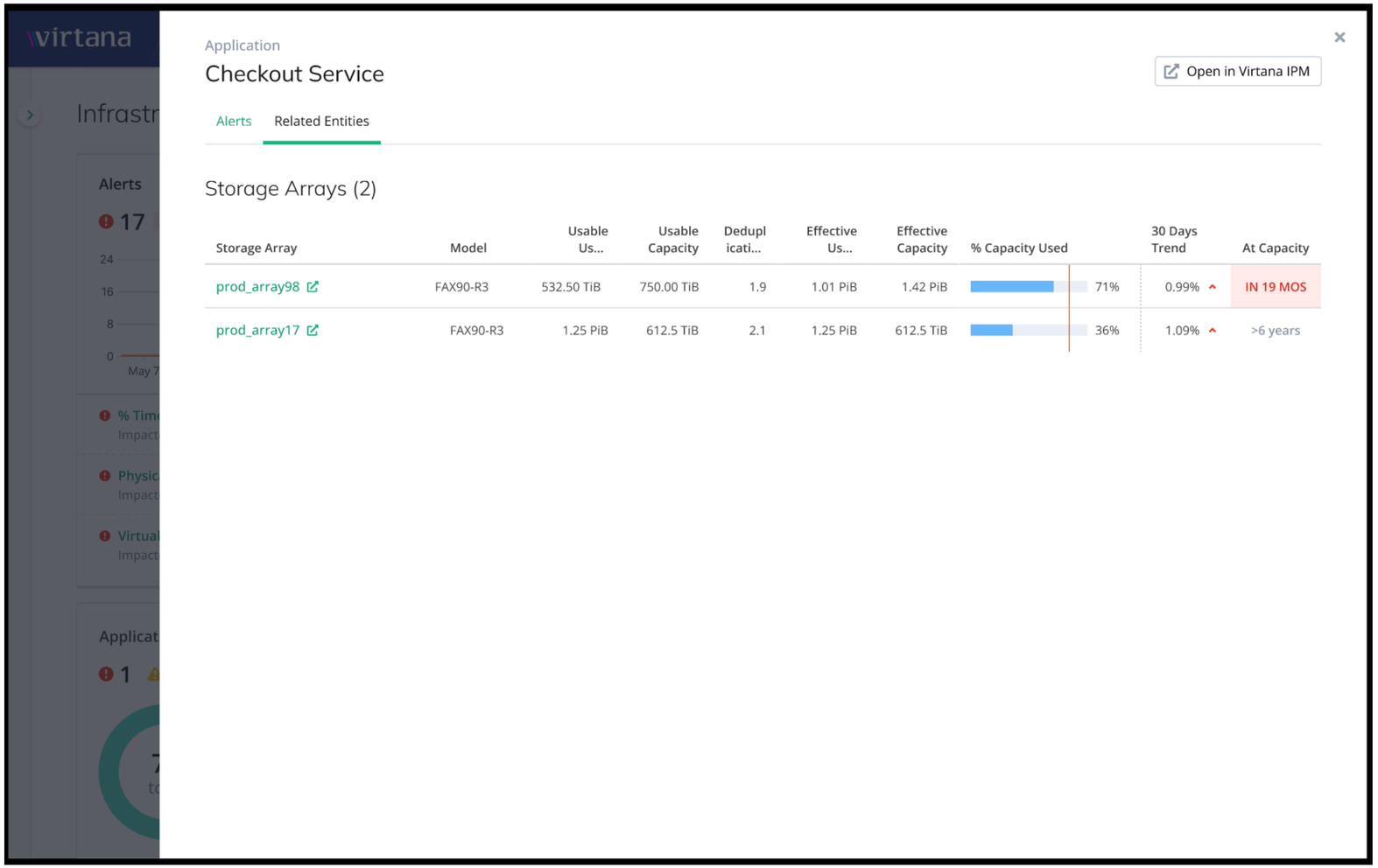
# FIGMA HIGH FIDELITY: Application Details

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



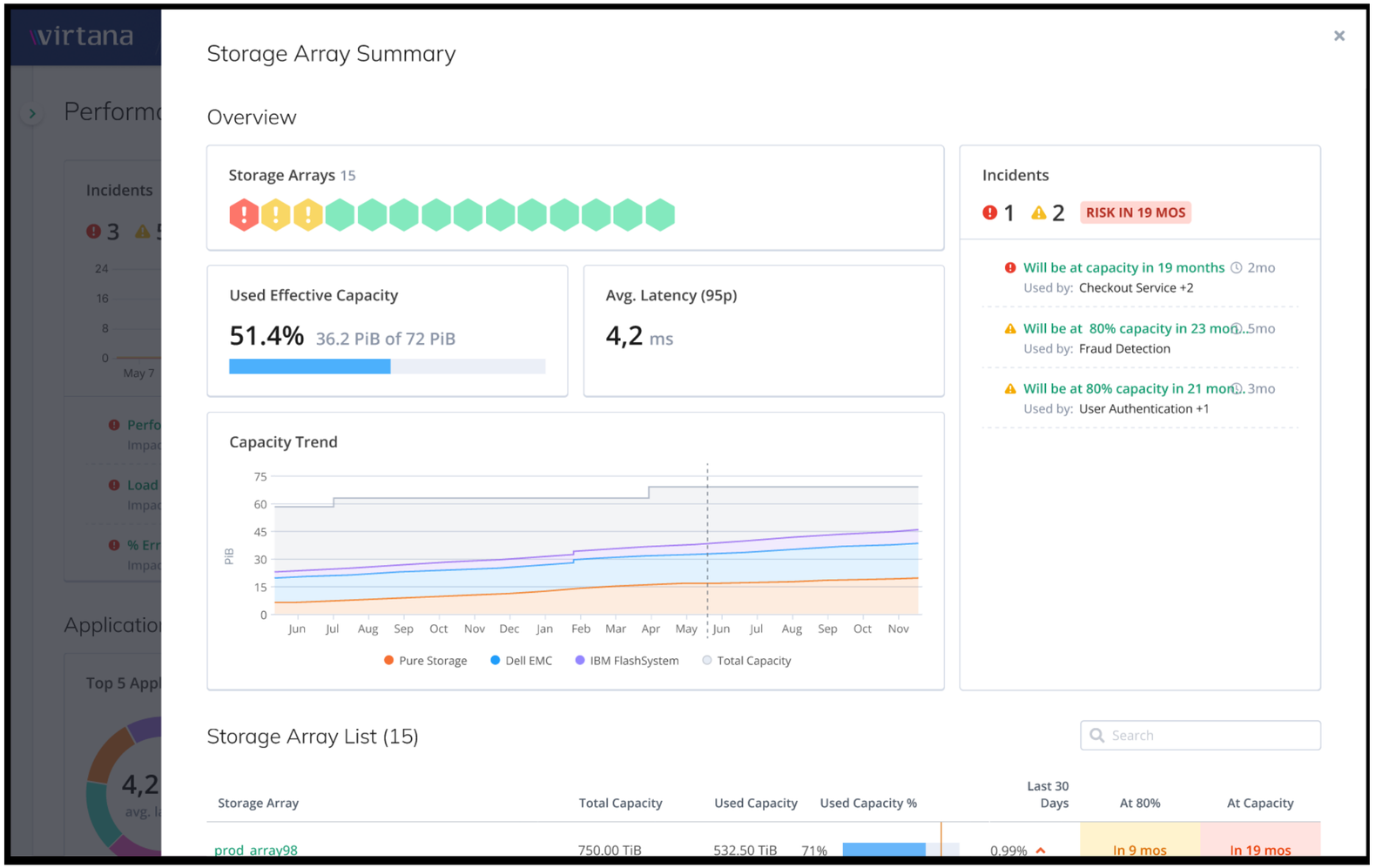
# FIGMA HIGH FIDELITY: Application - Capacity

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



# FIGMA HIGH FIDELITY: Storage Array Details

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes



# USER FEEDBACK

- Research
- Goals & strategy
- Artifacts
- **Design**
  - Wireframe Concepts
  - Figma Prototypes
  - User Feedback
- Outcomes

## Edwin Jean-Gilles - Network Architect

IHG HOTELS & RESORTS

*"What I like is that I'm not bombarded with information, but I can drill down to the metrics and the graphs to get a detailed look!"*

## Ron Neroes - Storage Engineer

DIRECTV

*"Nice to spell out the problem, causes and recommendations could help junior team members out.*

## Jeremy Huddle - Security Engineer

Palo Alto Networks

*"I love the recommendations ... that would be a differentiator*

# OUTCOMES

Research

Goals & strategy

Artifacts

Design

**Outcomes**

- Based on my analysis of the initial user research, I was able to make specific recommendations of use cases to pursue and features to build as a UX vision

**Simplified interaction models, reduced navigation/complexity/data density**

**AI & Insights into: What to do, what's next, being pro-active**

**Higher value / order experiences to enhance productivity**

- Imagine a slider that lets a user mediate between performance and cost

**Re-envisioning the platform as a knowledge / collaboration environment particularly around wisdom and expertise capture**

- Multiple teams interacting to drill into a problem
- Complex problem solving in a team-centric

- The product manager was immediately wrote up the full list of recommendations as an epic

- Virtana's executives agreed with my conclusions and the proposed direction. They also recognized that the prototypes tested well.
- Engineering produced the first version of AI-driven Troubleshooting by creating working POC's within 2 months incorporating ChatGPT functionality

servicenow

# Component Testing Console

## Project and Goal

ServiceNow needed a solution for an internal tool experience to monitor the outcomes of component performance testing

## Activities

- Worked with PM, users, engineers, and visual designers
- User research, concepting, wireframing
- Conducted research with PM, designed the initial concepts, and collaborated with visual designers on the final look and feel.

## Outcomes

- By innovating on the interaction model, I was able to create a product that developers were far more productive with.

servicenow

# Project Goals

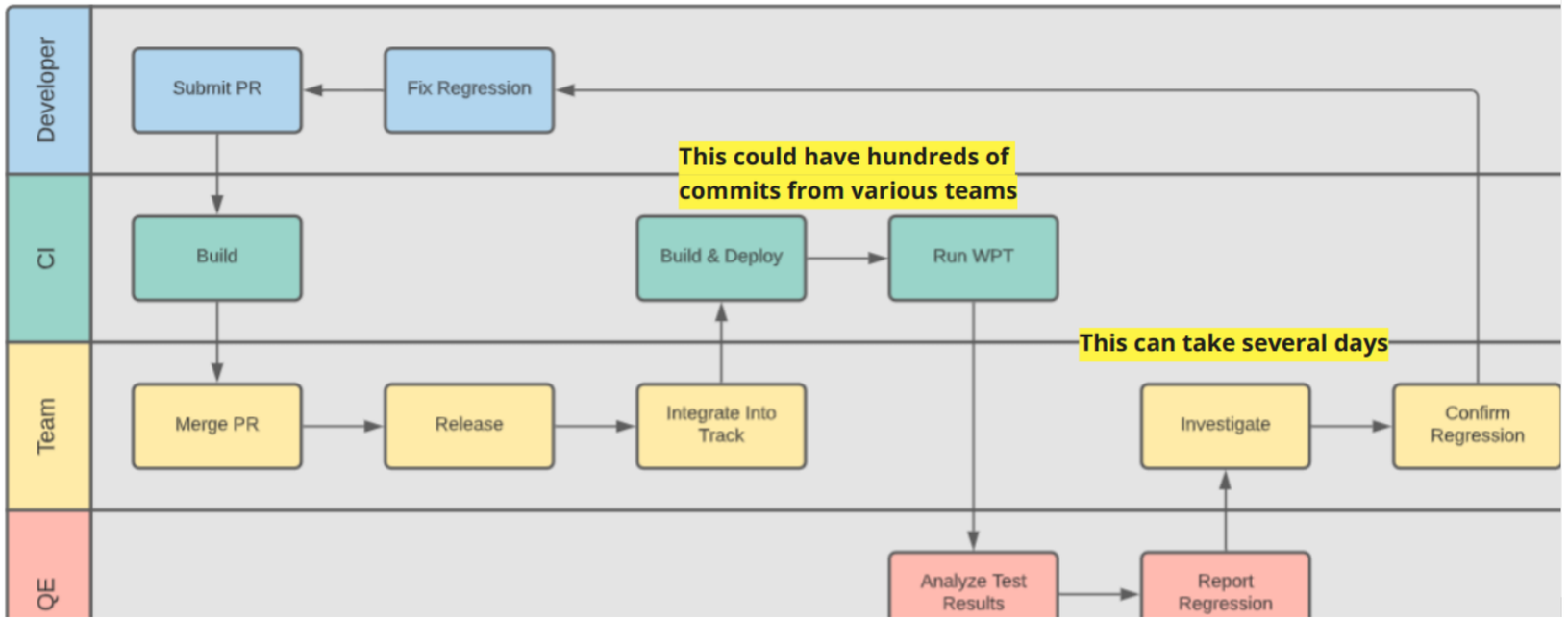
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

- Utility - Help engineers identify performance regressions and improvements early in dev cycle
- Quick feedback about performance results
- Provide UI to help understand results
- Generate data, make it actionable

# Problem Context

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interaction
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototyp

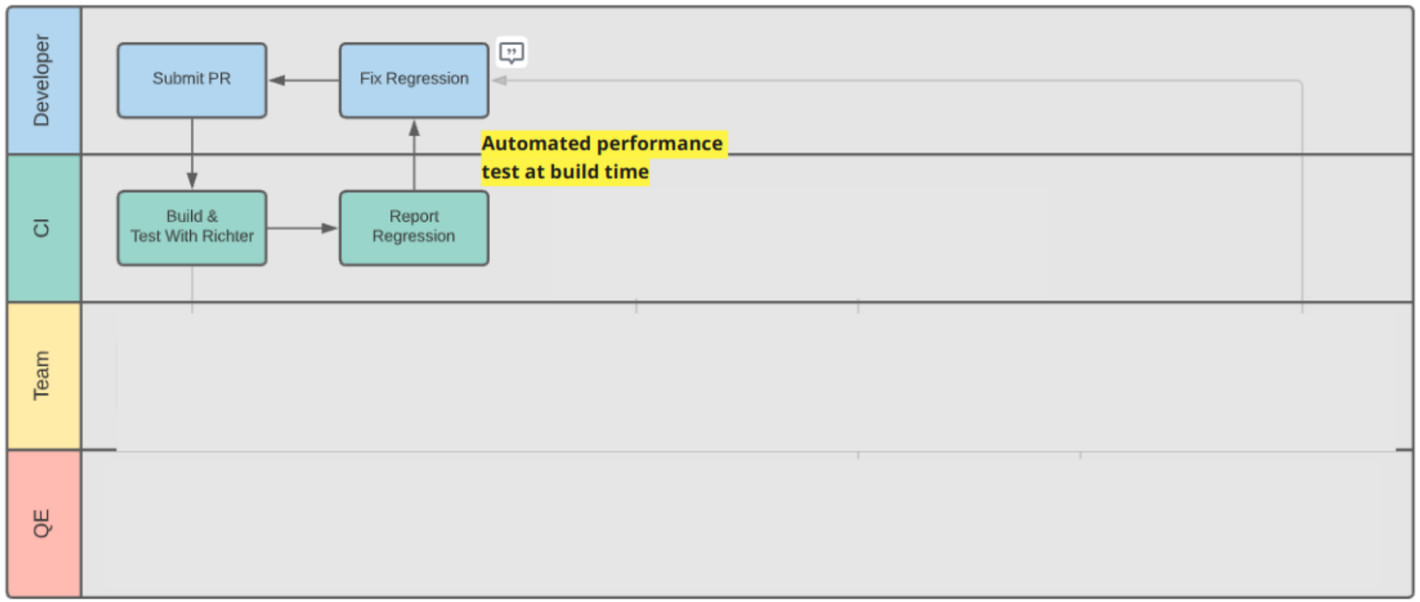
1. **Hard to identify root cause of performance regressions at the application/page level when there are multiple changes from different components. Long investigation cycles and churn.**
2. **Current component performance testing leverages current WPT infrastructure and has a high overhead and maintenance cost.**



# Revised Model

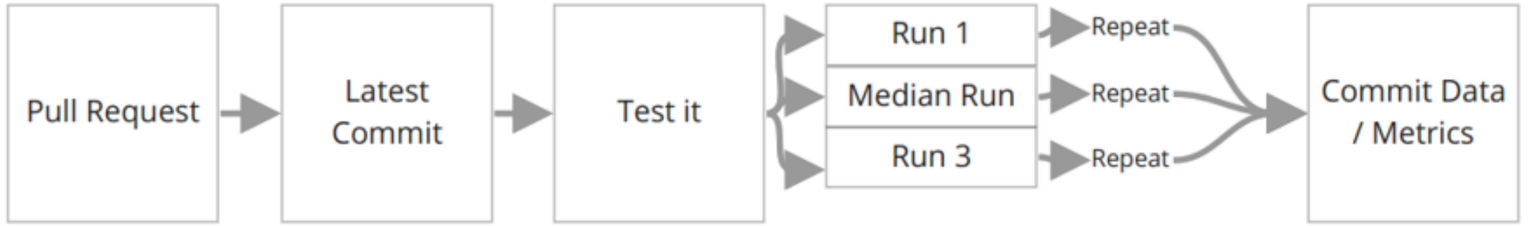
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

1. Shorten feedback cycle on performance impact for component authors from days to minutes via build automation
2. Controlled environment test component performance in isolation first
3. Provide a summary view if there is a performance impact
4. Provide test results details that shorten investigation



# Data Flow

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

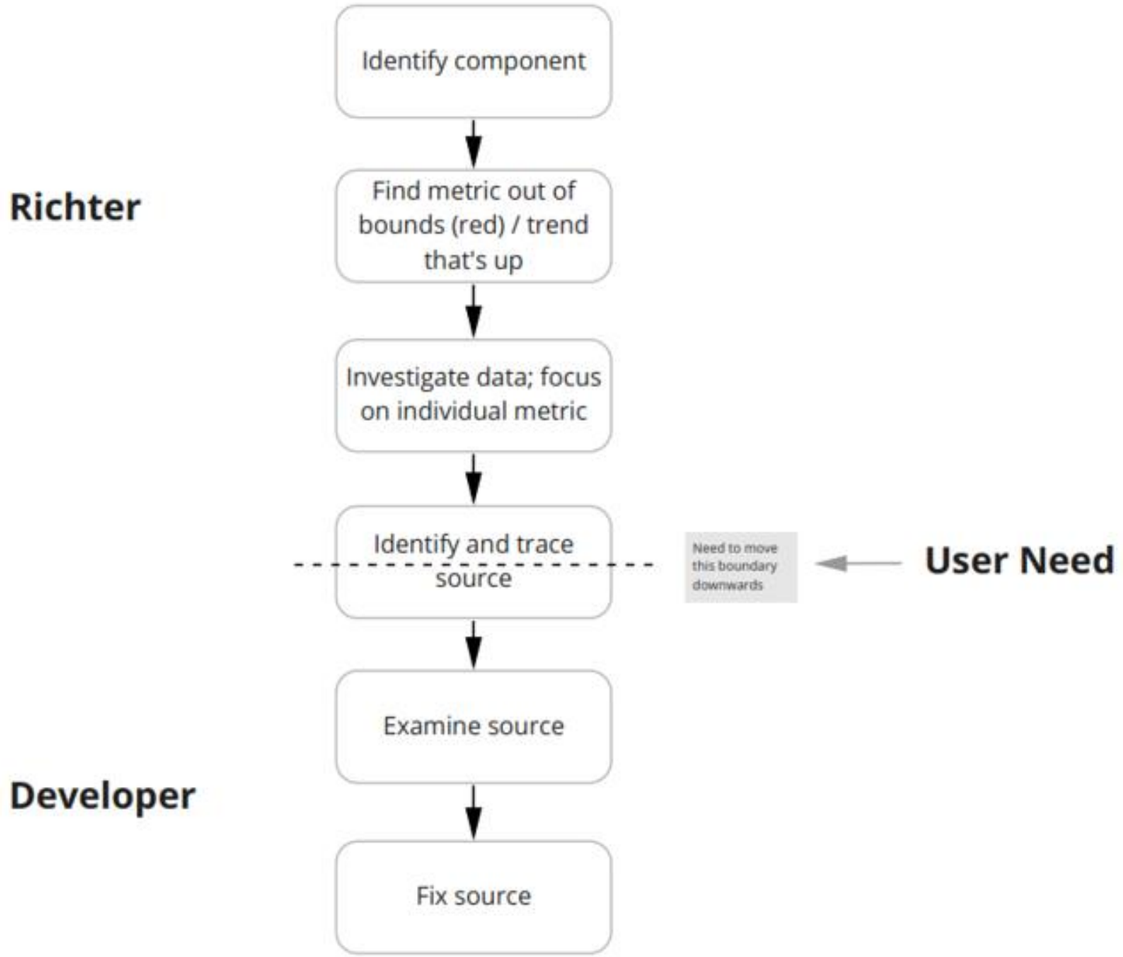



**Data Shape**

- One component
  - Commit (3-8 per day)
    - Median Run data
      - Multiple Metrics
      - API List

# Abstracted Use Case

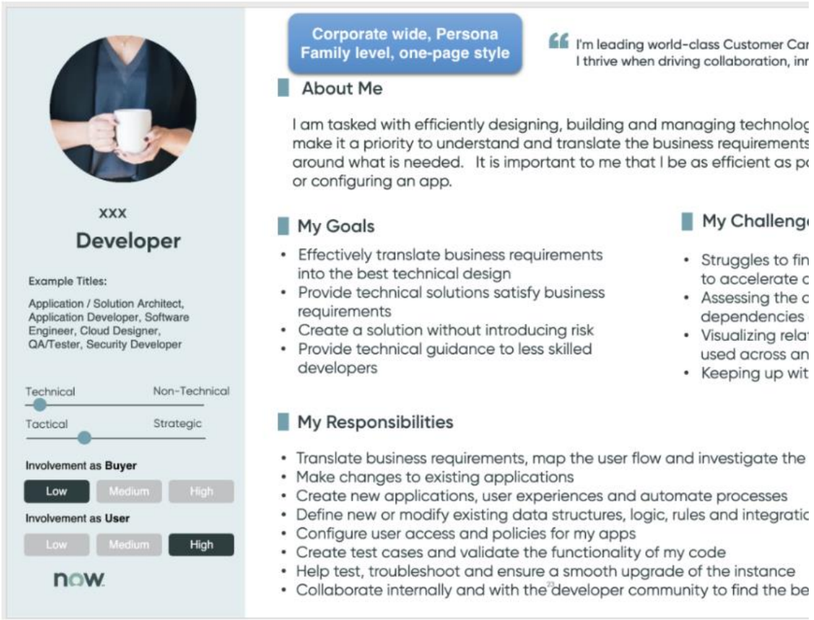
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes



# User Personas

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

- Devs/QEs working on authoring or changing components at the code level
- A given developer focuses on one component




- **Component Dev: I want to see whether there's a performance impact from the code changes I checked in**
- **QE: I see performance problems in checking this component and want to share with the developer**
- ***I want to see what's broken***
- ***I want to show my boss all the fixes/improvements***

servicenow

# User Interviews

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

## We Saw

- Users want to focus on trends over time (increasing is bad)
- Need to know what is out of bounds (establish the baseline)
- Users seemed to specifically want to focus on one metric, not see all of them when the drilled in
- Users care more about the *median* data, rather than individual runs (summary & progressive disclosure)

