



# CalProc from Start to Finish

<https://www.calproc.website/>

March 3, 2017



# Table of contents



1. From Start to Finish: Our Methodology and Approach
2. The Team
3. User-centric design and usability
4. Technical Architecture
5. DevOps
6. Our Commitment to Agile
7. Conclusion and Next Steps
8. See Also



# From Start to Finish: Our methodology and approach

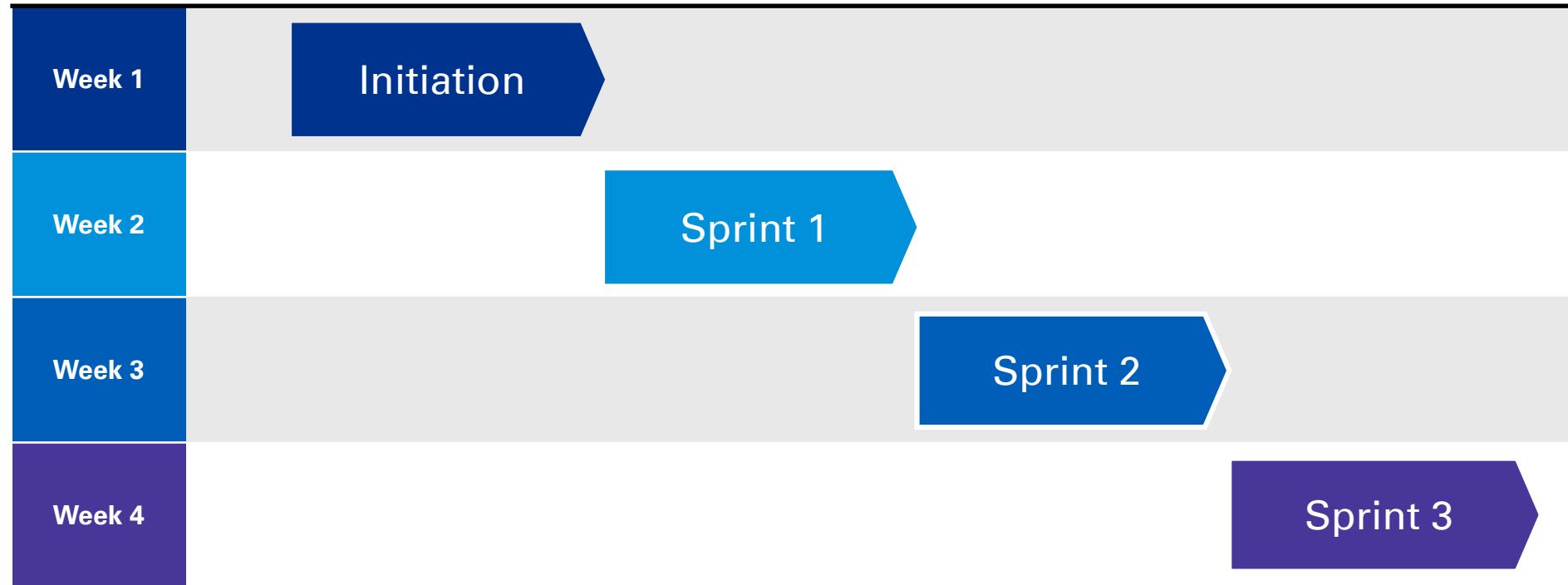
# We selected Prototype A

## **Prototype A Requirements (from the RFI):**

The working prototype will be an application that will allow authorized users to compare and order end-user computing hardware (e.g., desktops, laptops, monitors), software (e.g., office productivity tools), and related services from pre-established state contracts, and allow authorized users to cancel, track and analyze their orders.