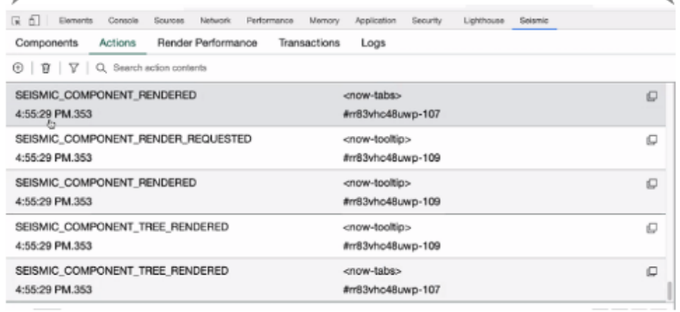
## We Concluded

- No need to have users go through a component chooser screen
    - Could be a popup or other control
- Multi-metric graphs are fairly hard to scan as the legend has to be referenced visually
- Users want to focus on a single metric
    - Let users do a 'close up' of a selected metric
- Drilling down should be fluid
- Tab structure doesn't scale (and users mainly care about median data)

# User Themes

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

| Performance | Actions | Traces | Causality |
| --- | --- | --- | --- |
| Raw numbers that show how the execution is performing | Showing when components get rendered and how many times | See the flow of information as the code executes and the page renders | Find out if there are known problems with the rendering and data trace as represented in the component metrics |



# User Goals and Flows

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

**User Goals**

| Understand | Inspect | Optimize |
| --- | --- | --- |
| Understand whether there are legitimate performance deviations and which commit is responsible | Find problems within the detailed traces of the page building process | Find out if there are known problems with the execution trace of the code as represented in the component metrics |

**Tasks**

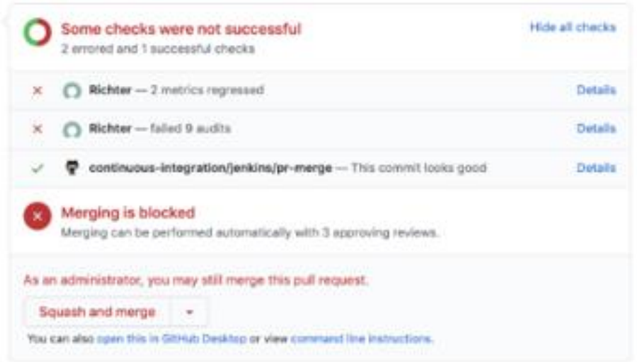
1. Get more information on the deviation
2. Trace the deviation to a particular commit
3. Understand the mechanics of the performance degradation
4. Understand what issues are present and what can be done for performance

# Competitive Analysis

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes


SpeedCurve

DebugBear

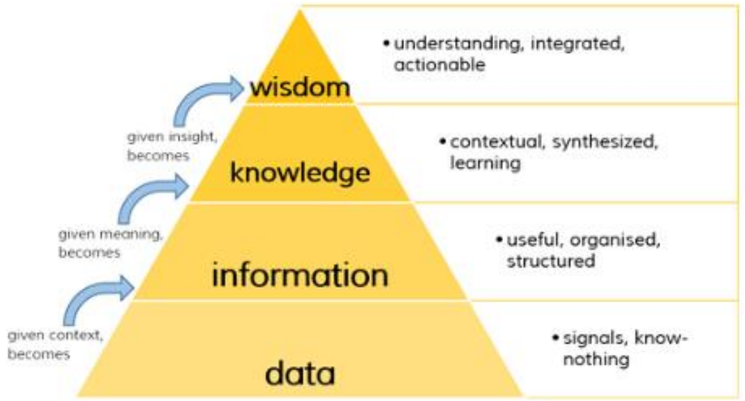



Github

# Design North Stars

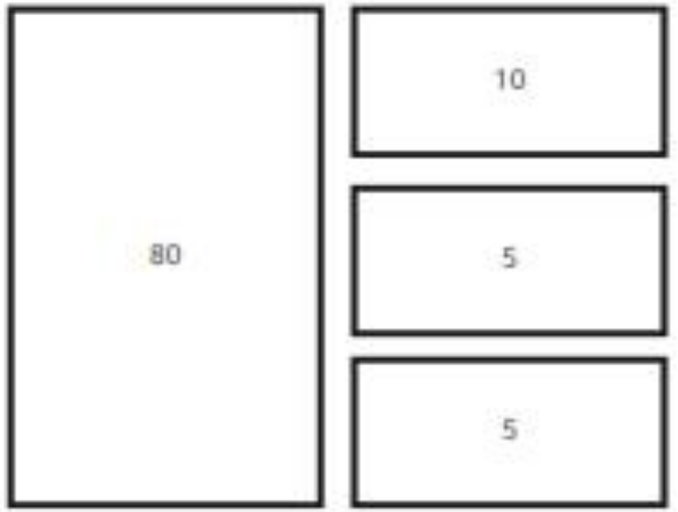
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8.

| INVESTIGATIVE | Higher Order | Visual | Create Value |
|---|---|---|---|
| The design supports the beginning stages of root cause analysis | Wisdom not data<br><br>Point to problem areas for the user and interpret them | Use visual cognition to allow users to recognize out of bounds conditions quickly | Create value by interpreting data, summarizing information, and recognizing patterns to deliver recommendations and best practices |



# 80/20 Rule

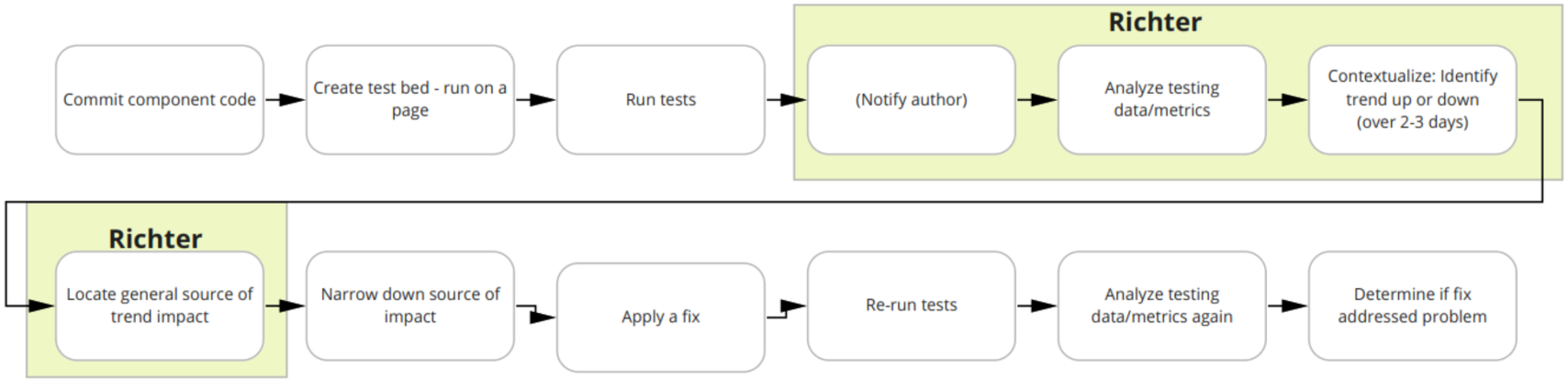
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

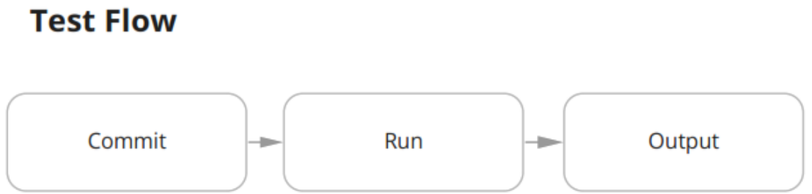



**80% of the time should be spent on the first page before navigating**
**80% of the useful info should be on the first page**

# Task Flows

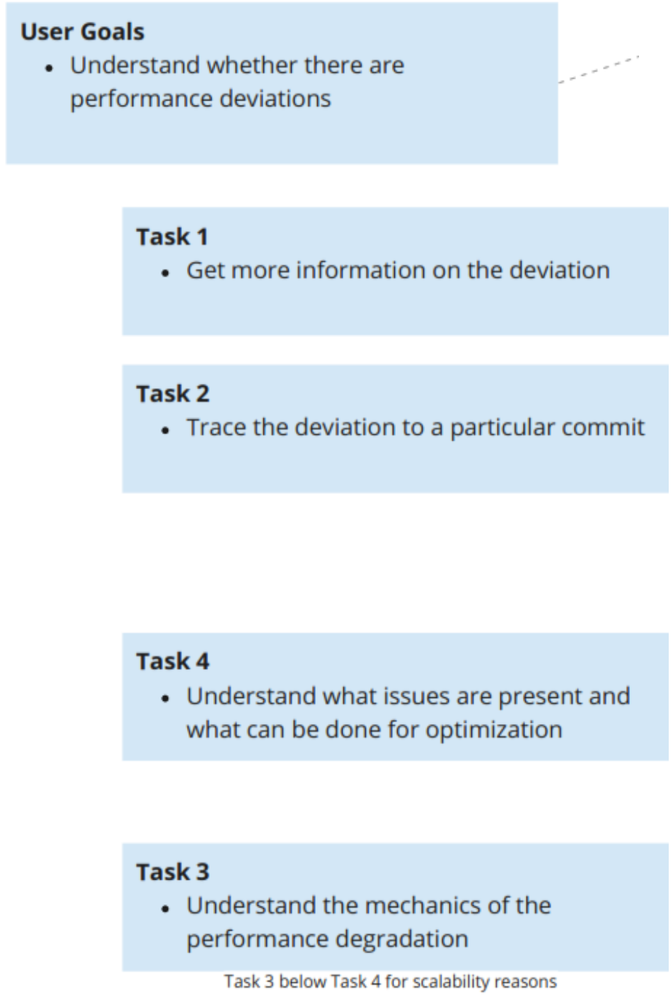
1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes



# Task Hierarchy

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototype

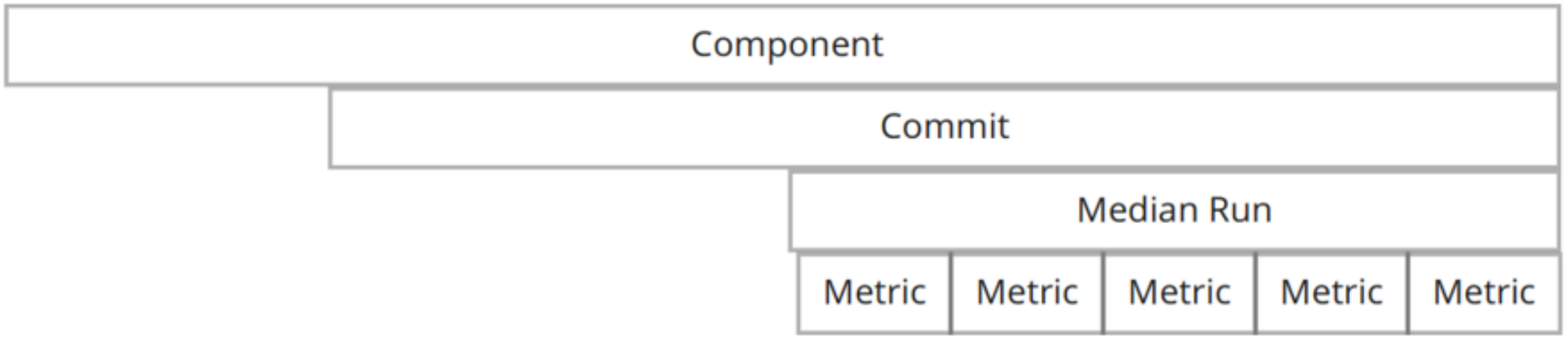


# System Concepts

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

# Object Hierarchy

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes



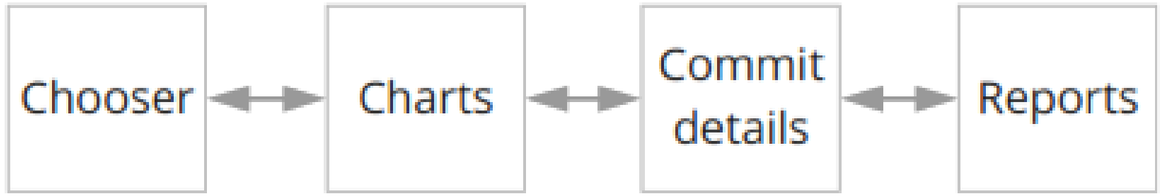
# Proposed Interaction Model

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

# Navigation Improvements

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

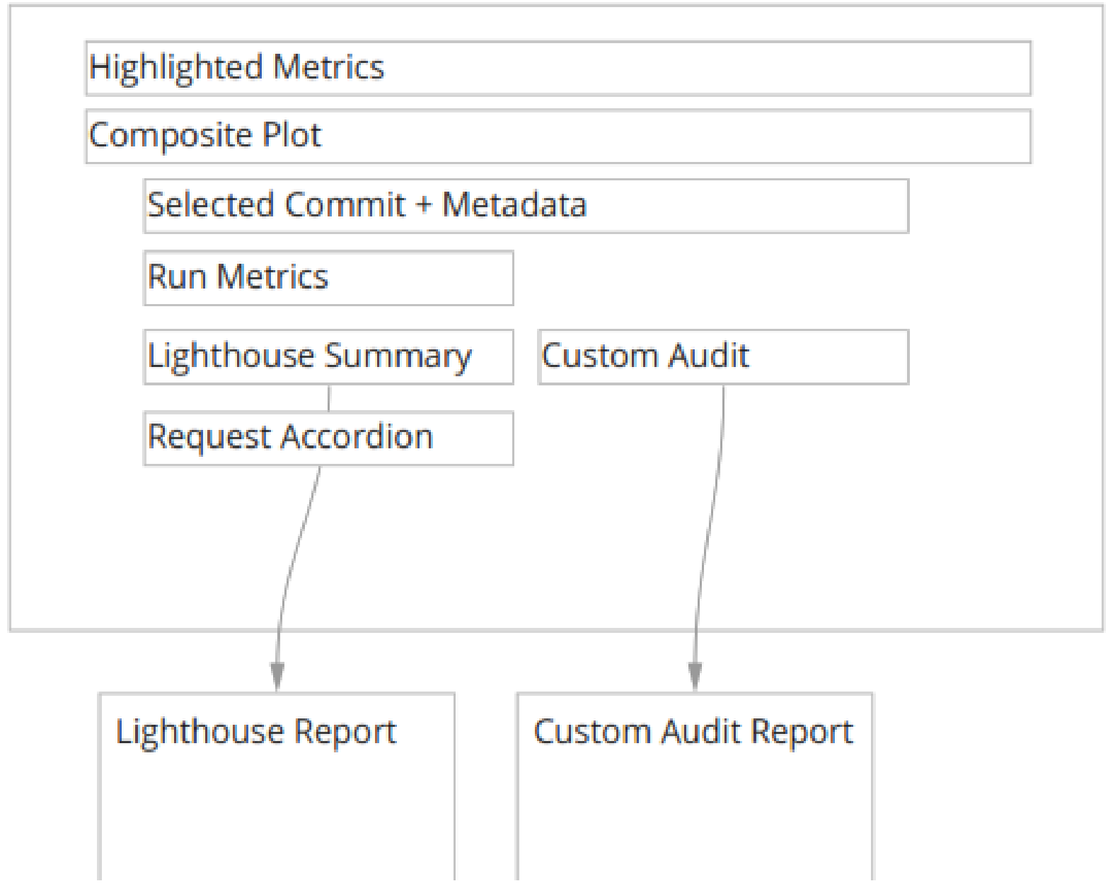
**Instead of**




**Do this**

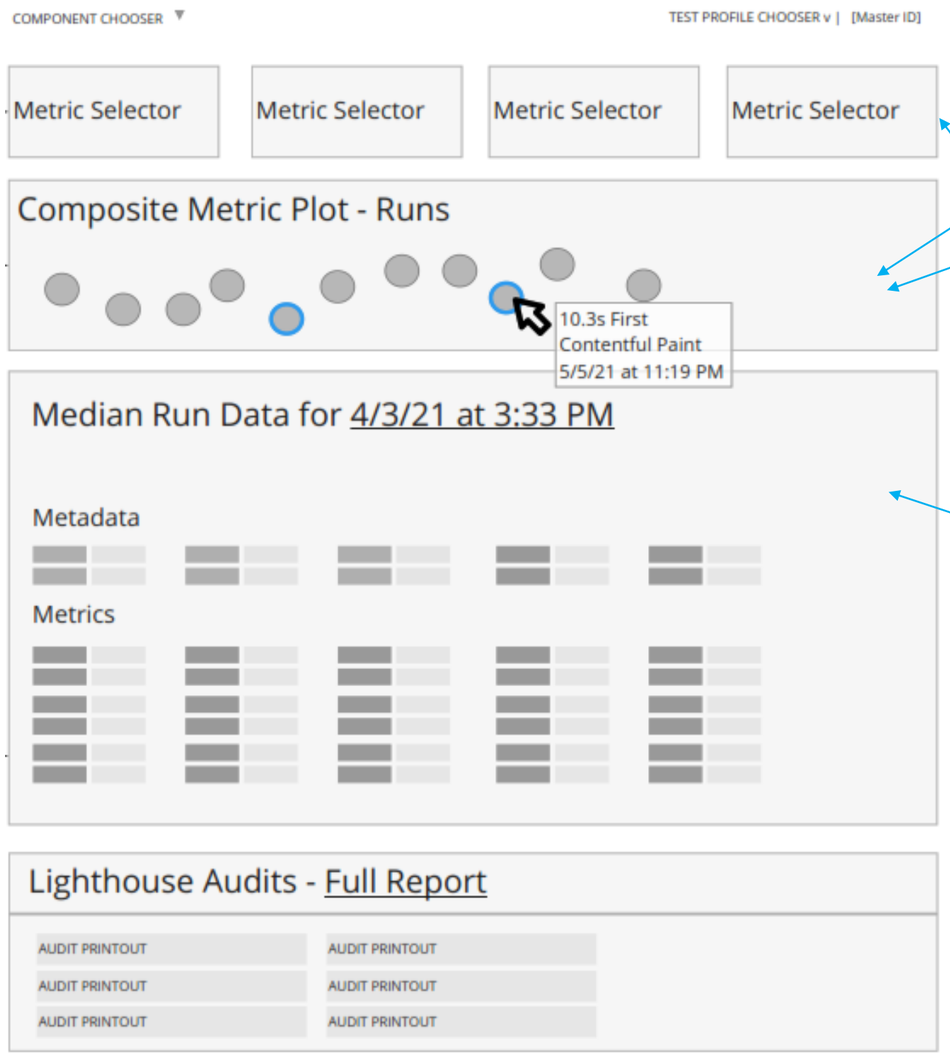
# Page Level IA

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

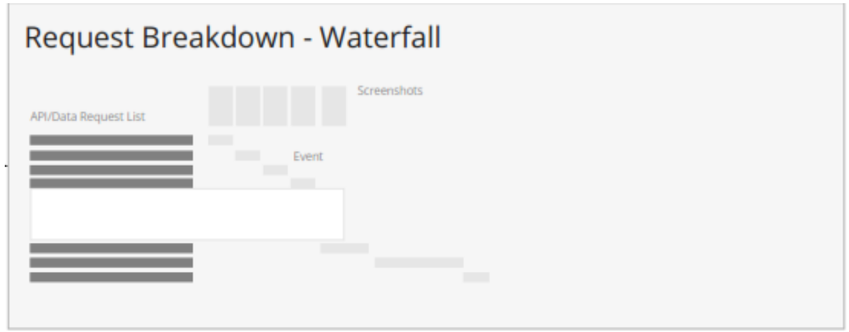


# Initial Wireframe

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

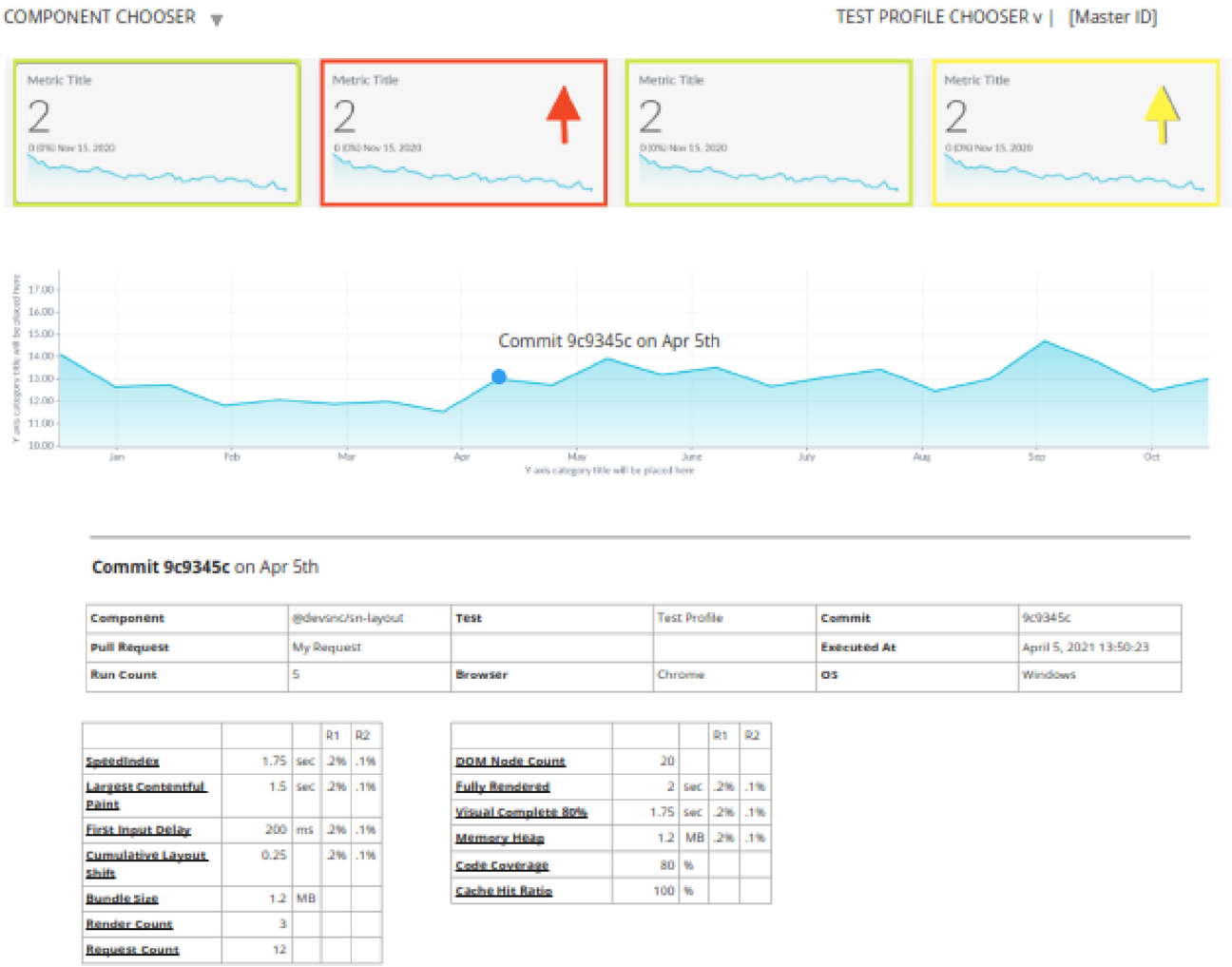



- Users want to focus on trends over time (increasing is bad)
- Need to know what is out of bounds (establish the baseline)
- Users seemed to specifically want to focus on one metric, not see all of them when the drilled in
- Users care more about the *median* data, rather than individual runs (summary & progressive disclosure)



# Wireframes with visuals

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

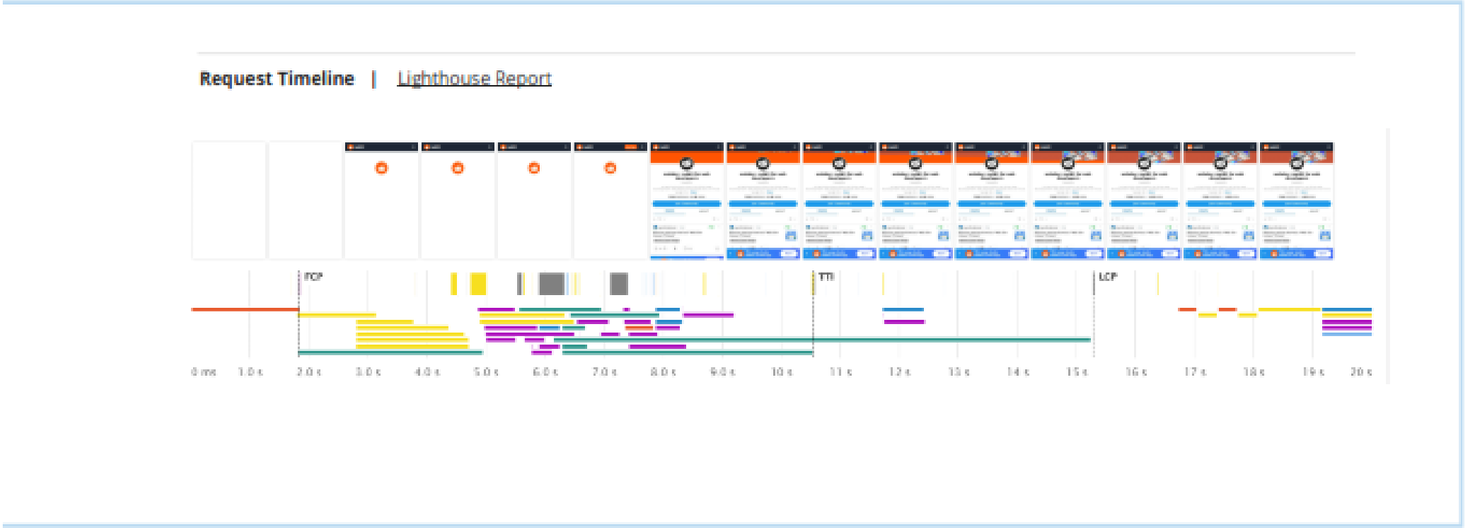


# Wireframes with visuals

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes

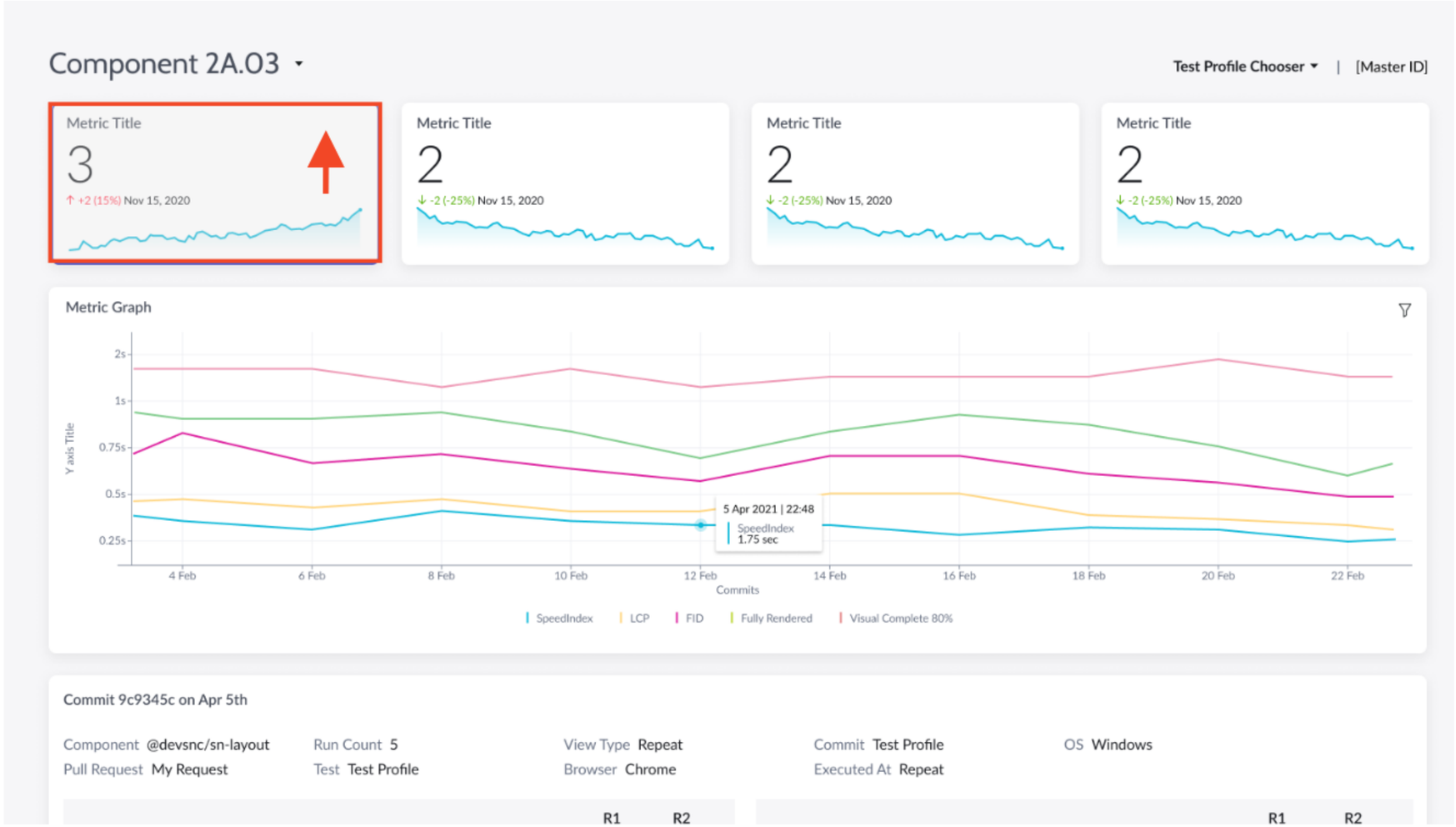
| | | | | |
|---|---|---|---|---|
| First Input Delay | 200 | ms | .2% | .1% |
| Cumulative Layout Shift | 0.25 | | .2% | .1% |
| Bundle Size | 1.2 | MB | | |
| Render Count | 3 | | | |
| Request Count | 12 | | | |

| | | | | |
|---|---|---|---|---|
| Memory Heap | 1.2 | MB | .2% | .1% |
| Code Coverage | 80 | % | | |
| Cache Hit Ratio | 100 | % | | |



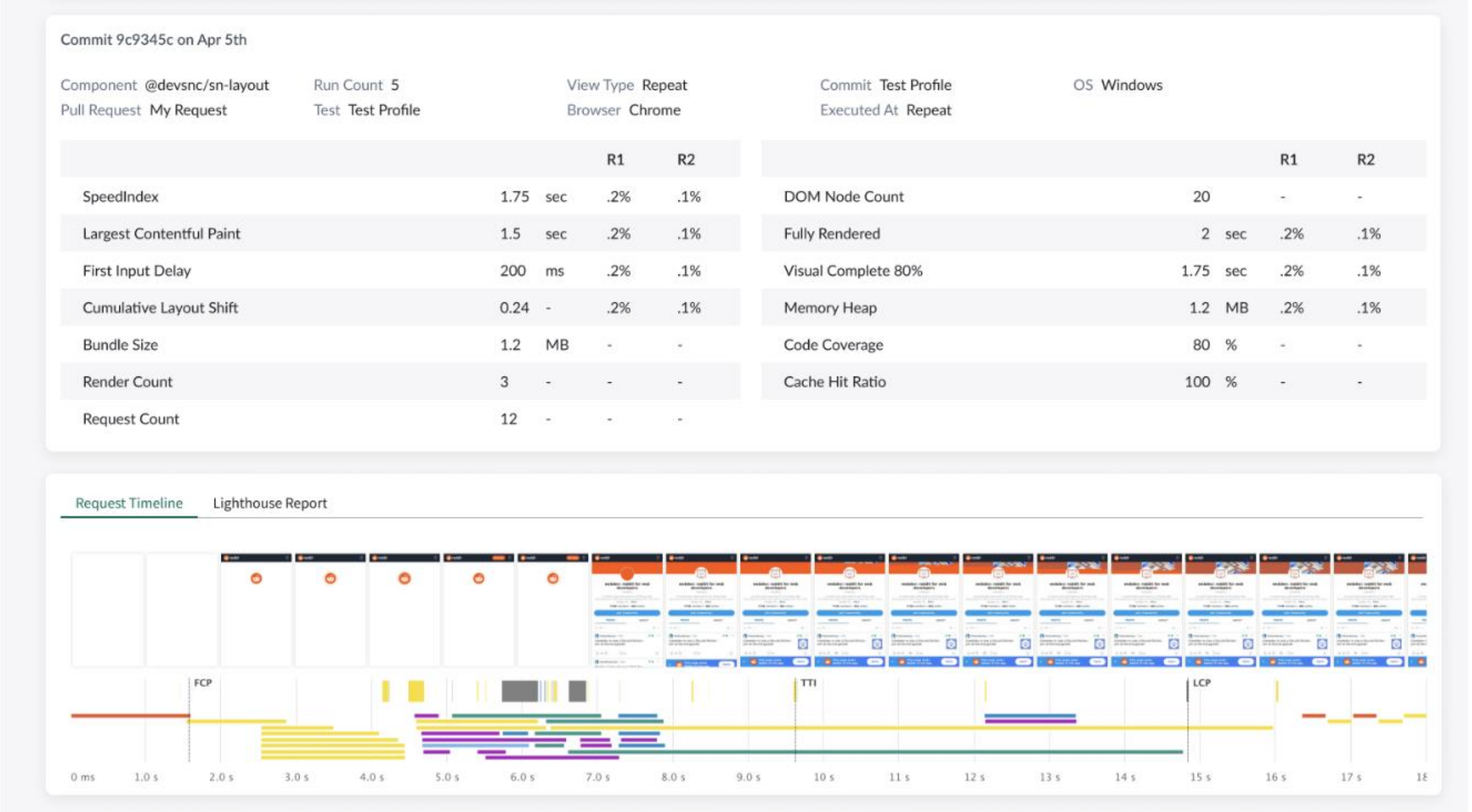
# Finalized Figma Prototype

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes



# Finalized Figma Prototype

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes



Commit 9c9345c on Apr 5th

Component @devsnc/sn-layout
Pull Request My Request
Run Count 5
Test Test Profile
View Type Repeat
Browser Chrome
Commit Test Profile
Executed At Repeat
OS Windows

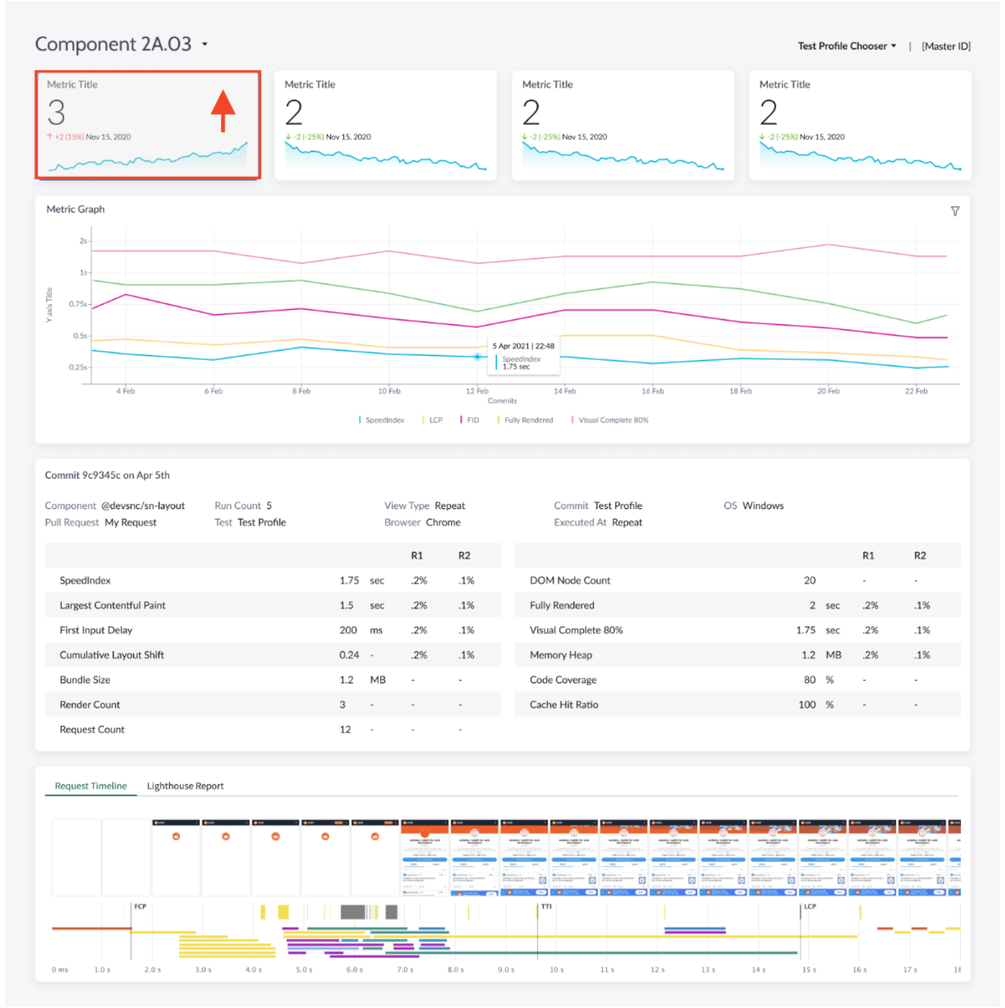
| | | | R1 | R2 |
|---|---|---|---|---|
| SpeedIndex | 1.75 | sec | .2% | .1% |
| Largest Contentful Paint | 1.5 | sec | .2% | .1% |
| First Input Delay | 200 | ms | .2% | .1% |
| Cumulative Layout Shift | 0.24 | - | .2% | .1% |
| Bundle Size | 1.2 | MB | - | - |
| Render Count | 3 | - | - | - |
| Request Count | 12 | - | - | - |

| | | | R1 | R2 |
|---|---|---|---|---|
| DOM Node Count | 20 | | - | - |
| Fully Rendered | 2 | sec | .2% | .1% |
| Visual Complete 80% | 1.75 | sec | .2% | .1% |
| Memory Heap | 1.2 | MB | .2% | .1% |
| Code Coverage | 80 | % | - | - |
| Cache Hit Ratio | 100 | % | - | - |

Request Timeline | Lighthouse Report

# Engineering Specs

1. Goals & Analysis
2. User Research
3. Design Goals
4. Tasks
5. Objects & Interactions
6. IA & Architecture
7. Wireframes
8. High-Fidelity Prototypes




.

Page Level

a. Show last component viewed upon page entry - save state such
b. Show recently-used components in the list

1. **Component chooser** - dropdown component that allows for k
2. **Test Profile chooser** - Select an available test profile
3. **Master branch** ID, Branch Name, Commit ID?
4. **Metric Chooser** - use a carousel control if there are more thar
    a. **Metric Tile control**: Metric Name, latest value, directior previous commit, trend line plotting all the individual da
    b. Selection: user clicks on one metric tile to update the ma
        i. Default is the leftmost metric (should be a compo single value)
5. **Graph** - Dependent pane based on selected metric tile control
    a. Graph can be compound or single lined - for compound,
    b. Rollover interaction: A dot or vertical set of dots shows t
        i. Tooltip: The tooltip displayed will be of all the data
    c. Click interaction: User can click anywhere in a vertical lin
        i. The area immediately below the graph is depende
6. **Commit metadata** - Show high-level metadata that describes
7. **Commit metrics** - Show an exhaustive list of metric in a two-c
    a. Layout of columns: Metric Name, value, unit, repeat 1 pe
8. **Subtab mechanism** - Request Timeline and Lighthouse Repor
9. **Request Timeline**
    a. Screenshot sequence - mapped to each sequential times
    b. Data/API request - plot these out as the requests were r timeline, and later ones are added on top as a stack
        i. Requests can cross timeslices
    c. Metric events - plot a metric when completion happens
    d. Interaction - user can click and hold to scroll the timeline
    e.


## Project and Goal

One of the Sales Cloud projects I was tasked with by the VP of Product Management was to design a proof of concept for Sales Pipeline Management, but first I needed to understand the user, the Field Sales Representative.

## Activities

- Worked with PM and senior Oracle Field Sales Reps to collect needs, tasks, questions, and scenarios
- User research, concepting, wireframing
- Designed initial concepts

## Outcomes

- Since we identified user needs beyond pipeline management, I expanded the scope to include general field sales use cases
- I was able to get buy-in from management that this was a gap for Sales Cloud

# Develop a UX concept around Field Sales

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

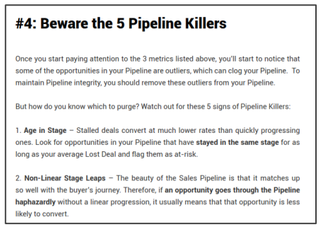
5: Architecture

6: Concepts

7: Outcomes

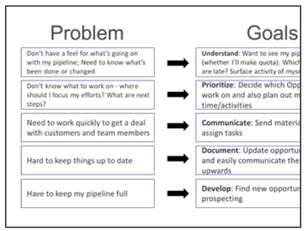
Need to uncover use cases centered around field sales

## How do we get there?




**Understand the domain**


**Empathize with the user**




**Articulate Problems, Goals, & Solutions**


**Author a proof-of-concept**

# Research the Domain

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

**Get to know the typical concepts and problems in this space.**

## #4: Beware the 5 Pipeline Killers

Once you start paying attention to the 3 metrics listed above, you'll start to notice that some of the opportunities in your Pipeline are outliers, which can clog your Pipeline. To maintain Pipeline integrity, you should remove these outliers from your Pipeline.

But how do you know which to purge? Watch out for these 5 signs of Pipeline Killers:

1. **Age in Stage** – Stalled deals convert at much lower rates than quickly progressing ones. Look for opportunities in your Pipeline that have **stayed in the same stage** for as long as your average Lost Deal and flag them as at-risk.

2. **Non-Linear Stage Leaps** – The beauty of the Sales Pipeline is that it matches up so well with the buyer's journey. Therefore, if **an opportunity goes through the Pipeline haphazardly** without a linear progression, it usually means that that opportunity is less likely to convert.

3. **Opportunity Size** – An opportunity's size dictates a lot about how it will act and how

# Field Sales Rep Interviews

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

- Oracle field sales reps interviewed on multiple occasions + iterations
- CX, CRM space, Health, Upmarket
- Tend to have less than 10 opportunities at a time that they’re working

**Getting a variety of interviewees helps identify universal issues**

# Users, In Their Own Words….

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

Bill V
people.oracle.com/@b
CRM Application Sales Manager
Beachwood OH, US · 12:58 AM Thu

**“Need a perspective on what's going on with an account”**

Annie K
people.oracle.com/@a
Application Sales Manager
Atlanta GA, US · 1:00 AM Thu

**“My contacts can't be visible to my sales team (*Inside Sales reps*) will pick up contact and activity and start calling into that account - throws a wrench into that relationship!”**

Cris P
people.oracle.com/@c
Application Sales Manager
Chicago IL, US · 12:07 AM Thu

*What would make you 10x more effective?*

**"Text or instant thing on phone that says ‘Do you want document that?’”**

**I asked sales reps macro questions like: “What would make you 10x more effective?”**

**Thanks to using higher-level questions, we were able to identify unanticipated needs like *‘keep them from stealing my deals’***

# Sales Rep's Own Questions

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

- **(When your boss asks) "How are you getting to your number this quarter?"**
- **"What are other deals, assets, purchases, or trouble tickets for this customer?"**
- **"What do I do now that I found out a competitor is on this deal too?"**
  - **"How do I put together a Kill Sheet for a particular competitor?"**
- **"The customer's CIO is outraged on a P1 issue open for an entire month"**
- **"Sales Manager needs you to bring in one or two deals in the next few days to make the team's numbers this quarter"**

**Questions reveal needs as well as set the context for how Field Sales reps operate.**

# Field Sales Rep Persona

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

## DANIEL RICHARDSON

Field Sales Rep

AGE: 41

LOCATION: San Francisco, CA

EXPERIENCE: 11 years

WORK PATTERN: Road Warrior

| Job Roles by Area | Persona | Typical Tasks | Critical Tasks | Focal Task(s) |
|---|---|---|---|---|
| **Sales Representative** | Daniel Richardson | • Prep for Client Meetings<br>• Conduct Post-Meeting Activities<br>• Communicate w/Sales Team and Management<br>• Networking<br>• Create proposals<br>• Schedule meetings and follow-ups | • Meet with stakeholders<br>• Close Deals<br>• Sign Contracts | • Maintain Customer Relationships<br>• Create trust<br>• Help customer define the problem |

**Break tasks into TYPICAL, CRITICAL, and FOCAL types to understand user needs**

# Eye-opening Findings

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

- ⚠ Our biggest competitor is Outlook
- ⚠ Oracle isn’t even using its own software
  - Outlook integration isn't set up
- ⚠ Several custom solutions have been made because the software doesn't provide the necessary solutions

**There were some surprising lessons-learned.**

# Field Sales Journey

1: Goals & Process

2: User Interviews

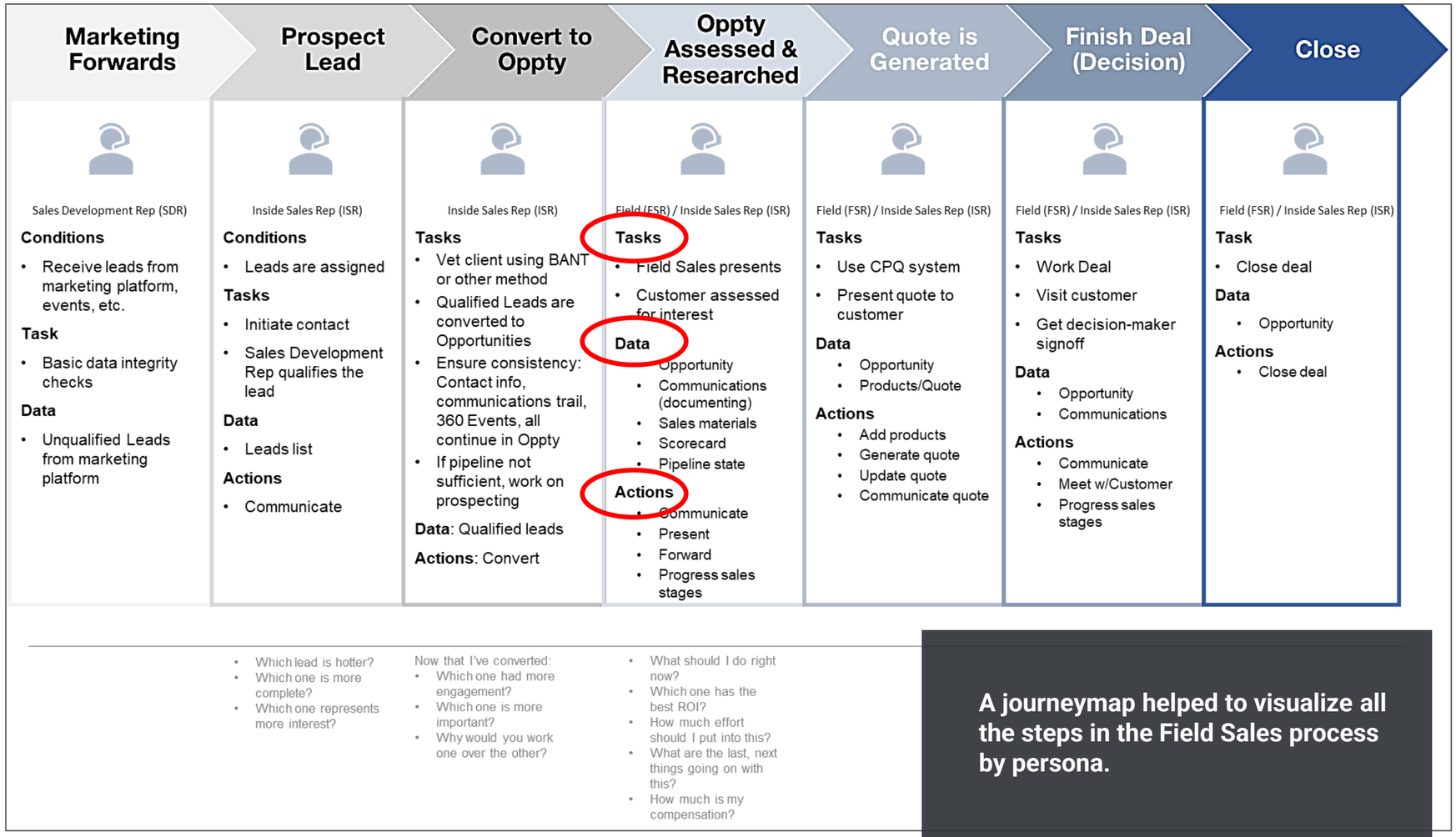
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**A journeymap helped to visualize all the steps in the Field Sales process by persona.**