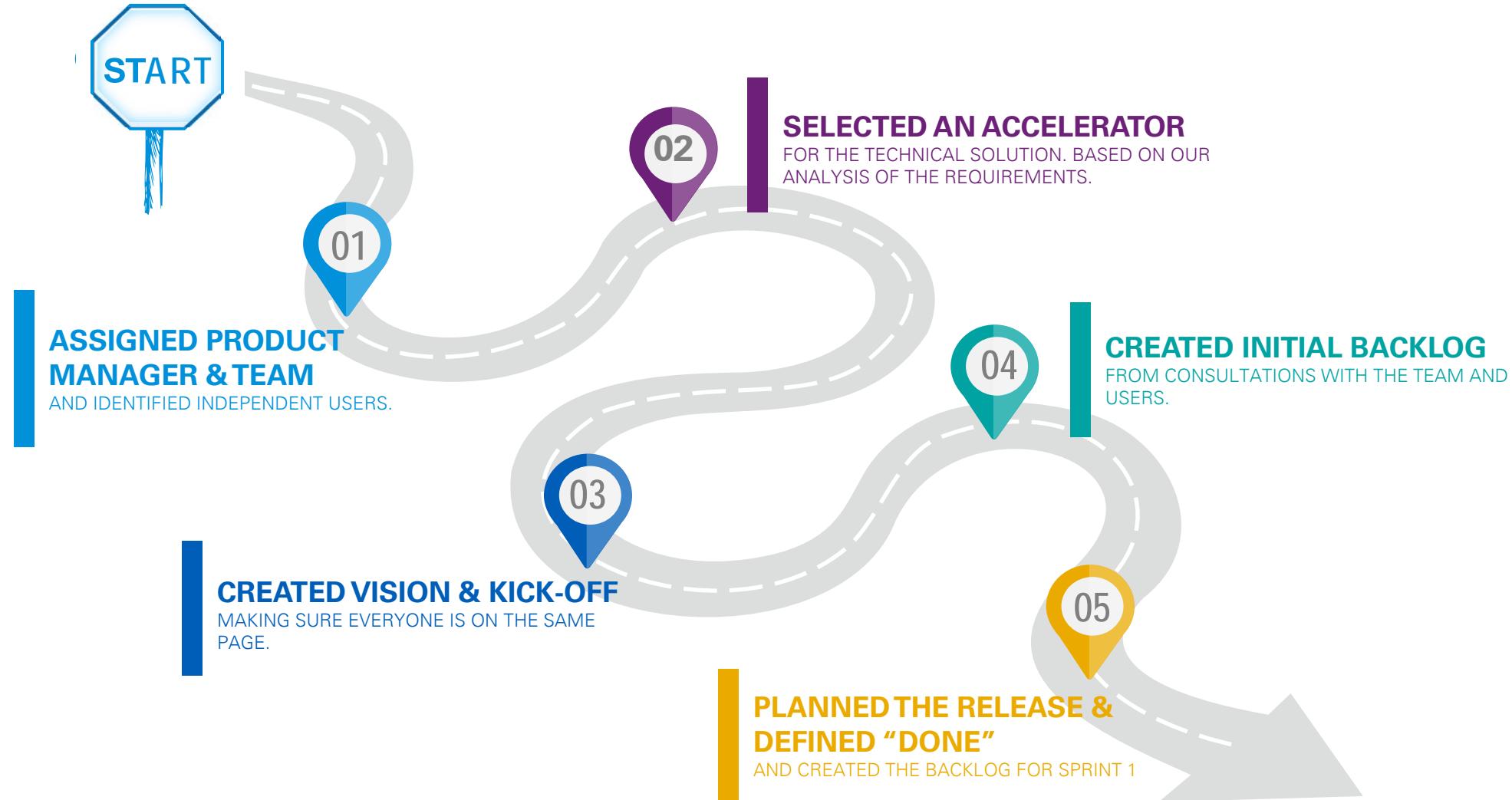
In addition, the working prototype will provide the authorized administrative users who are employees of state's lead purchasing organizations – the Department of General Services and the Department of Technology – with the ability to publish product and service information and track, analyze and visualize order data. The working prototype does not need to implement any authentication or authorization against an external directory or authentication mechanism.