# Problems and Goals for an FSR

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

| Main Problems | Goals | |
|---|---|---|
| Don’t have a feel for what's going on with my pipeline; Need to know what’s been done or changed | **Understand**: Want to see my pipeline health stats (whether I’ll make quota). Which opportunities are late? Surface activity of myself and team… | Understand 1 |
| Don't know what to work on - where should I focus my efforts? What are next steps? | **Prioritize**: Decide which Opportunities to work on and also plan out my time/activities | Prioritize 2 |
| Need to work quickly to get a deal with customers and team members | **Communicate**: Send materials, follow-up, assign tasks | Communicate 3 |
| Hard to keep things up to date | **Document**: Update opportunities quickly and easily communicate their status upwards | Document 4 |
| Have to keep my pipeline full | **Develop**: Find new opportunities by prospecting | Develop 5 |

# Key insight: There's a Progression to This

1: Goals & Process

2: User Interviews

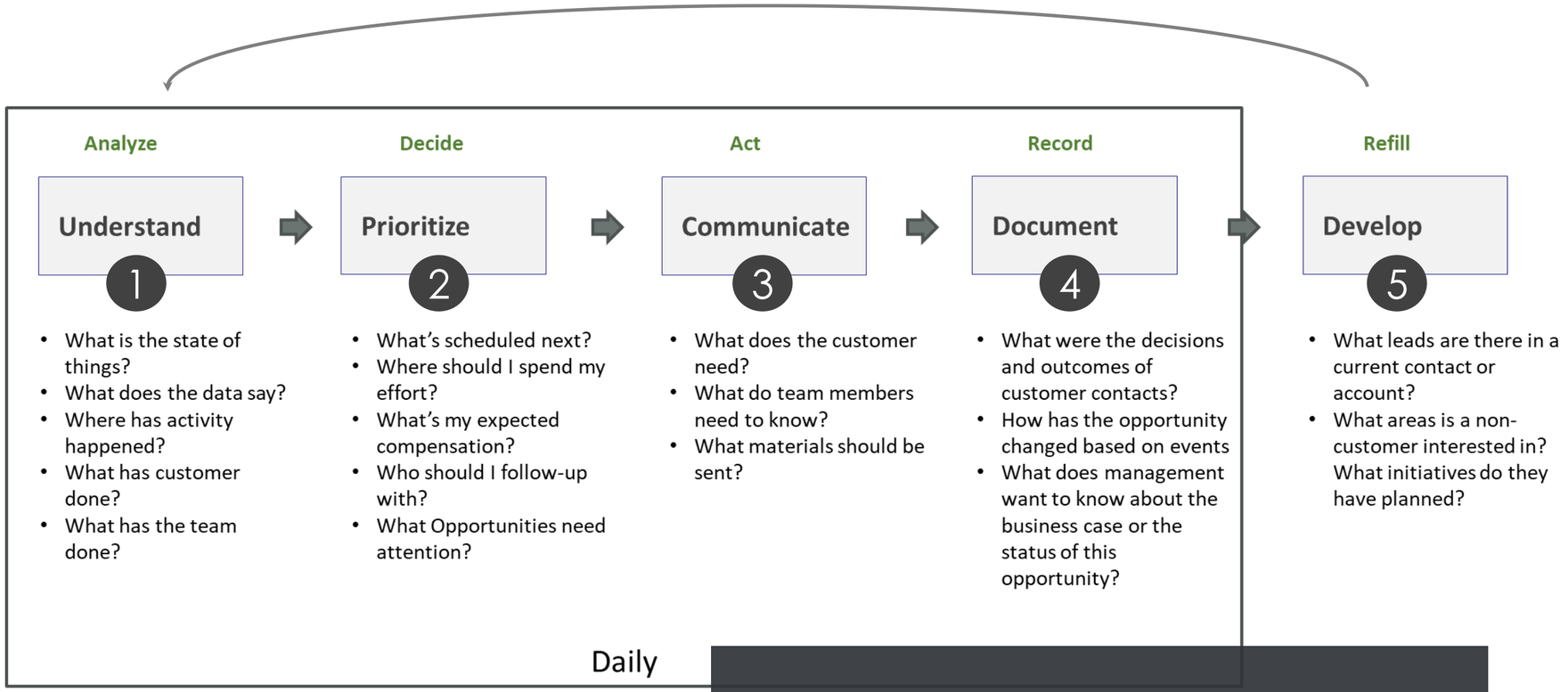
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**Another key insight: The goals fit together in a coherent process flow.**

# Solutions articulate tasks and data

1: Goals & Process

2: User Interviews

3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes

| Goal | Solutions: Tasks | Solutions: Data |
|---|---|---|
| **Understand** | Browse the data; Monitor the state of contacts, accounts, team activities | KPI's, Stats, Visualizations, and Timeline<br>Compare Oppty vs. Avg. Won Deal or Avg. Win sales cycle; Timeline (Macro), my/team activities to date, updates to contacts and accounts (Micro) |
| **Prioritize** | Look at compensation, and the most promising Opportunities; Figure out my next steps, and what needs to be done | W2 compensation per deal, (win % x deal size); Recommendations, Day/Quarter desired activities, follow-ups, tasks |
| **Communicate** | Send materials, follow-up, assign tasks | Task content, sales materials, meeting agendas |
| **Document** | Post-meeting data entry, build biz case for mgmt. | Call Logs, Meeting summaries, Scorecards, Actions to Close |
| **Develop** | Get leads from contacts, keep up-to-date with potential customers, look at leads per account | New Lead objects, Account leads |

**By pairing goals with solutions, tasks and data objects can be identified that users will need**

# Pairing Goals and Use Cases

1: Goals & Process

2: User Interviews

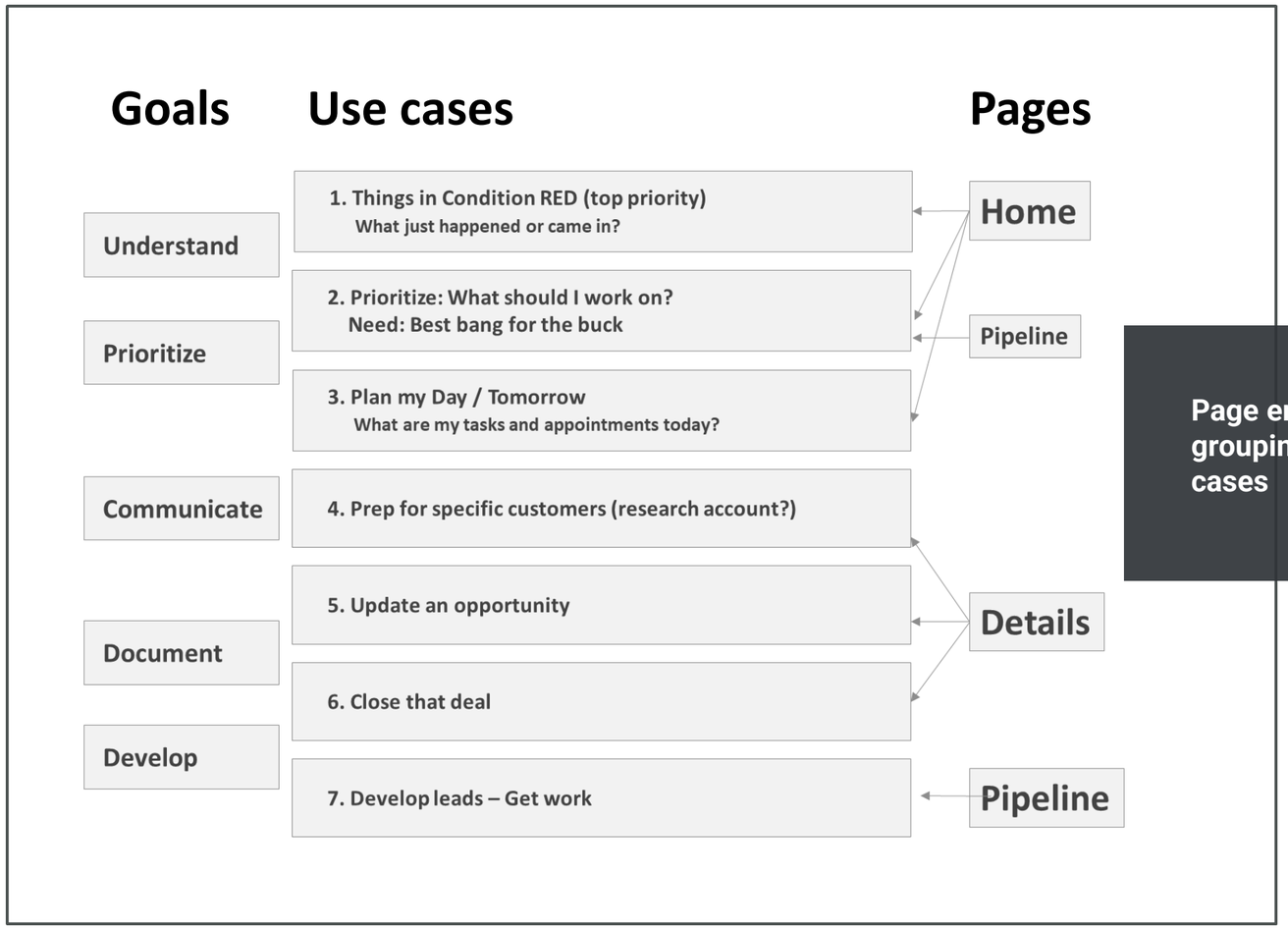
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**Page entities emerge through grouping the appropriate use cases**

# Page Rationales

1: Goals & Process

2: User Interviews

3. Analysis

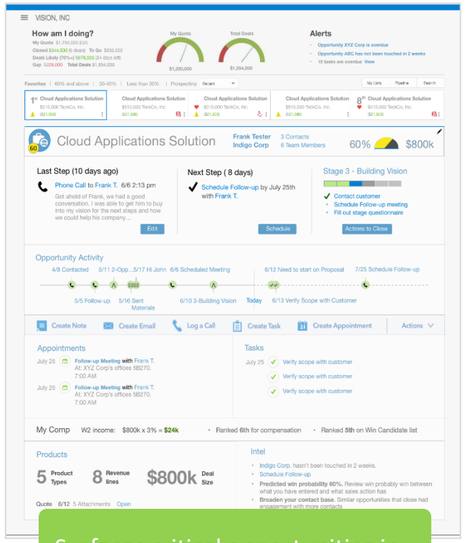
4: Problems

5: Architecture

6: Concepts

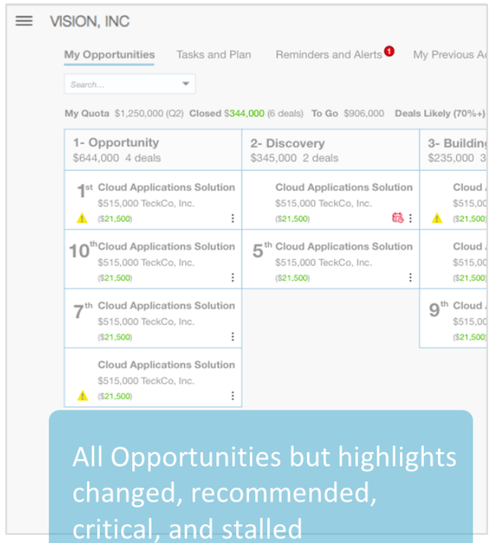
7: Outcomes

## Home Page



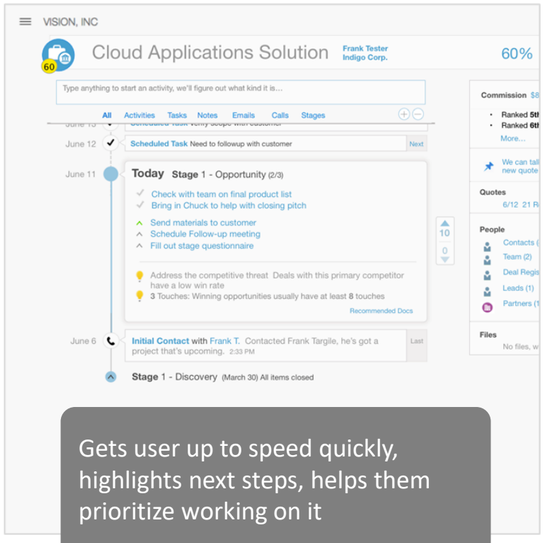

Surfaces critical opportunities in a fast-switch interaction mechanism

## Pipeline (Kanban)




All Opportunities but highlights changed, recommended, critical, and stalled
(Filter for favorites or other dimensions)

## Detailed Opportunity




Gets user up to speed quickly, highlights next steps, helps them prioritize working on it

General > Specific

# New Home Page Concept

1: Goals & Process

2: User Interviews

3. Analysis

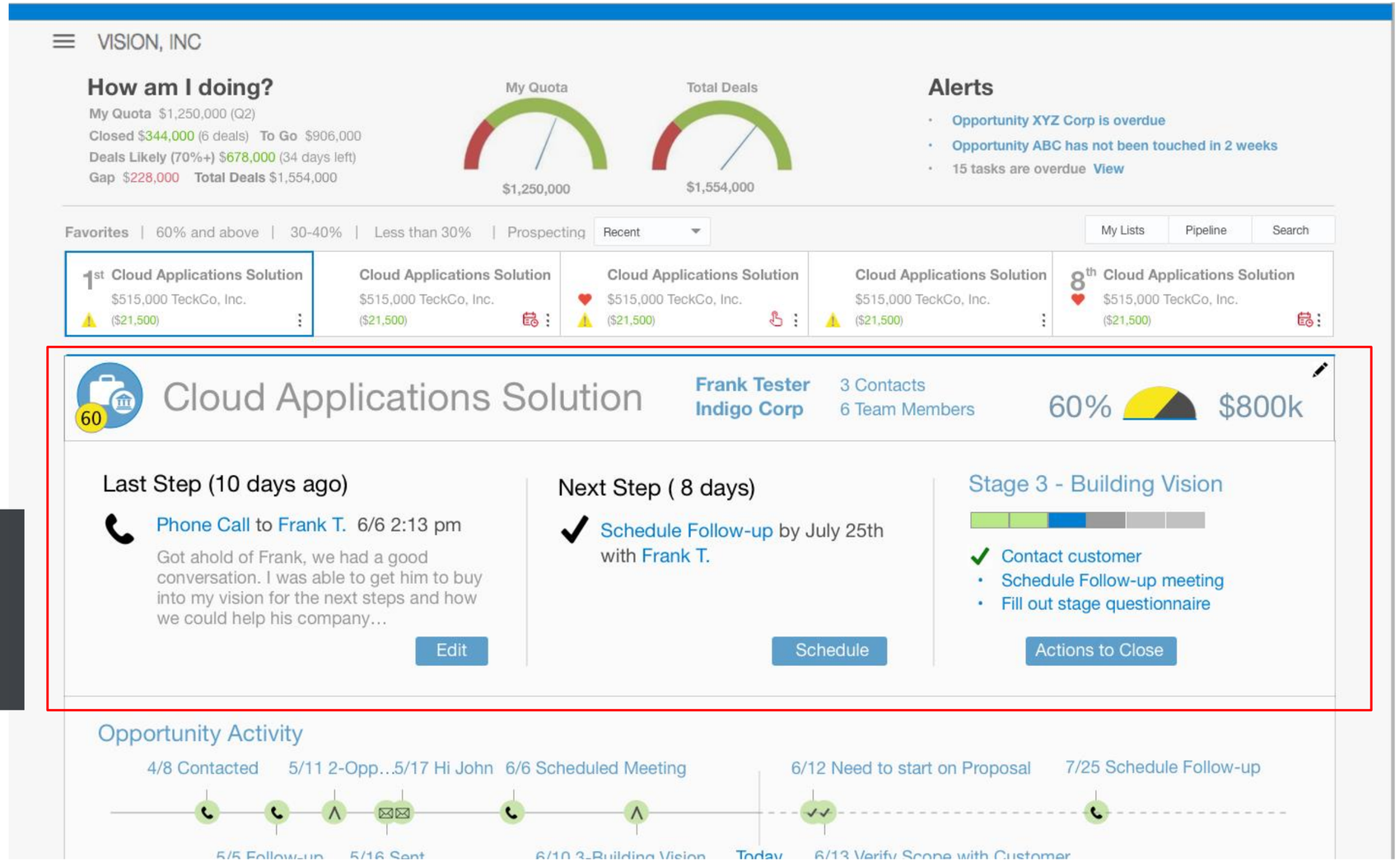
4: Problems

5: Architecture

6: Concepts

7: Outcomes

**Everything to get up to speed on this opportunity can be seen in seconds**



# Pipeline Dashboard Wireframe

1: Goals & Process

2: User Interviews

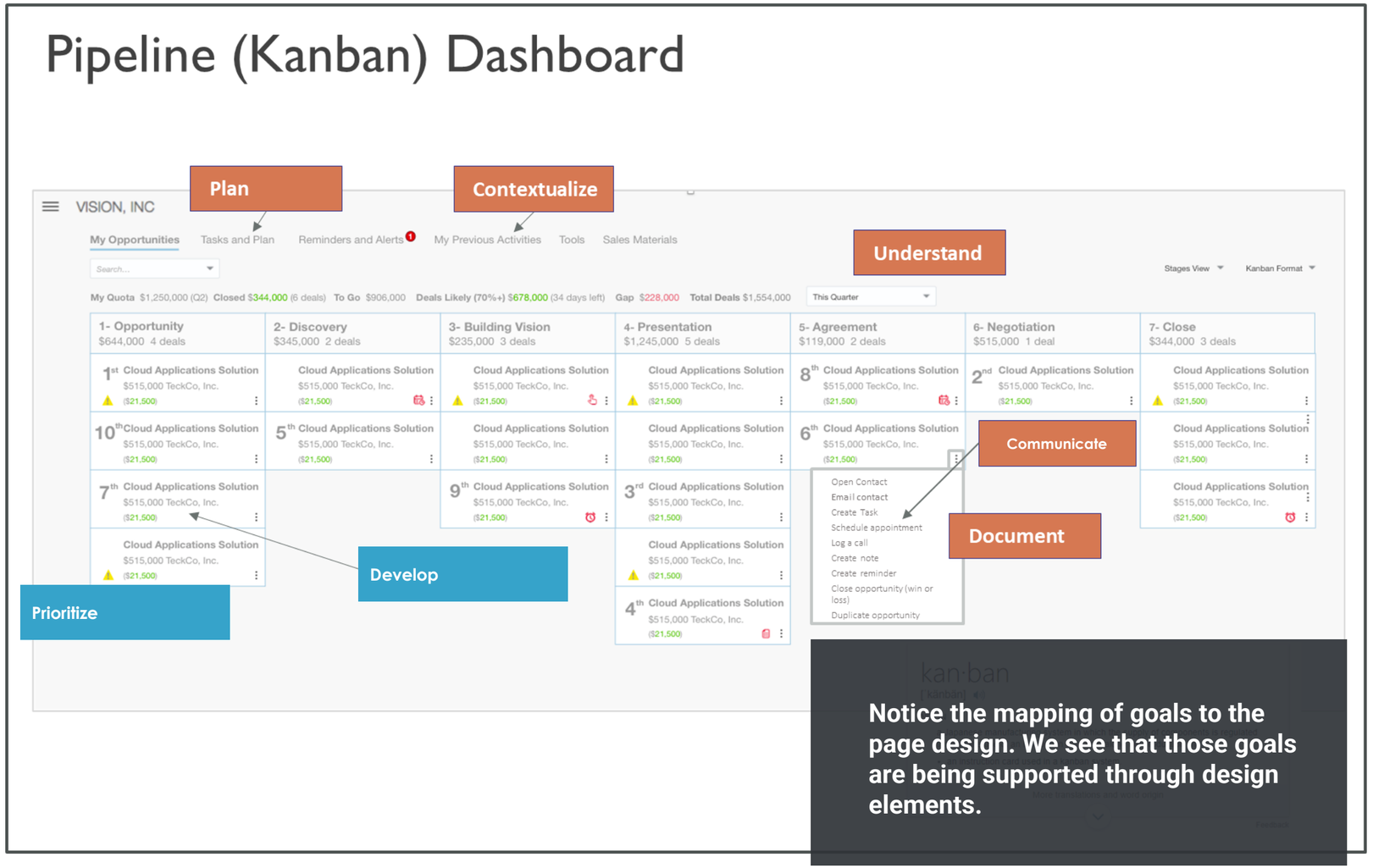
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes



# Oppty Details Concept 1: Timeline

1: Goals & Process

2: User Interviews

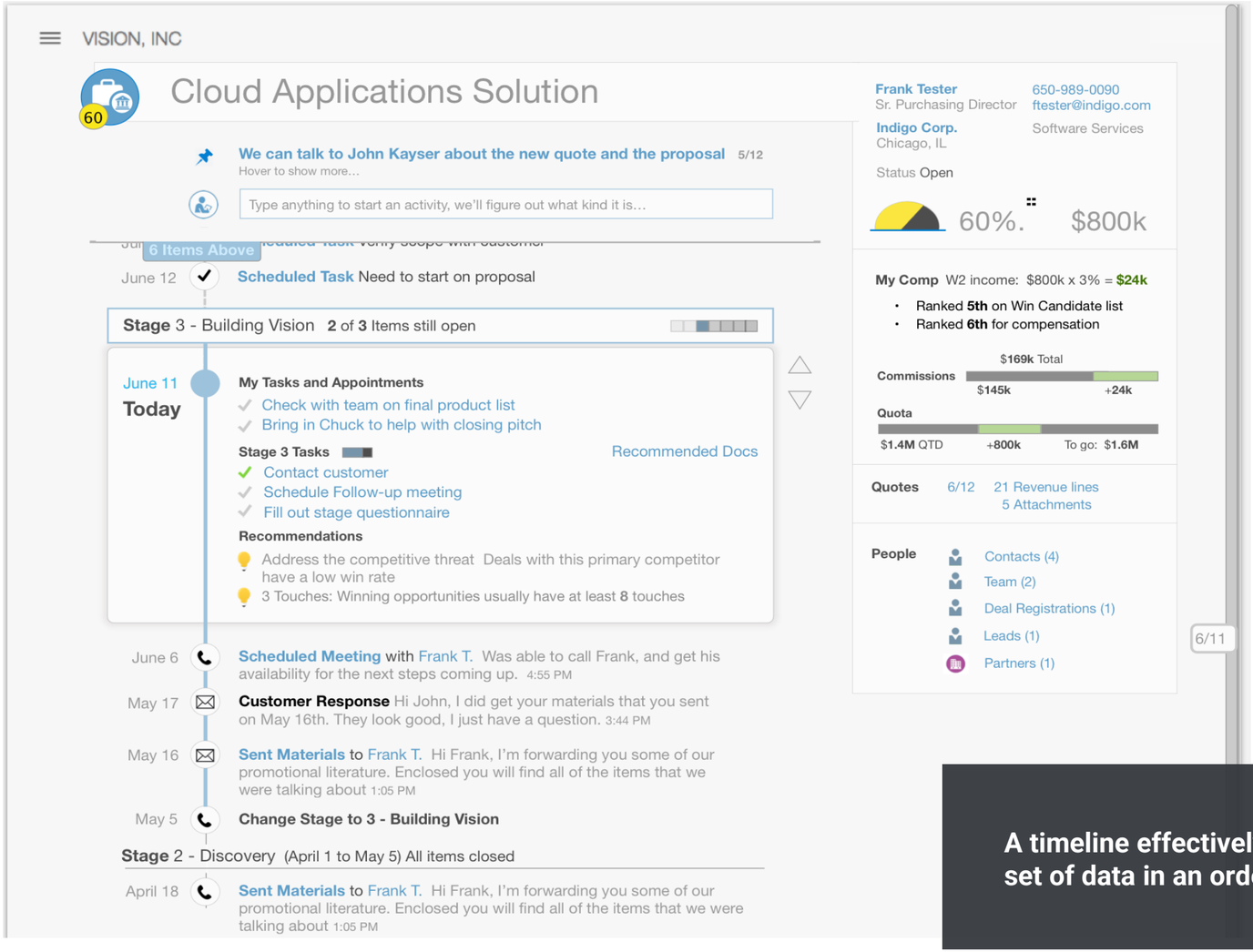
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**A timeline effectively handles a large set of data in an ordered structure.**

# Oppty Details Concept 2: Grid Version

1: Goals & Process

2: User Interviews

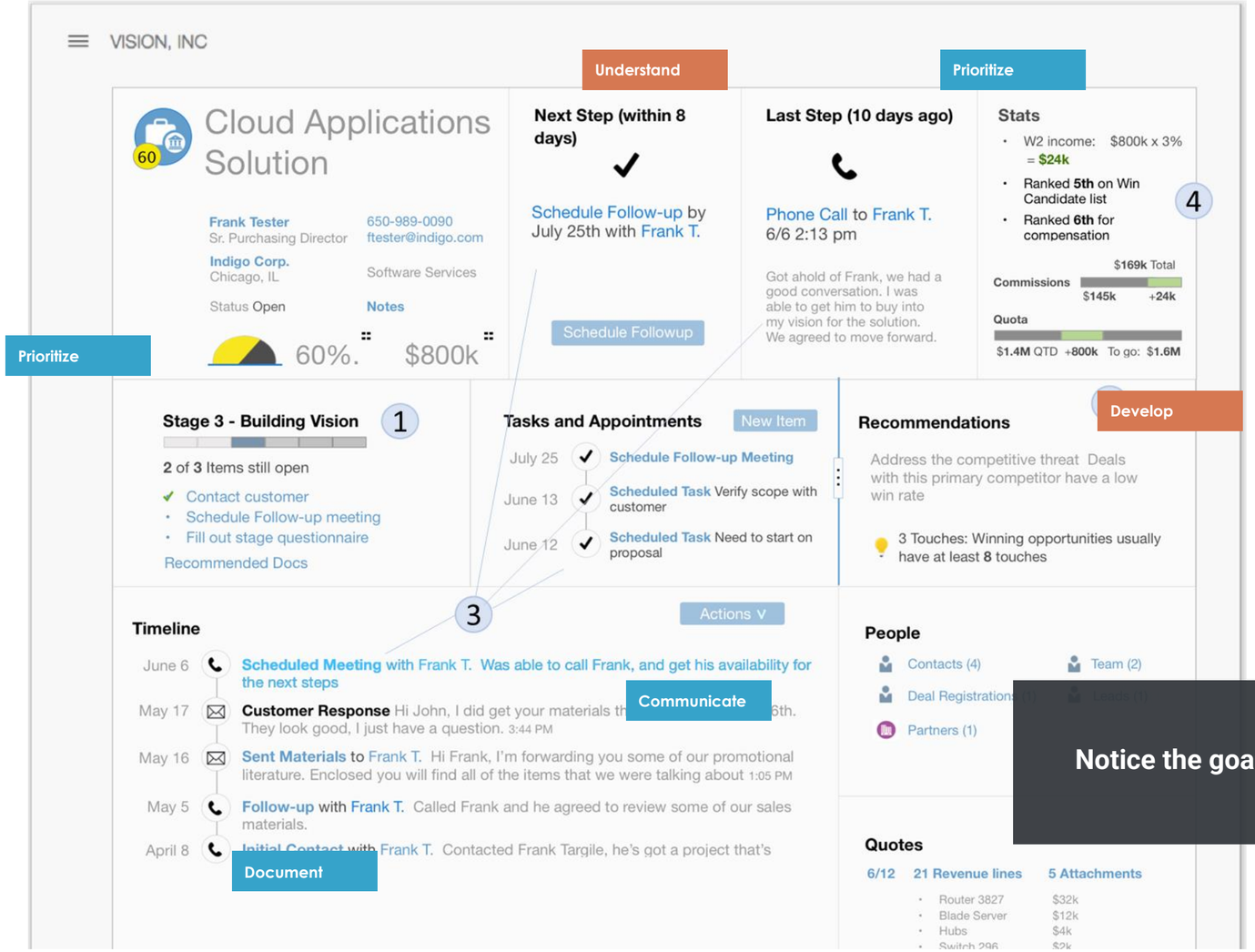
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**Notice the goal mapping here as well.**

# Oppty Details Concept 3: Guided

1: Goals & Process

2: User Interviews

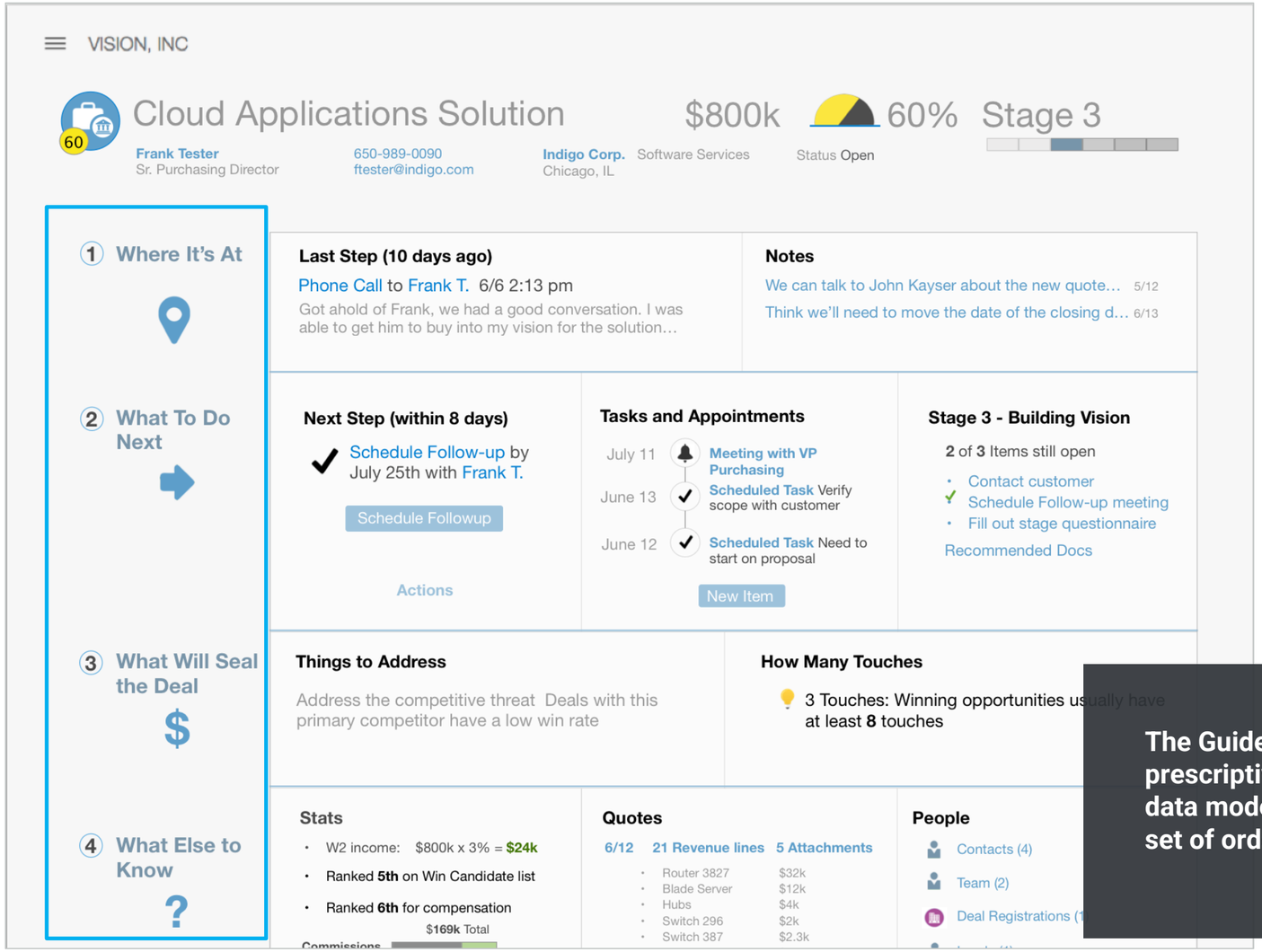
3. Analysis

4: Problems

5: Architecture

6: Concepts

7: Outcomes




**The Guided concept is prescriptive and chunks the data model into a scannable set of ordered sections.**

# Outcomes

1: Goals & Process

2: User Interviews

3. Analysis

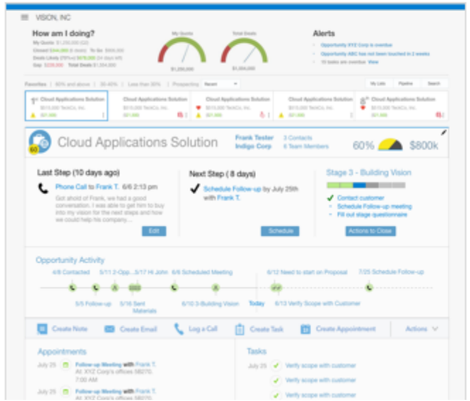
4: Problems

5: Architecture

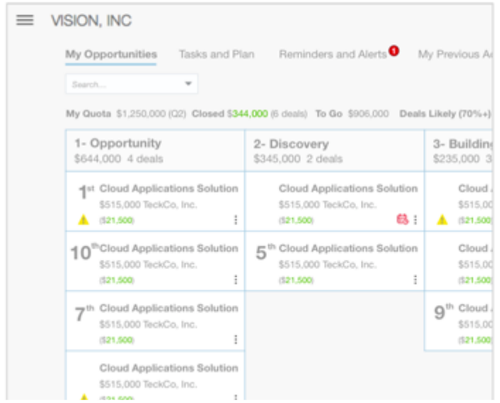
6: Concepts

7: Outcomes

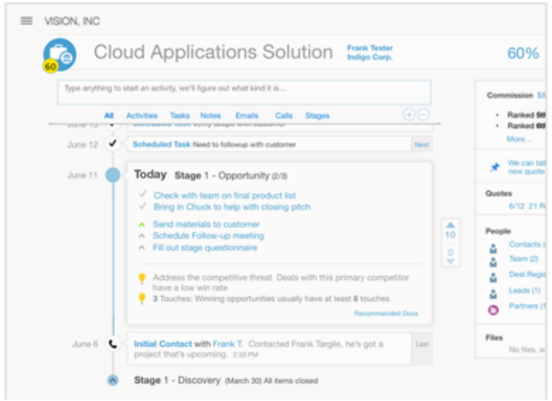
## Home Page




## Pipeline (Kanban)



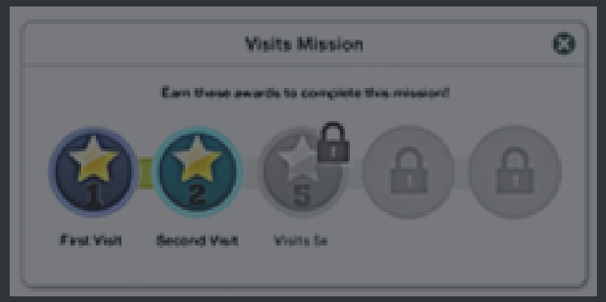

## Detailed Opportunity




**The POC effort helped convince Oracle's UX leadership that Field Sales and Pipeline management use cases needed to be addressed by our Sales Cloud product.**

Badgeville
The Behavior Platform™

# Badgeville Platform Redesign




## About the Company

Badgeville was an enterprise gamification startup dedicated to helping other companies leverage game mechanics to incentivize user behaviors.

## Project and Goal

I left Apple to start a UX practice at Badgeville, reporting to the VP of Product Management. The founder and CEO tasked me with leading a ‘radical revision’ to the level of usability of the product which had not had formal UX design practices applied.

## Activities

- UX design practices: user research, task flows, wireframes
- Worked extensively with customers to collect needs, tasks, and other inputs
- Redesigned the platform
- Created a new VX design language
- Worked with PMs and Engineers to get the new designs implemented

# Goal: ‘Radically improve’ the usability of the platform

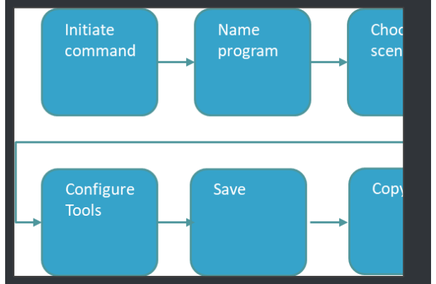
- in a matter of months, because our $70 million dollar startup is centered on this platform

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

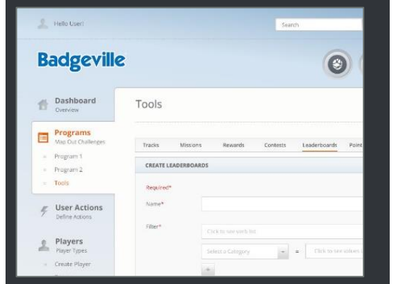
## How do we get there?


Understand the domain


Empathize with the user




Map key user flows




Propose new UX system

Badgeville
The Behavior Platform™

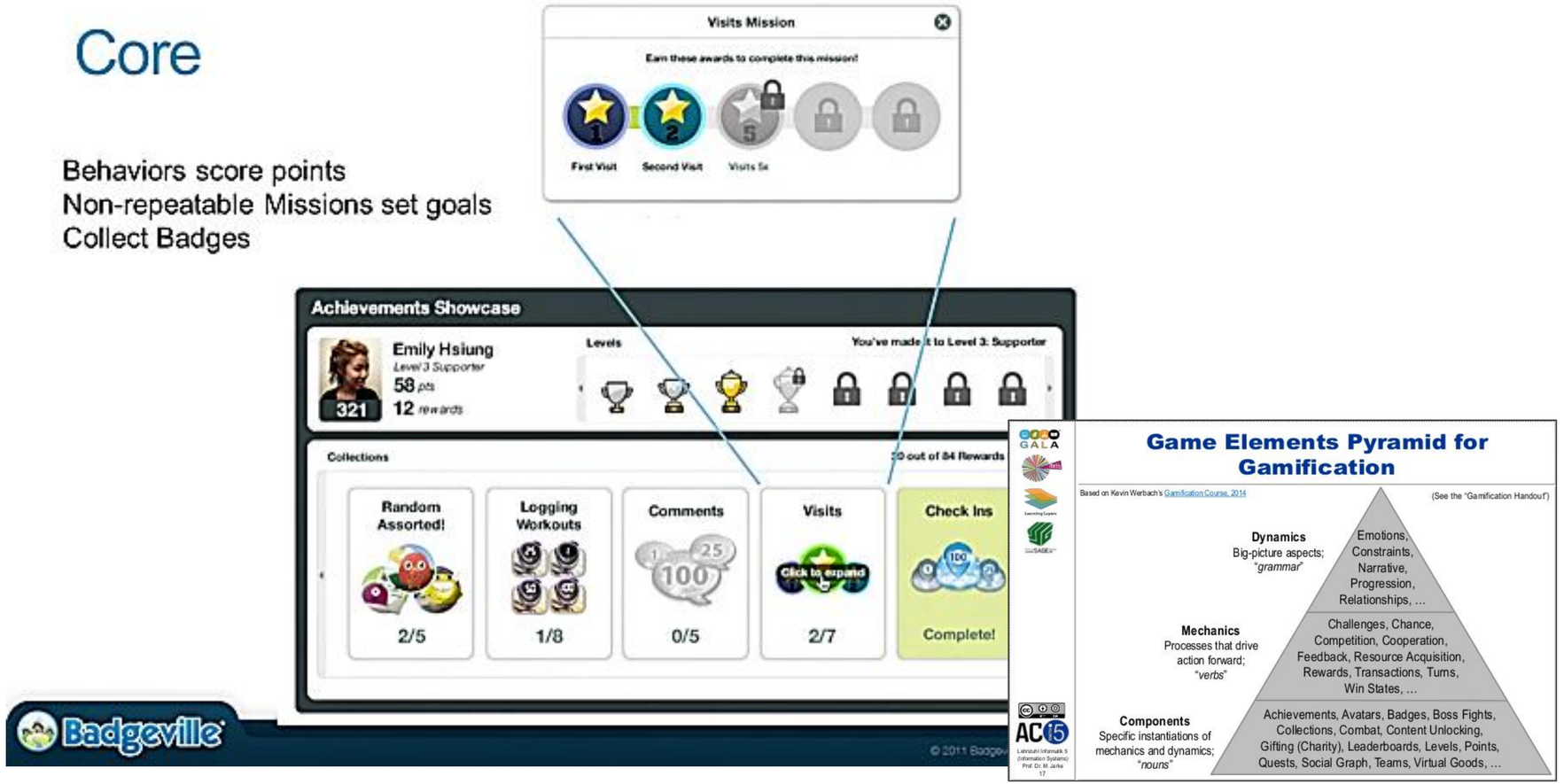
# Primary goals for the redesign project



- Increase self-service in the product
- Lower campaign producer and support costs
- Re-orient the product from a primarily developer platform to a campaign design & management one.
- Reduce the implementation time by lowering the learning curve
- Create more customer success stories

# Platform Game Mechanics

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

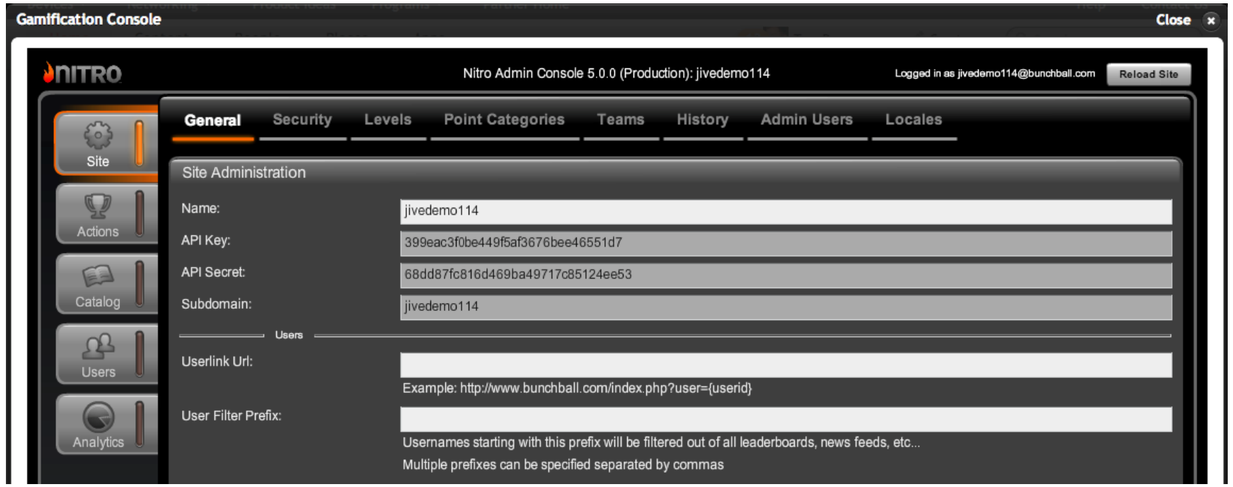


Badgeville
The Behavior Platform™

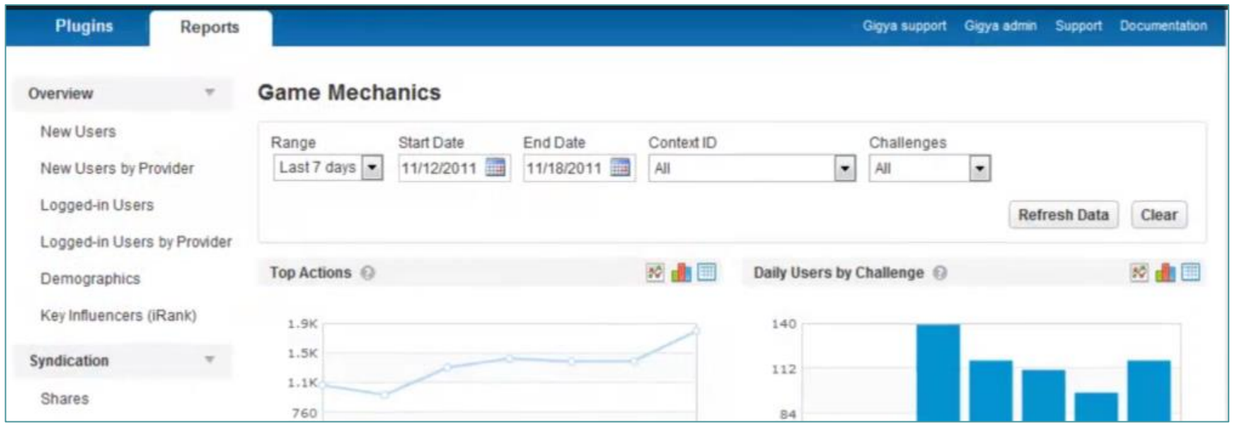
# Competitive Analysis

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

Bunchball Nitro




Gigya




**Are there competitors in this space?**

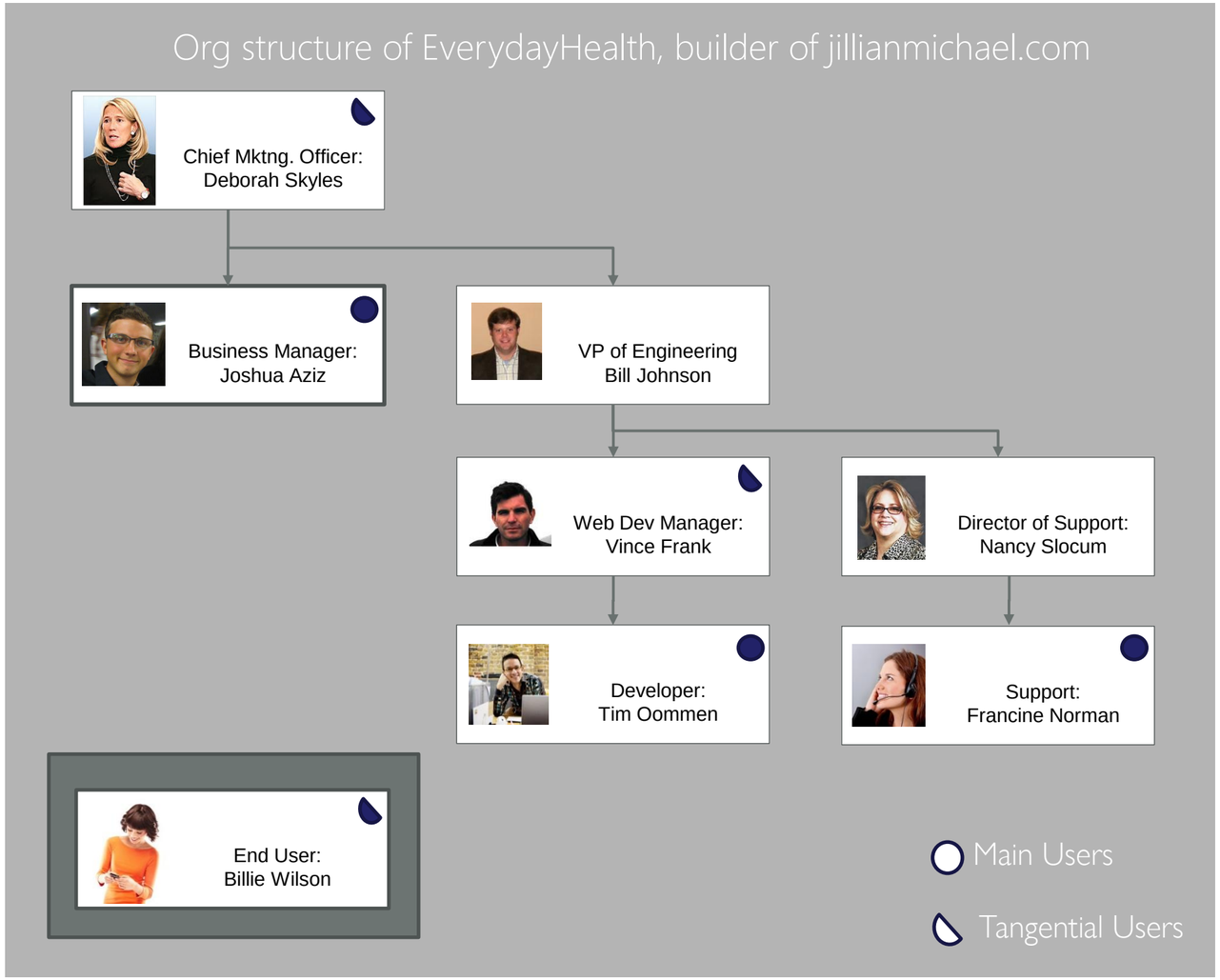
**Are they doing something interesting?**

**How are they solving the same problems?**

**What can I steal?**

# Customer Personas and Org Structure

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact




**After working with the VP of Account Management and identifying key personas, I then built an org structure to see the relationships between personas.**

# Persona-driven Scenarios



**Chief Marketing Officer: Deborah Skyles**

- Deborah wants her product or social platform manager to develop a gamification strategy and to get up and running quickly. She needs her product manager to report to her often about effectiveness and ROI.

**Business Manager: Joshua Aziz**

- Joshua needs a new campaign for Jillian Michaels' site. They want to increase retention and need to create a new site in the console.

**Developer: Tim Oommen**

- Tim gets tasked with making sure that new [illegible] configured in the BV console. He needs to have them s[illegible]correctly on the targeted Web pages.

**Developing scenarios alongside the identified personas helps contextualize the usage of your product.**

**Support: Francine Norman**

- A customer calls with questions about where their badge went. It turns out that the badge was taken away when a mission was repeated. Francine files a support ticket with Badgeville about the issue.

**End User: Billie Jo Wilson**

- Billie Jo knows is proud of her Super Shopper badge but it disappeared one day. Feeling annoyed, she called up customer support to ask for it back.

# Developing Persona Templates

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact


**It helped to have real users to build Personas around**

# Core User Task Flows



## 1. Design Game

### 1. Create Program

1. Initiate command
2. Name program
3. Choose scenario
4. Choose goal(s)
5. Review template and add/subtract from list of Tools
6. Add existing component instances
7. Configure Tools
8. Save
9. Copy code
10. Copy widgets

## 2. Monitor Program

1. Go to program or to dashboard
2. View the desired set of reports
3. Click on a report to go to full page

## 3. Manage Players

### 1. View Profile Information

a. Adjust player points, badges, rewards
b. Manage teams player is on
c. Examine player graph
d. View Player Activities

### 2. View Team

a. View Team lineup
b. Edit Team info
c. Assign rewards
d. Assign Points
e. View Team Activities
f. Examine Team Graph

c. Configure levels
    1. Set specific behaviors for levels (or all)
    2. Set count thresholds for levels

### 4.2 Create Behaviors (optional)

1. Name behavior
2. Add metadata
3. Select primary metadata pair
4. Add Points - Optional
5. Add/Create Point Systems (w/value) - Optional
6. Add scheduling & Timed reset - Optional

### 4.3 Create Leaderboard

1. Select Type (Behavior-*create points schedule*/Reward/Player/Team/Virtual Team)
2. Select behaviors, and units/metadata (if appropriate)
3. Filter teams (if appropriate)

### 4.4 Create Mission

1. Select Mission type (random or progressive)
2. Select if this is a 'Levels Mission' (if progressive)
3. Add Rewards

### 4.5 Create Reward

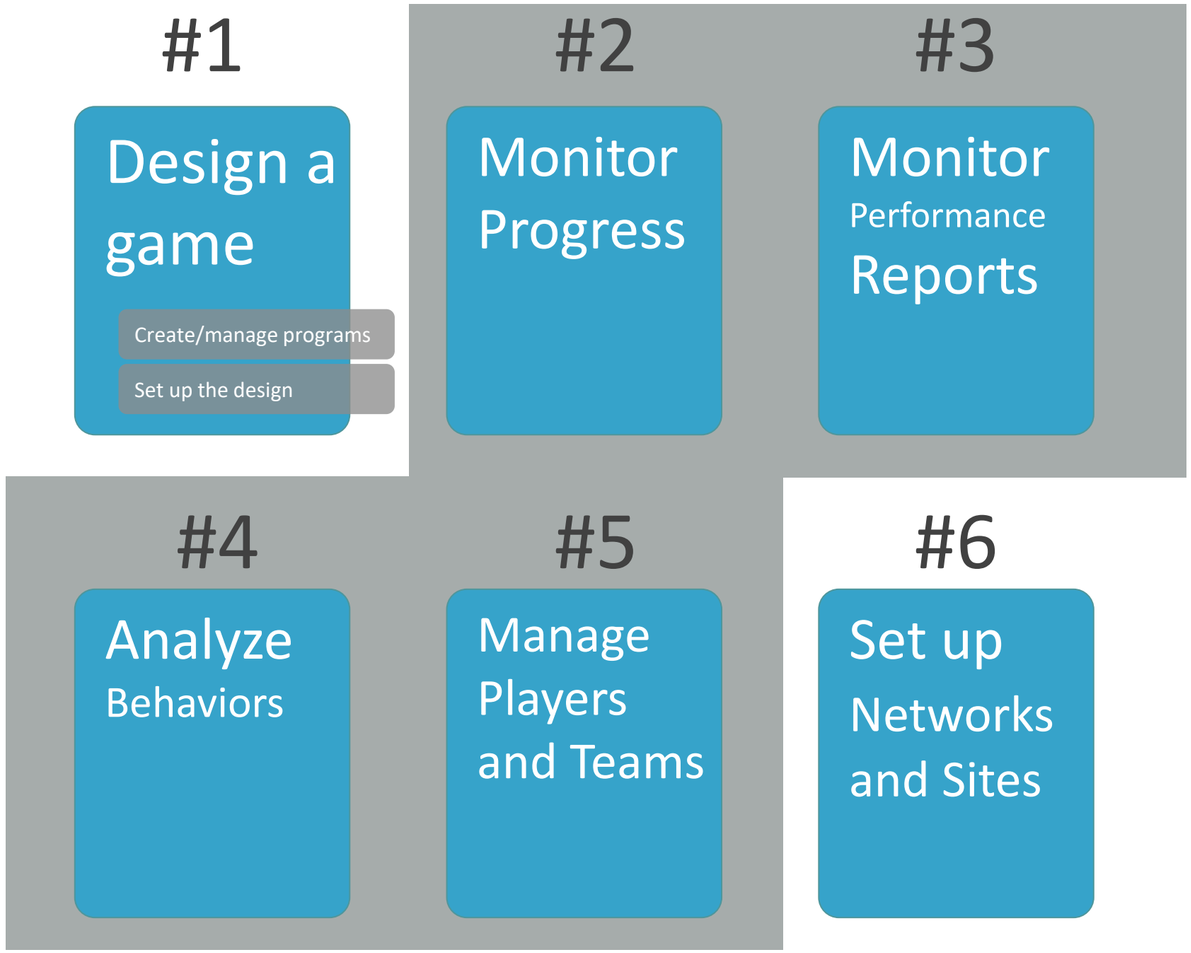
1. Select Player or Team reward
2. Add image
3. Add Points
4. Add/Create Point Systems (w/value) - Optional
5. Add scheduling & Timed reset – Optional
6. Select/Add Category Value (metadata)
7. Trigger: Behavior/Social Behavior, X times, Category and Value(s) (metadata)

### 4.6 Create Contest

1. Select Leaderboard
2. Make Active

# Core User Task Groupings

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact




**Let a business manager persona take over step #1 by making games much easier to configure. Tasks #2-5 can be accomplished by a campaign manager and #6 by an admin or developer.**

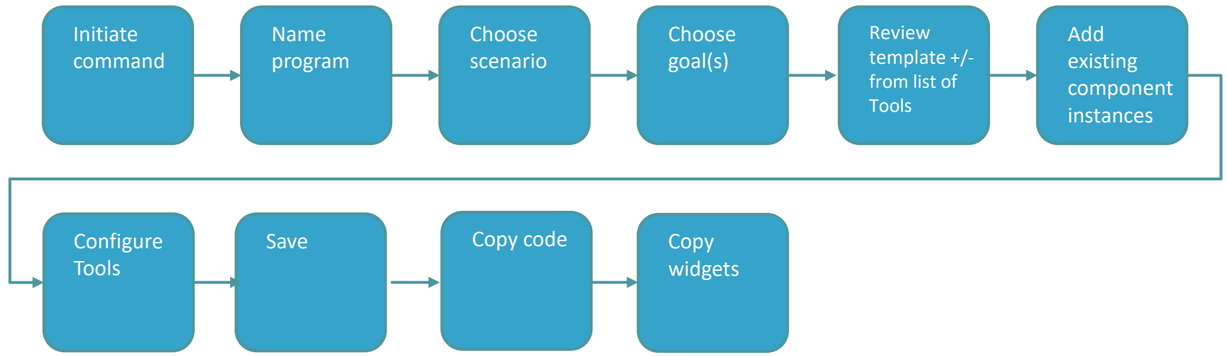
# Core User Task Flows



## 1. Design Game

A. Create Program

1. **Initiate command**
2. **Name program**
3. **Choose scenario**
4. **Choose goal(s)**
5. **Review template and add/subtract from list of Tools**
6. **Add existing component instances**
7. **Configure Tools**
8. **Save**
9. **Copy code**
10. **Copy widgets**




## 2. Monitor

1. **Go to program or to dashboard**
2. **View the desired set of reports**
3. **Click on a report to go to full page**

# Badgeville Usage Stats



| Site | Simple Be | Advanced | Levels | Rewards | Advanced | Missions | T |
|---|---|---|---|---|---|---|---|
| Baseline Configuration (baseline.hsn.net) | 0 | 1 | 1 | 1 | 1 | 1 | |
| Basic (basic.api.com) | 0 | 1 | 1 | 0 | 1 | 1 | |
| Beat The GMAT (www.beatthegmat.com) | 0 | 1 | 0 | 0 | 1 | 0 | |
| Bedsider Production (bedsider.org) | 0 | 1 | 1 | 0 | 1 | 1 | |
| Best Life (www.thebestlife.com) | 0 | 1 | 1 | 1 | 1 | 1 | |
| Beyond the Guide (beyondtheguide.com) | 1 | 0 | 1 | 1 | 0 | 1 | |
| Bill Hennessy Test (hennessygrp.com) | 0 | 1 | 1 | 1 | 1 | 1 | |
| Binekarac SilentMode (binekarac.vw.com.tr) | 1 | 1 | 1 | 0 | 1 | 1 | |
| BlackBook Production Site (blackbookmag.com) | 1 | 0 | 1 | 1 | 0 | 0 | |
| BradsDeals (bradsdeals.com) | 1 | 1 | 1 | 1 | 1 | 1 | |
| Buddy Press KidzVuz (kidzvuz.com) | 1 | 1 | 1 | 1 | 1 | 1 | |
| Buzznet (buzznet.com) | 1 | 1 | 0 | 1 | 1 | 1 | |
| C3 (c3.com) | 0 | 1 | 1 | 1 | 1 | 1 | |

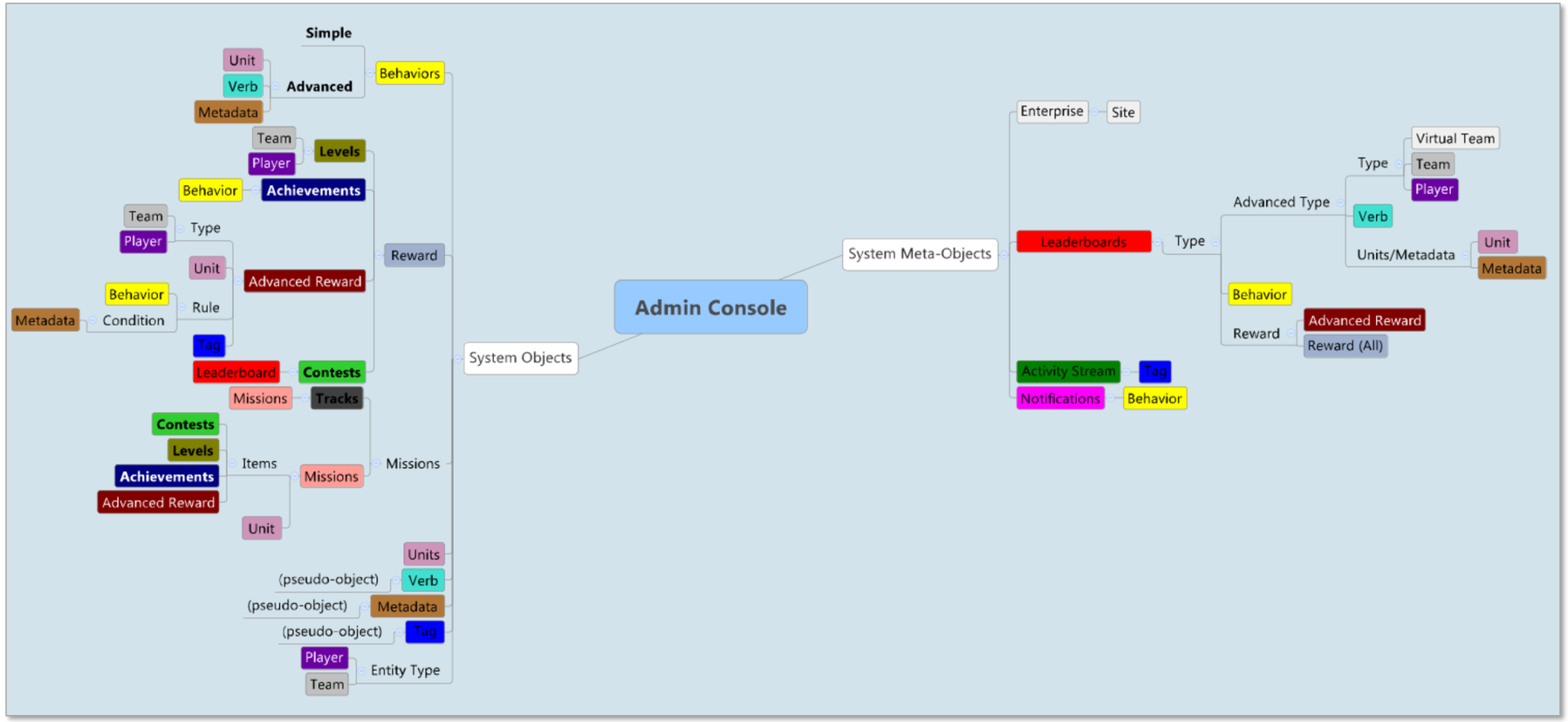
**Pulling data from the system tells us what's being used (or not)**

**I also worked with internal teams to get insight into customer support issues**

Badgeville
The Behavior Platform™

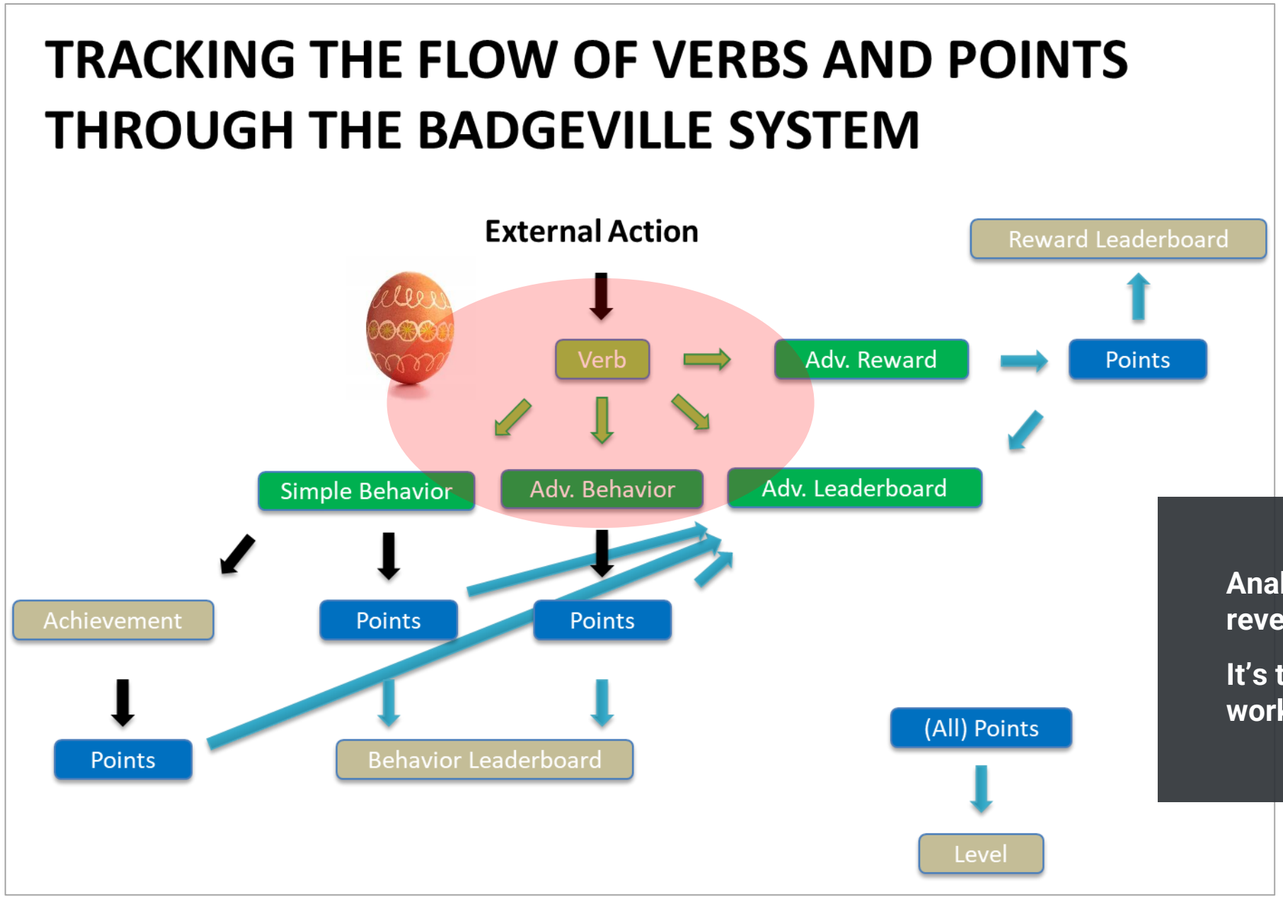
# Conceptual-mapping of Object Relationships

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact



# Mapping out Objects and Data Flows

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact




**Analyzing how the system works reveals a hidden truth:**

**It's the verbs that underlie the working of major system objects**

# Simplify and Consolidate



Treat all 3 types of behaviors consistently as a singular object (just **Behaviors**)

Verb + Simple Behavior + Advanced Behavior → Behaviors

Treat **Achievements** and **Advanced Rewards** consistently (just Rewards)

Achievement + Advanced Reward → Rewards

Deprecate **Levels** in lieu of a Mission type that accomplishes the same goal.