# What we did: At a glance



After an initiation week, the team iterated through 3 weekly sprints.

# Initiation week: At a glance



# Initiation week



## ASSIGNED PRODUCT MANAGER & TEAM

AND IDENTIFIED INDEPENDENT USERS.

Ben Rogers (CSM) was assigned as the Product Manager. He was given authority and responsibility and was held accountable for the quality of our prototype.

Based on the prototype requirements and individual skill sets, we commissioned the following team:

**1. Product Manager**  
Ben Rogers, CSM

**2. Technical Architect**  
Robert Levy

**3. Interaction Designer/User Researcher/Tester**  
Ryan Lee

**4. Writer/Content Designer/Strategist**  
Cory Fritzsching

**5. Front End Developers**  
Nick Pearce (1)  
Casey Rayl (2)

**6. Backend Developer**  
Sandeep Pedditi

**7. DevOps Engineers**  
Chris Robinson (1)

**8. Agile Coach**  
Matt Kwong, PSM

**9. Business Analyst**  
Stacy Lee, CSM

**10. Business Analyst**  
Simon Chen, CSM

# Initiation week

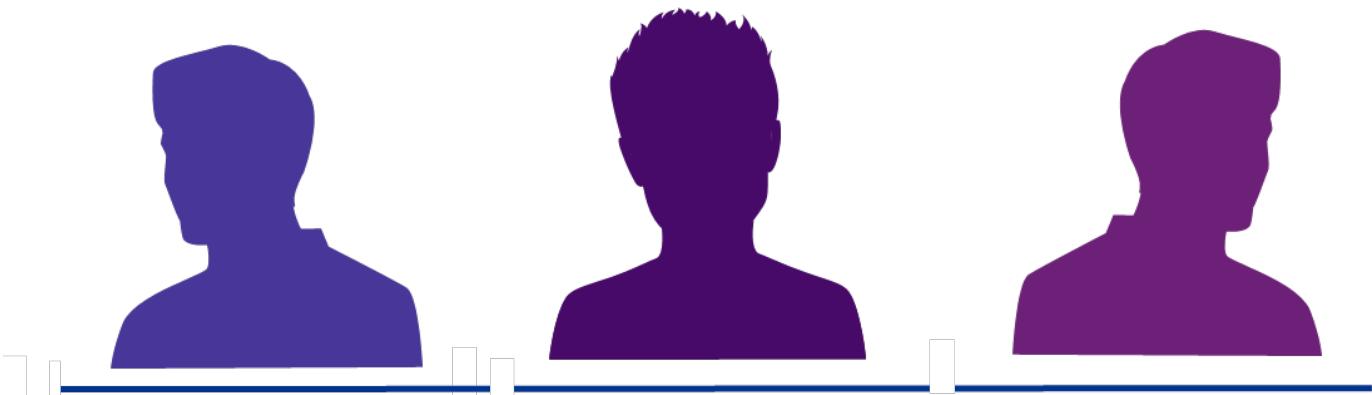


## ASSIGNED PRODUCT MANAGER & TEAM

AND IDENTIFIED INDEPENDENT USERS.



We identified independent users of the prototype. We recruited users from a wide range of backgrounds, experiences with the State of California, and with procurement systems. These users would provide feedback throughout.