Level → Type of progressive mission (Level-type)

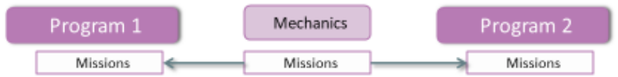
Deprecate Tags in favor of a single Metadata library (use Keys only in place of Tags)

Tags → Replaced by Metadata key

Create new object: **Programs**

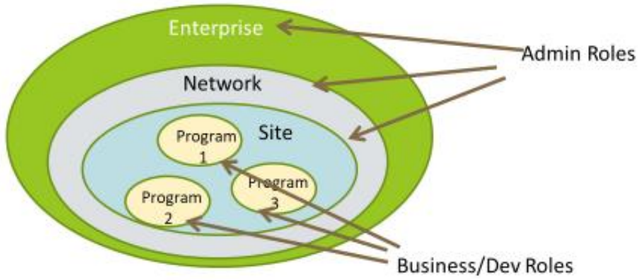
Program* (new)

Architect the product so that programs can share and reuse common system objects



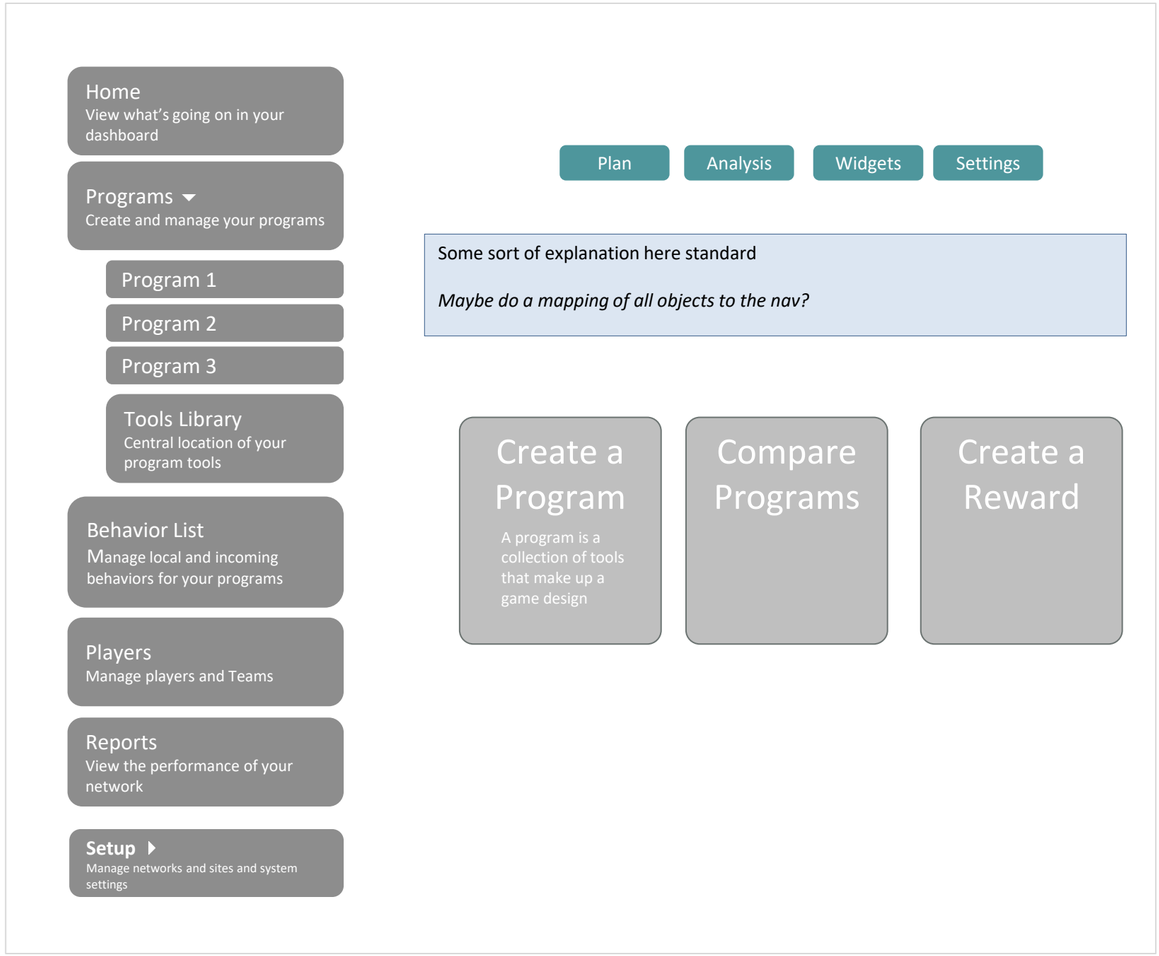

**There were a lot of opportunities to simplify the representation of the system for users.**

**Further, I proposed new UI concepts such as 'Programs' that would make it easier to structure campaigns as well as share and reuse game objects.**



# Improved Navigational Model

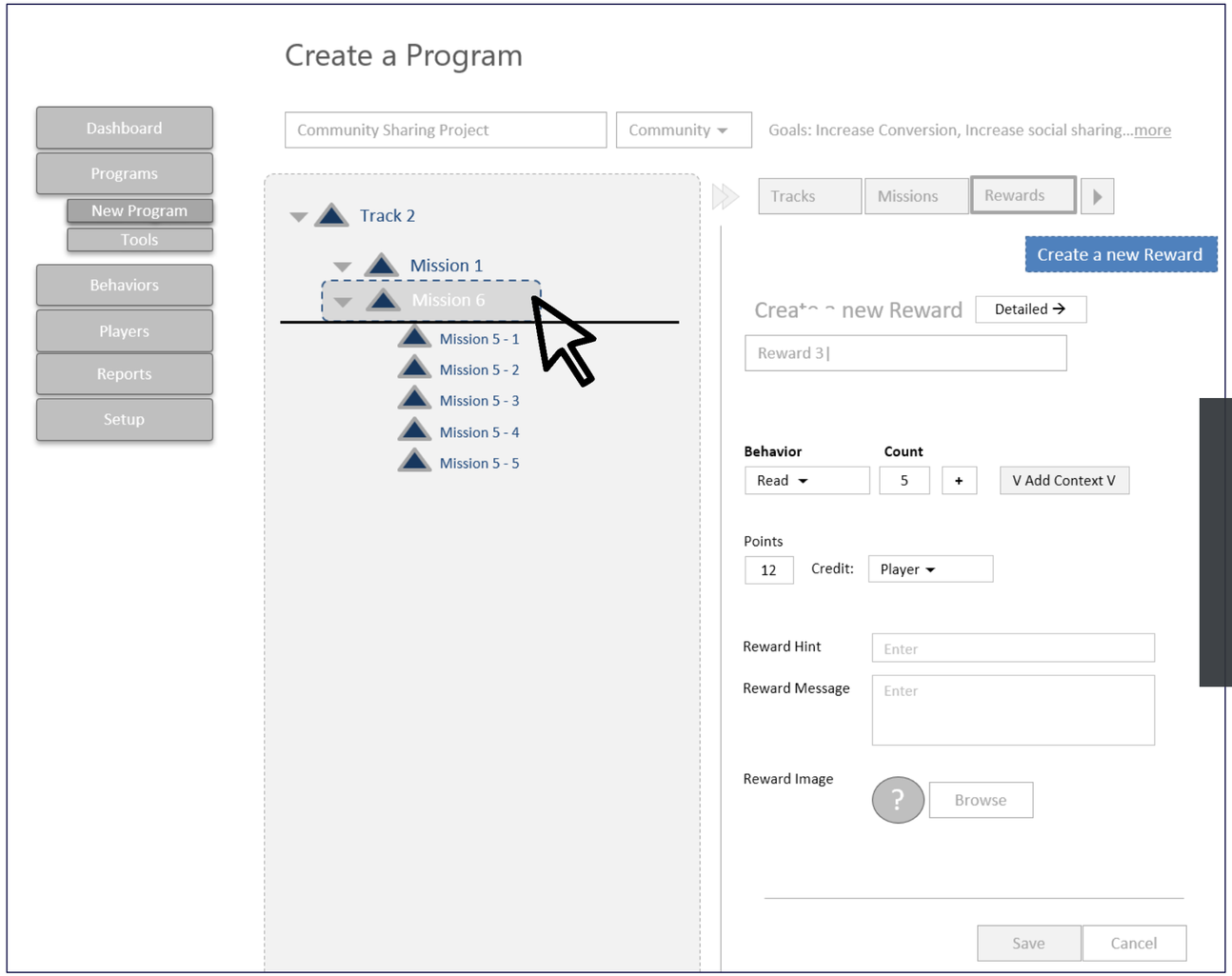





**A new information architecture supported the core user needs more simply and directly than before.**

# Create a Simple Object



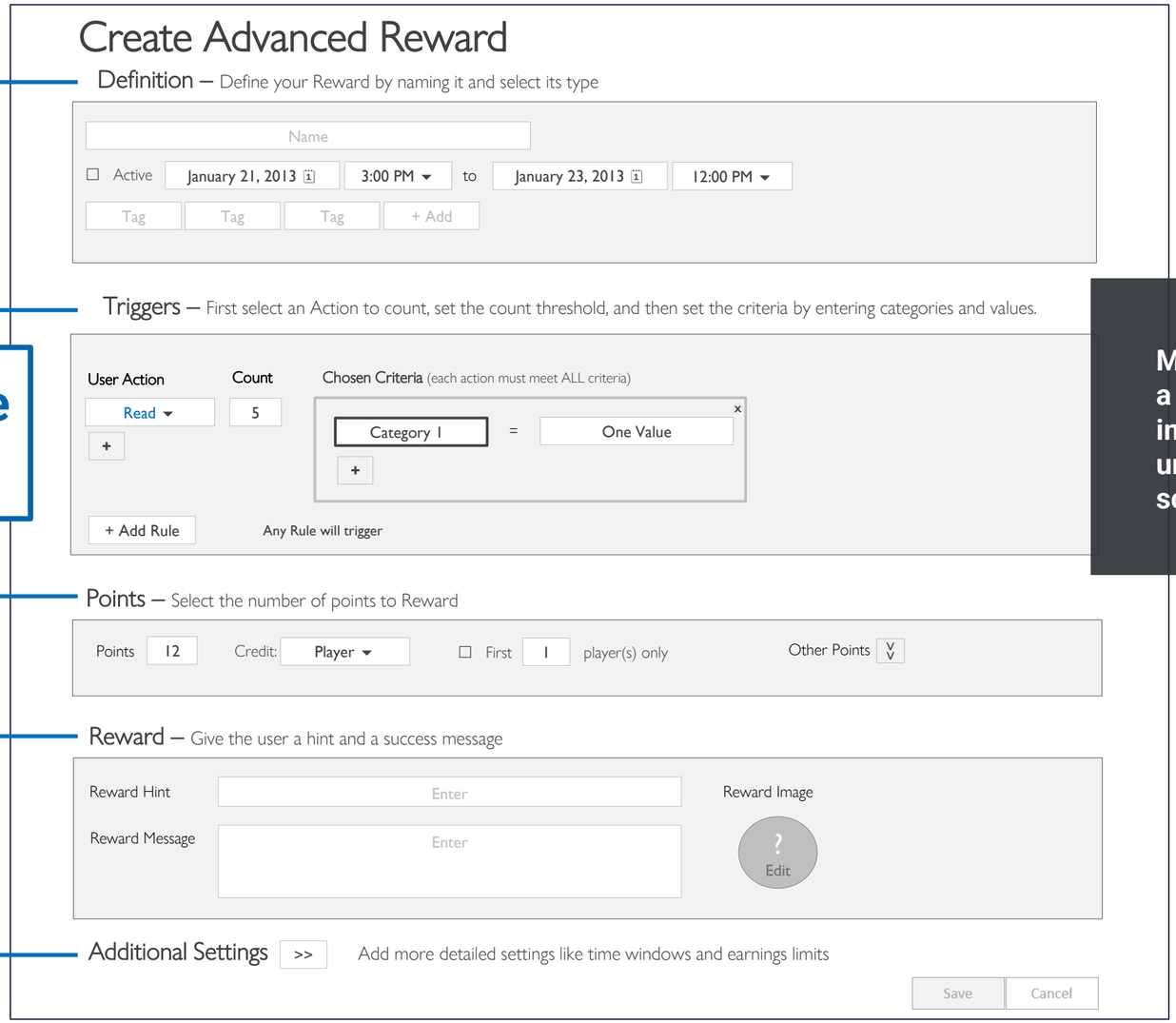



**I proposed using advanced interactions like direct manipulation to make the configuration experience easier and more natural.**

# Simplifying Complex Objects

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

**Scannable structure**

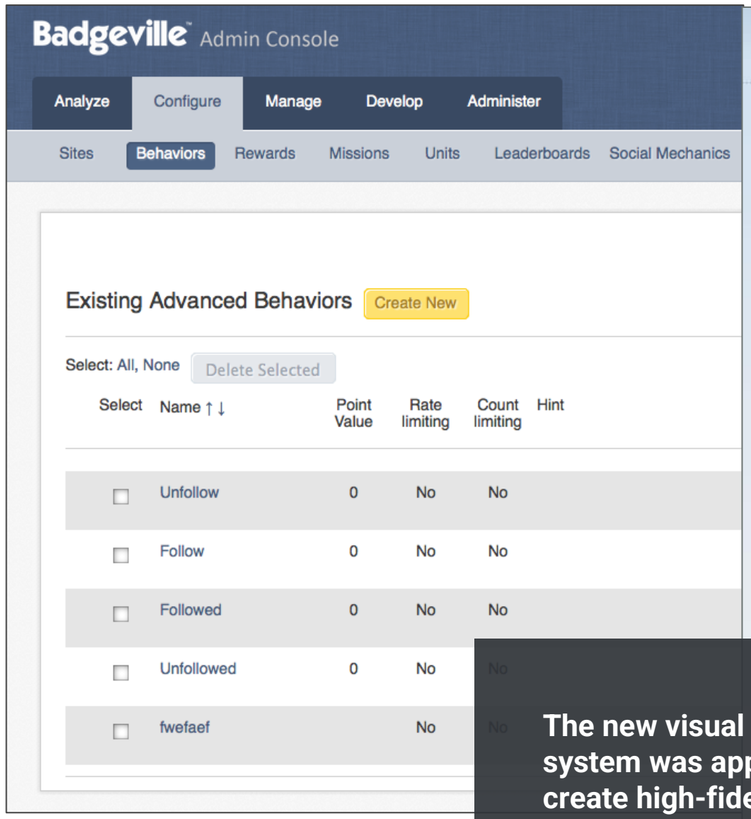



**More complex objects with a deeper data model were improved to be easier to understand by enhancing scannability.**

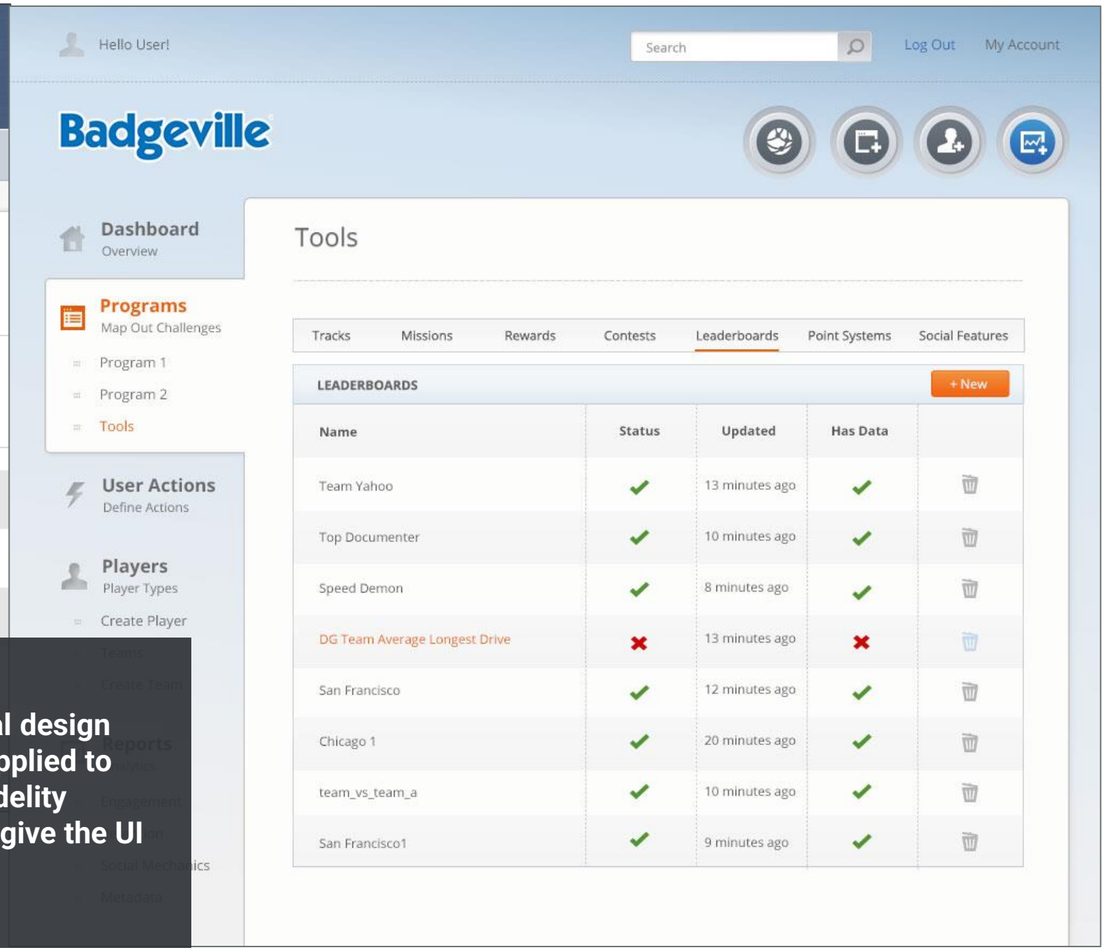
# New Visual System

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

From this:



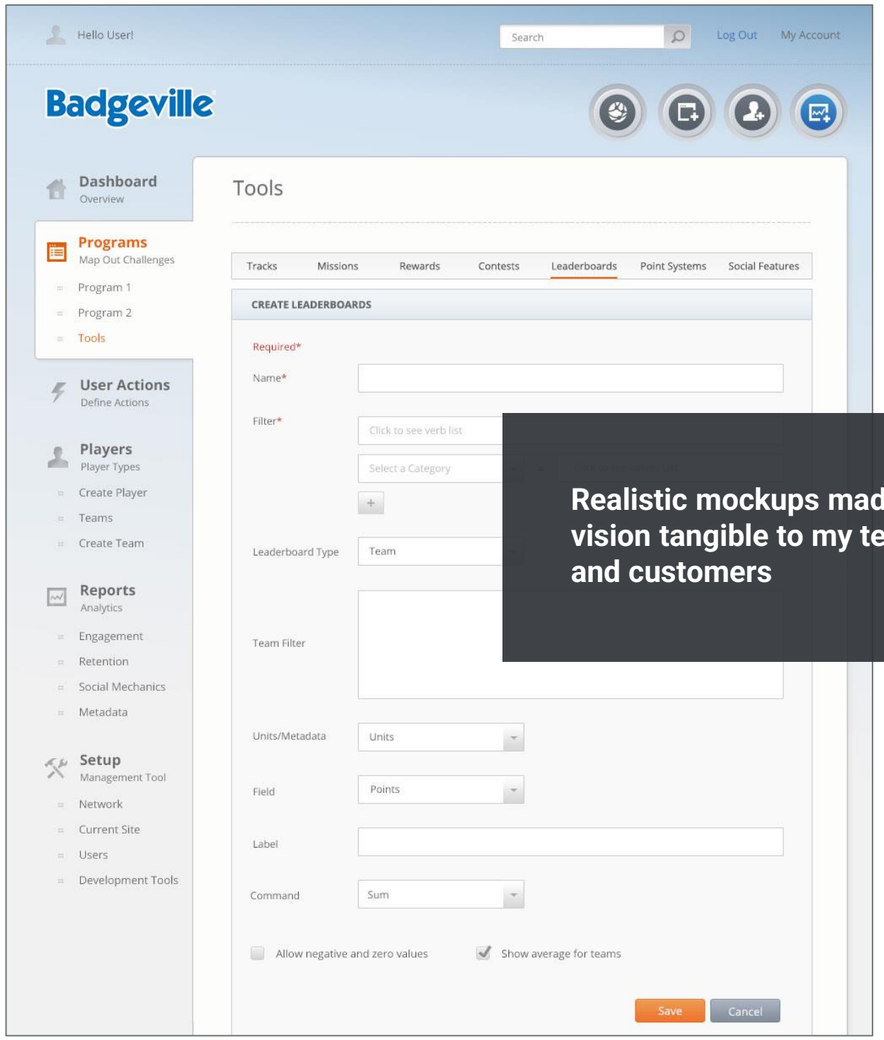

**To this:**



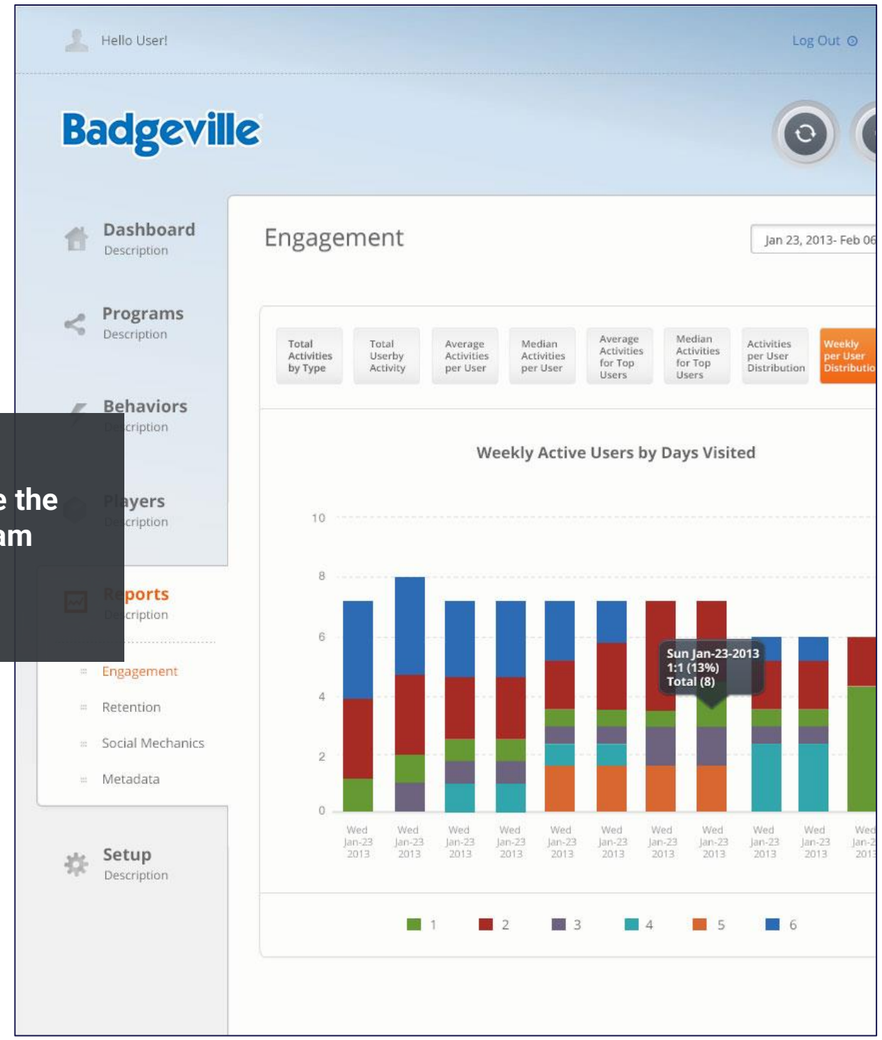

**The new visual design system was applied to create high-fidelity mockups and give the UI a fresher look**

# Benefits of High-Fidelity Mockups

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact

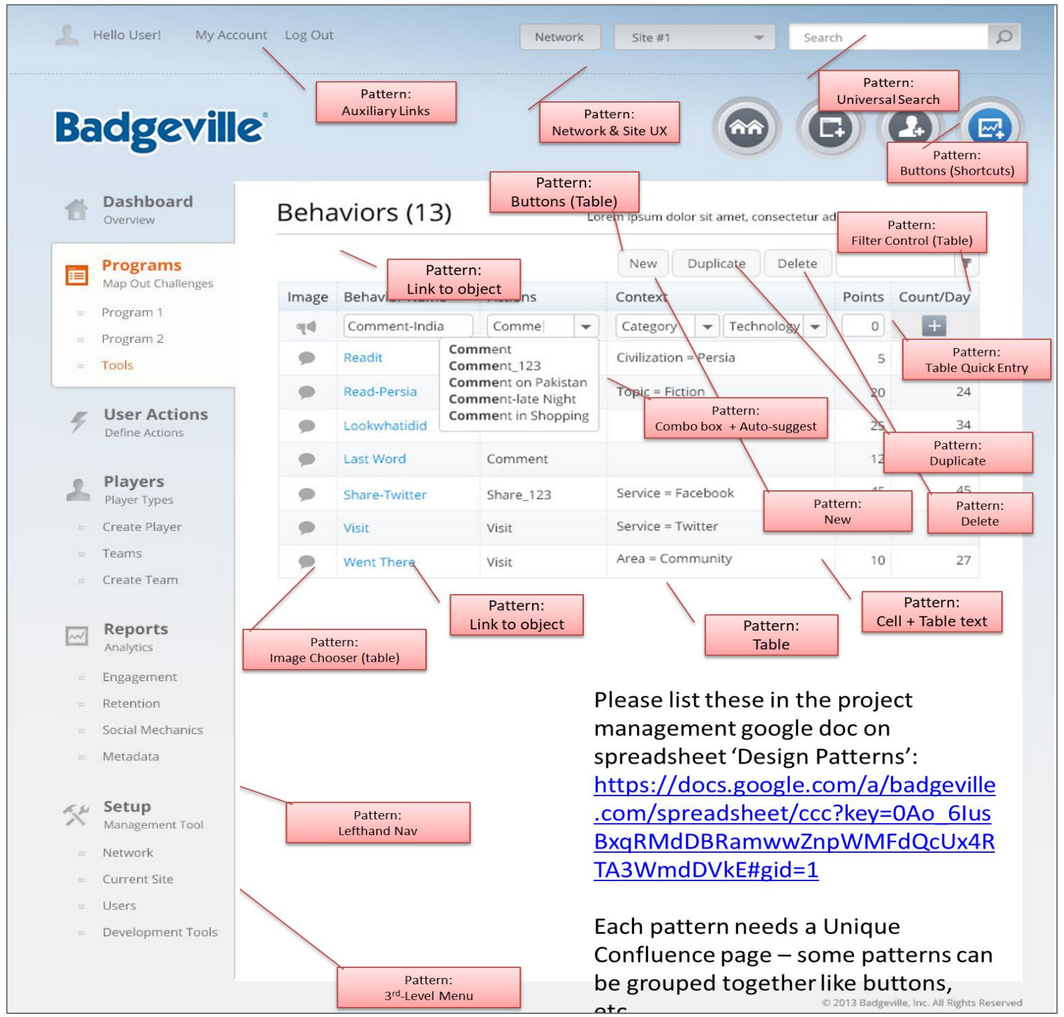



**Realistic mockups made the vision tangible to my team and customers**



# Patterns Library

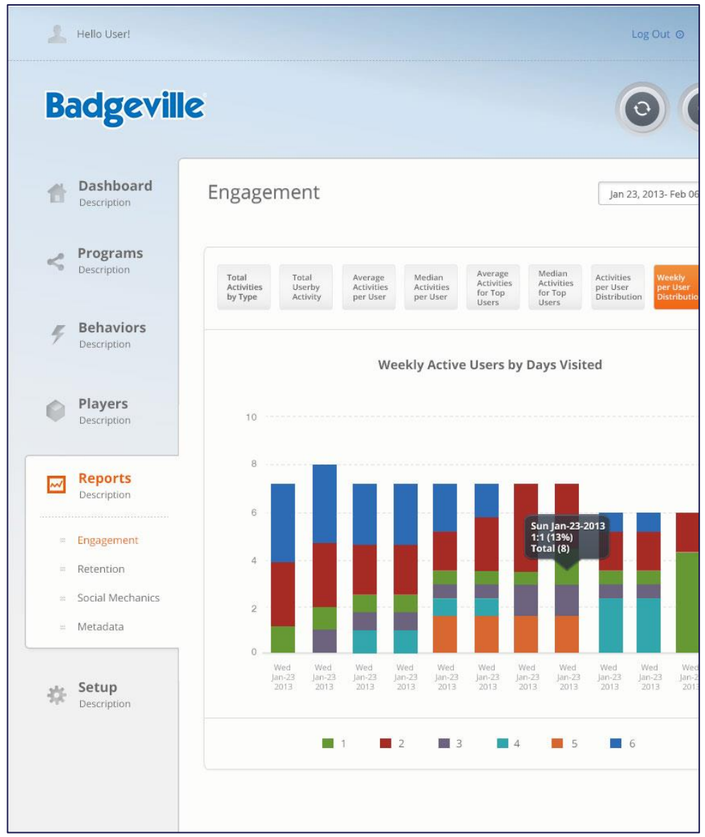
1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact




**After specifications, I created a pattern library to enhance UI component reusability.**

# Outcomes and Impact

1: Problem & Process
2: Domain
3: Personas
4: Task Flows
5: Objects & System
6: Proposal
7: Navigation and IA
8: Wireframes
9: High Fidelity Mocks
10: Patterns
11: Outcomes and Impact




**Customers were excited to see the prototypes which allowed me to collect detailed feedback and drive design iterations. The new UI went into the next major platform release.**

# END OF CASE STUDIES

# Designs

Intuit + Yahoo!

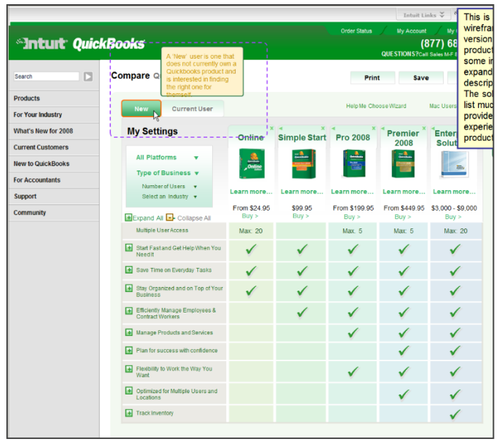
Apple

Dreamworks

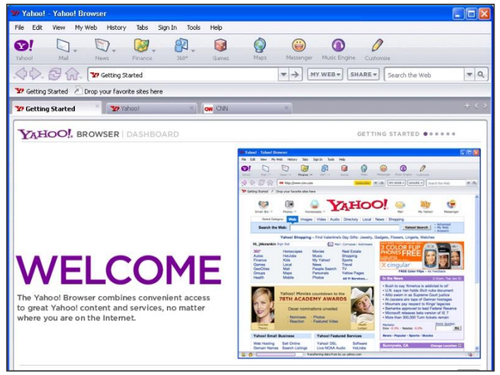
Oracle

# Consumer Design (10 million+ users)

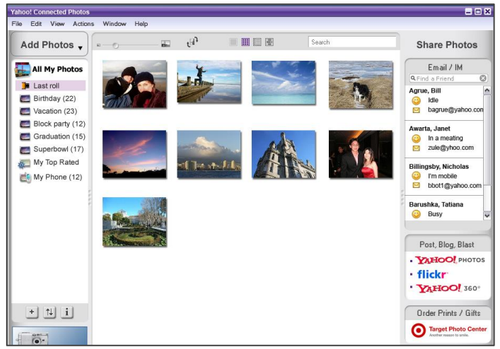

- Intuit.com redesign



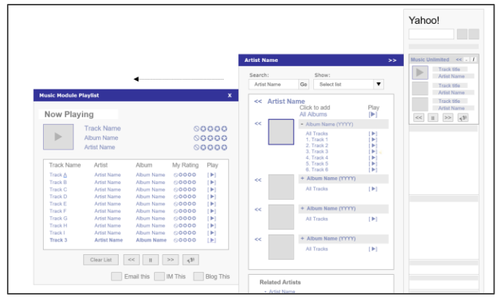

- Quickbooks Product Chooser
- Quickbooks Online
- Intuit Mortgages




- Yahoo! Social Browser



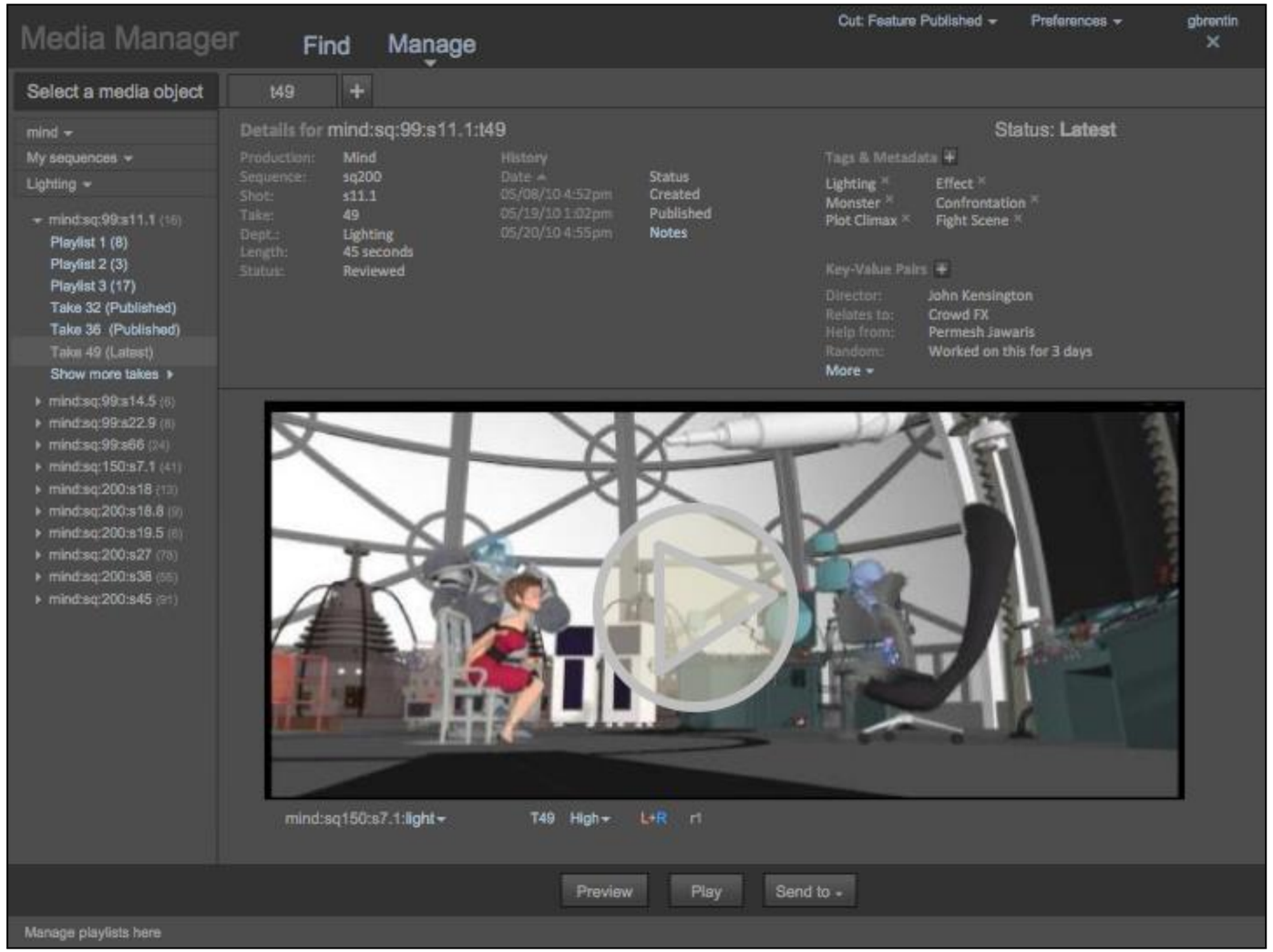

- Yahoo! Photos



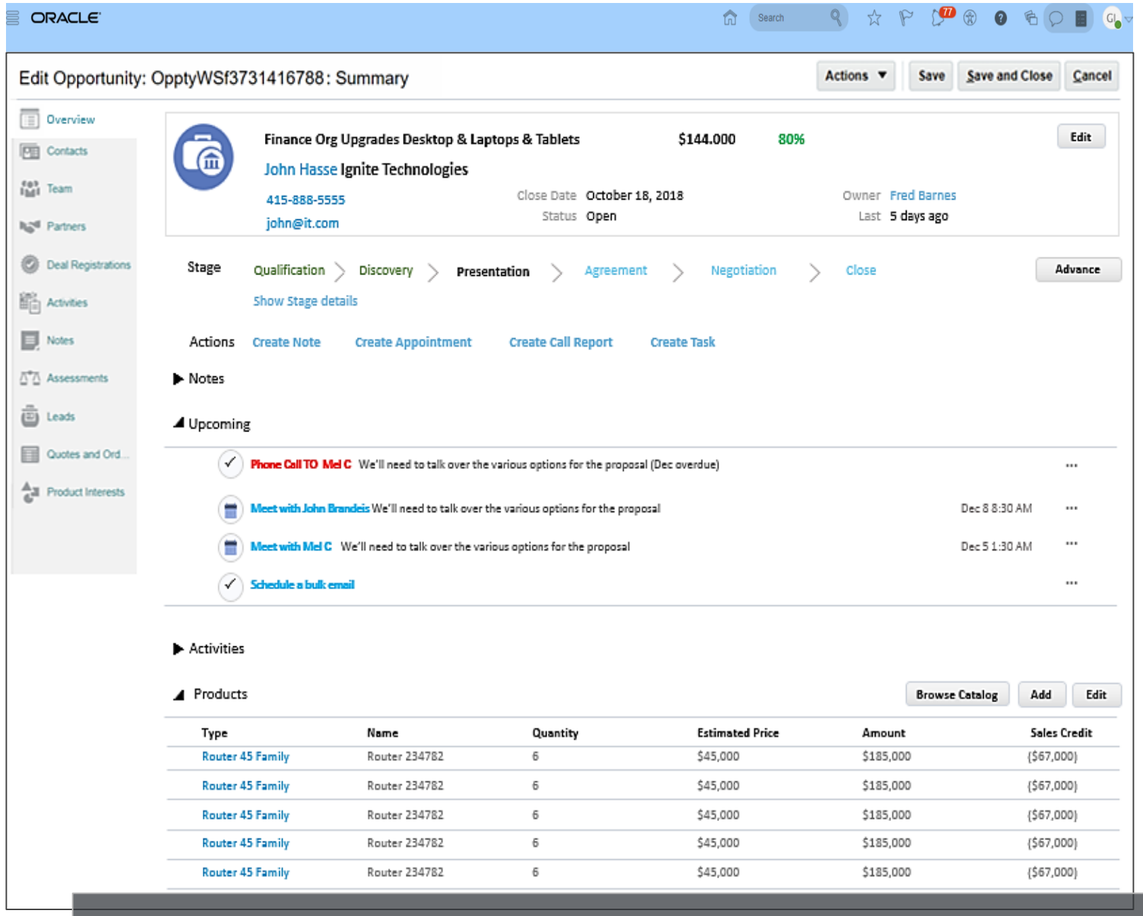

- Yahoo! Sidebar

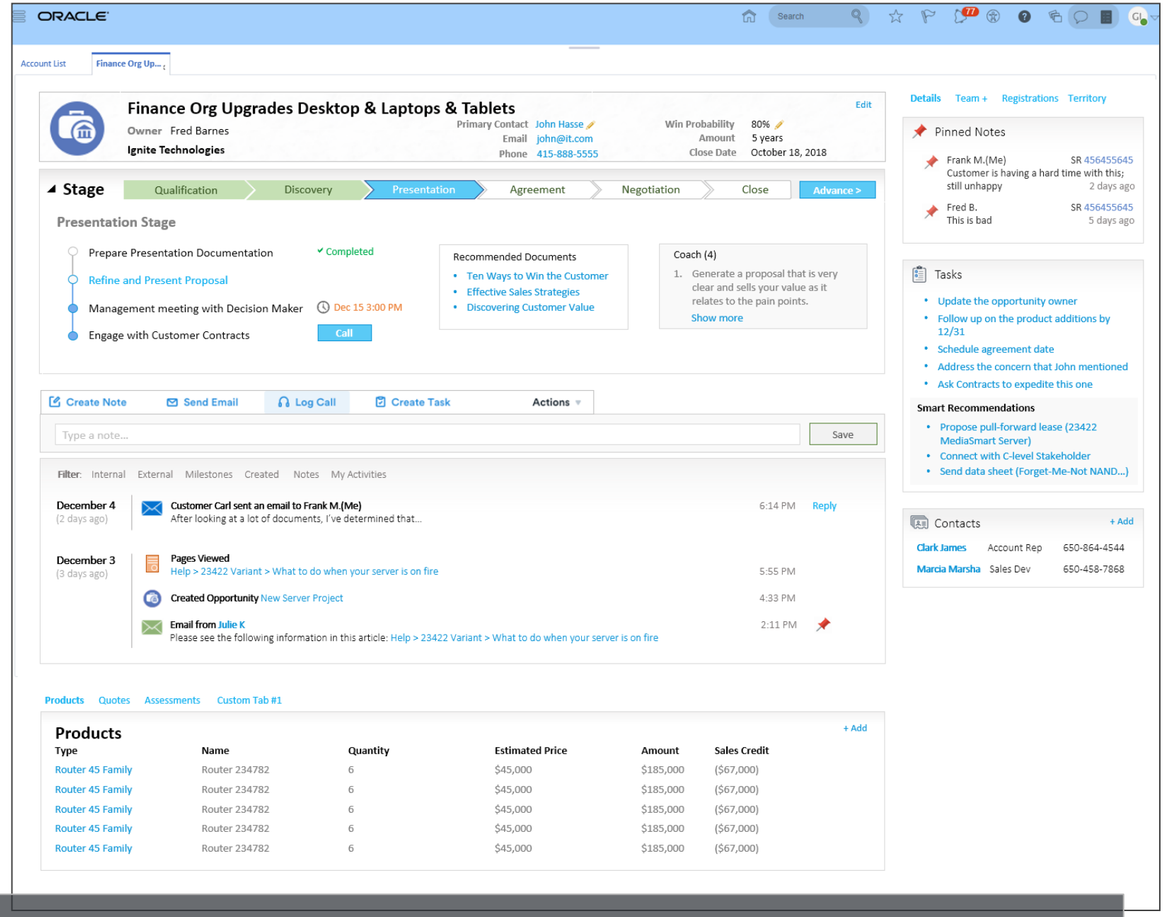
# Media Manager

I designed an internal tool for DreamWorks called Media Manager to allow animators and directors to manage the flow of daily animation sequences as movies are being crafted.
Media Manager was a content management platform designed to help animators manage large animation renderings and to move them through an approvals workflow.



# Sales Cloud Opportunity Modernization

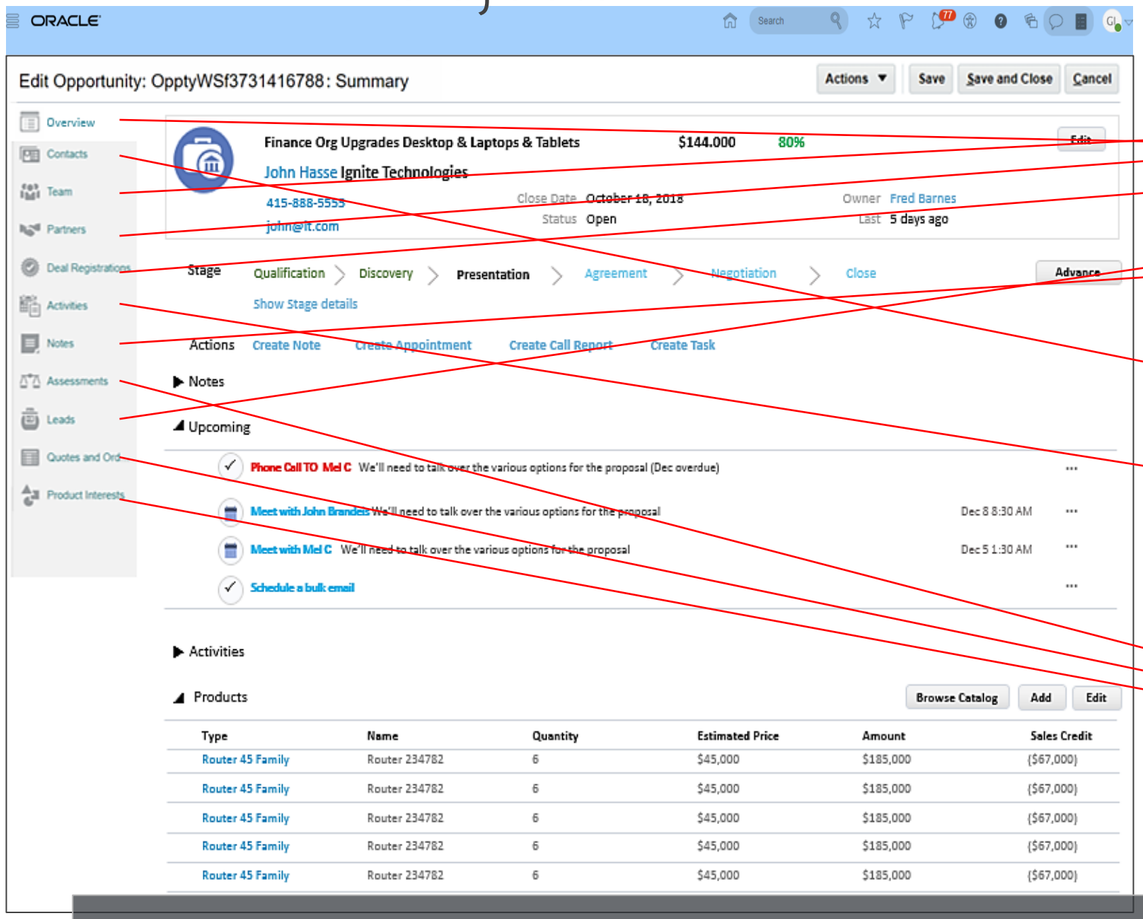


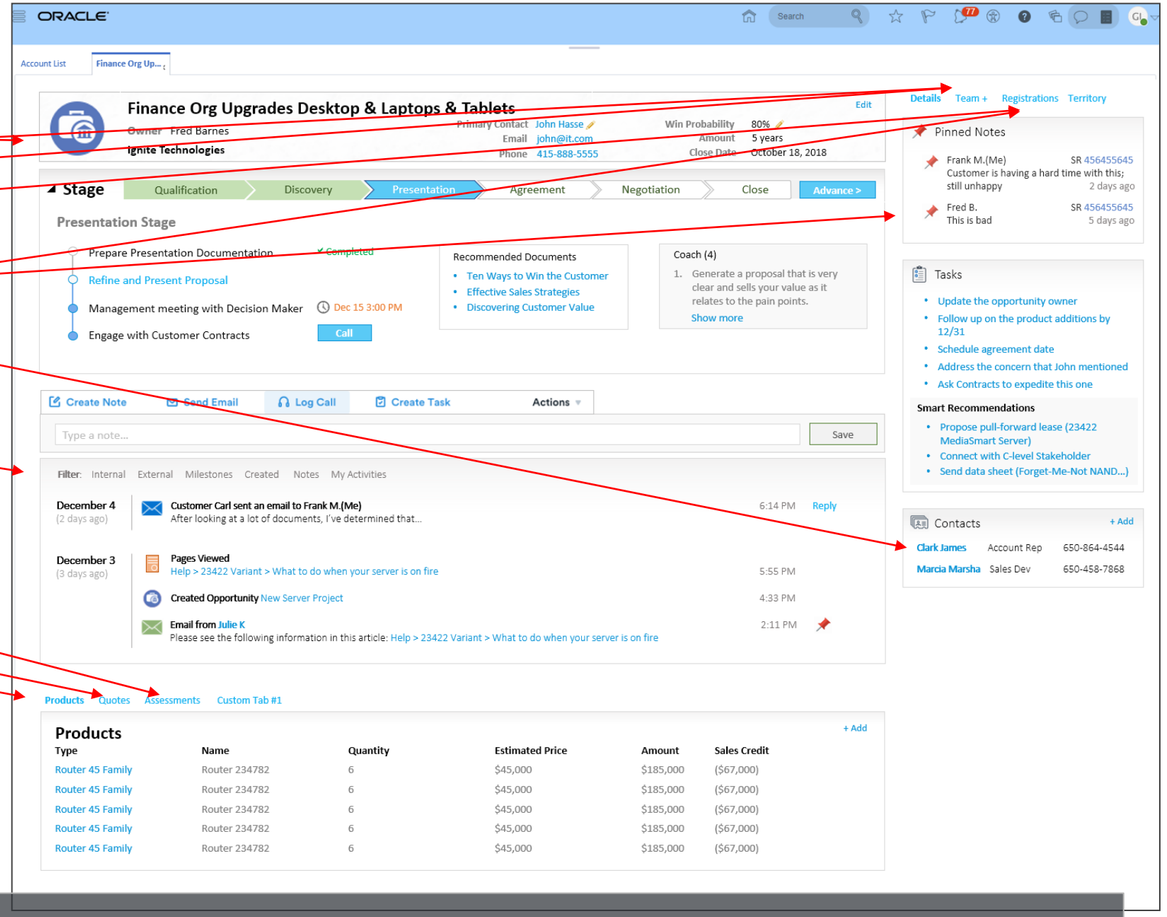

I redesigned the Overview page for Sales Opportunities to be richer, more actionable (you can now create notes, send emails etc.) and more structured visually with more curb appeal.

# Sales Cloud Opportunity Modernization

## Objects




## Tasks




Notice the mapping of the objects to an information architecture hierarchy that is determined by information priority and frequency of use.

# END OF PORTFOLIO