# Initiation week

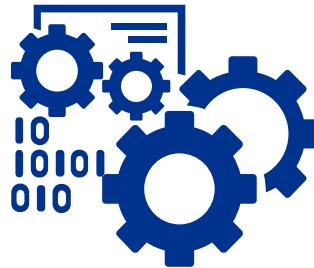


**01**  
**ASSIGNED PRODUCT  
MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.

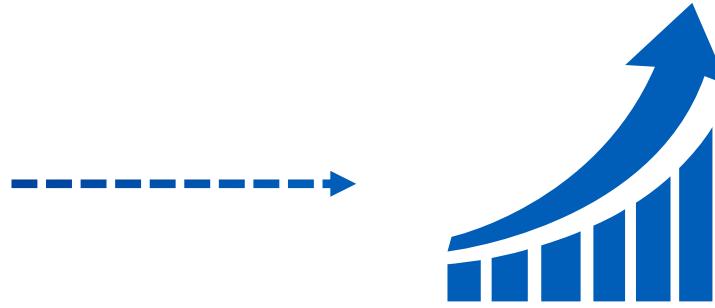


**02**  
**SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR  
ANALYSIS OF THE REQUIREMENTS.

After analyzing the prototype requirements and timeframe, the development team decided to use an accelerator from KPMG's asset library. The aim was to use existing software to reduce risk and maximize the functionality we could deliver in the timeframe.



**webstart v3**  
ACCELERATOR



**BENEFITS**  
▪ HIGH QUALITY  
▪ IMPROVED VELOCITY

The accelerator selected was the 3<sup>rd</sup> generation of webstart. This is a project starter for web development that orchestrates several open-source tools together into a cohesive workflow.

# Initiation week

01

ASSIGNED PRODUCT  
MANAGER & TEAM

AND IDENTIFIED INDEPENDENT USERS.

02

SELECTED AN ACCELERATOR  
FOR THE TECHNICAL SOLUTION. BASED ON OUR  
ANALYSIS OF THE REQUIREMENTS.

We created a Development Guide based on the initial accelerator, meaning the team could get setup quickly, and all follow the same standards.

Branch: master ca-pqvp / DEVGUIDE.md Find file Copy path

robertlevy Update unittest scaffolds (#120) f144902 5 days ago

2 contributors

176 lines (130 sloc) | 7.4 KB Raw Blame History

## Overview

This is the 3rd generation of webstart, a project starter for web development that orchestrates several open-source tools together into a cohesive workflow. This iteration concentrates on reducing complexity, abstractions and dependencies of the previous versions making this version easier to maintain.

### Whats under the hood?

- Webpack - The module bundler for client-side code, configuration files can be found in `config/webpack.config*.js`. Initial v1 was released February 2014
- Babel - Transpiles ES6+ standards back to ES5 compliant code to run in all browsers, configuration files can be found in `.babelrc`. Based on 6to5 released March 2014
- Express - The framework for the server-side code used for hosting the client-side code and also optionally for server-side API and/or rendering. Initial v1 was released November 2010
- RAML - API design and modeling with code and documentation generation. Hooks are already provided in both the client and server code of this project to handle this. v1 spec was finalized May 2016
- Karma - The runner for unit testing the application in a browser, configuration files can be found in `config/karma*.config.js`. v1 was released June 2016
- Protractor - The runner for end-to-end testing the application in a browser, configuration files can be found in `config/protractor.config.js`. v1 was released July 2014
- Istanbul - The code coverage tool used to test coverage of both Karma and Protractor. v0.1.26 was the first release on GitHub in January 2013
- Jasmine - The default expectation framework used for both Karma and Protractor. v1 was released September 2010
- ESLint - The linter for ES code, configuration files can be found in `.eslintrc`. v1 was released July 2015
- ESDoc - The documenter for ES code, configuration files can be found in `config/esdoc.config.js`. v0.0.1 was released April 2015
- i18next - The localization framework, configuration files can be found in `config/i18next.config.js`.



# Initiation week

01

**ASSIGNED PRODUCT  
MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.

02

**SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR  
ANALYSIS OF THE REQUIREMENTS.

03

**CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME  
PAGE.

The product manager worked with the team and users to create a product vision. We chose a brand name and created a logo as soon as possible. We found this kept the team closely aligned and motivated throughout the project.

## Product Vision

Ben Rogers edited this page 3 minutes ago · 3 revisions



For government employees within California who need to procure computing hardware, software and related services from pre-established state contracts, the "CalProc" procurement system provides ordering, tracking and analysis features. Also, it allows employees of the state's leading purchasing organization the ability to publish product and service information and then track and analyze order data.

Unlike other services, our product provides a very simple and intuitive platform for users.

Edit New Page

▼ Pages (32)

Find a Page...

Home

Adherence to the US Digital Services Playbook

Competitive Analysis

Contract Management  
Information Architecture

Database Credentials  
Configuration in .dbconfig file

Definition Of Done (DOD)

Design Links

Design Plan

Logo Concepts

Outline of Application

Persona Structure (Draft In  
Process)

Product Vision

# Initiation week

01

**ASSIGNED PRODUCT  
MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.

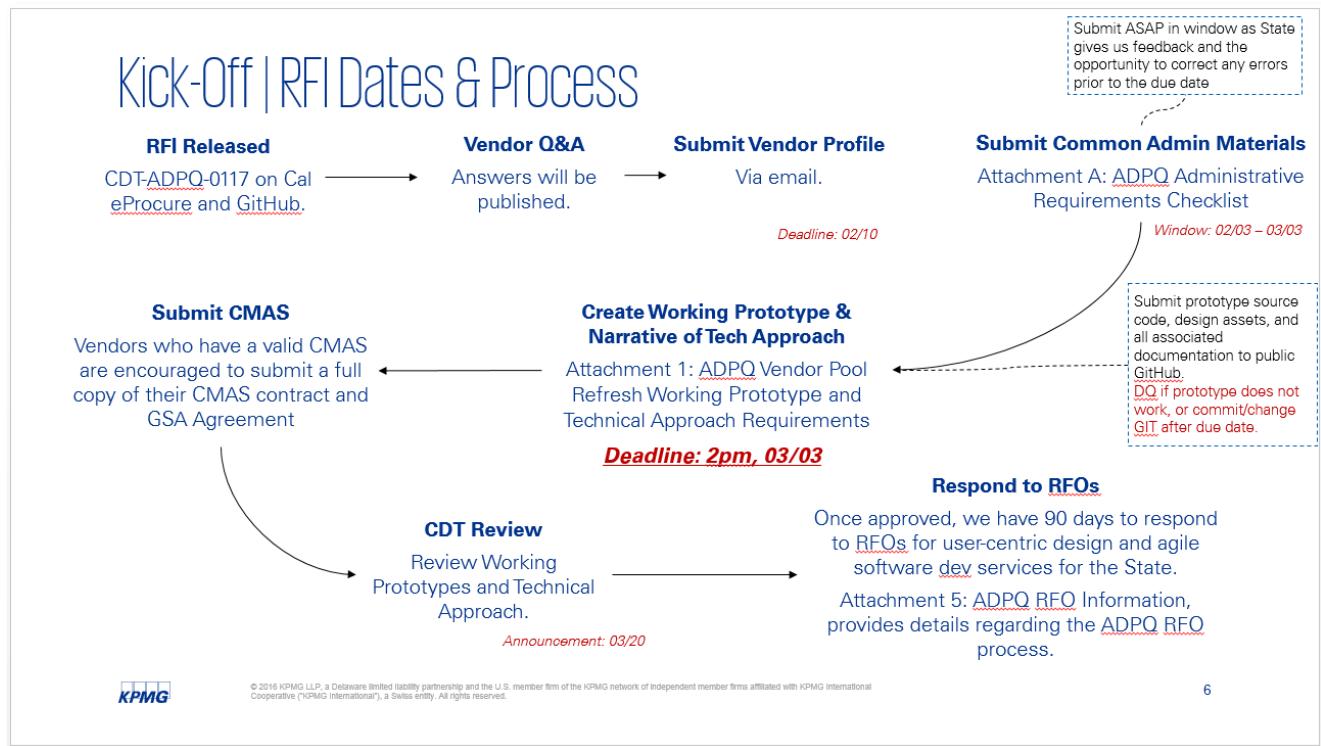
02

**SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR  
ANALYSIS OF THE REQUIREMENTS.

03

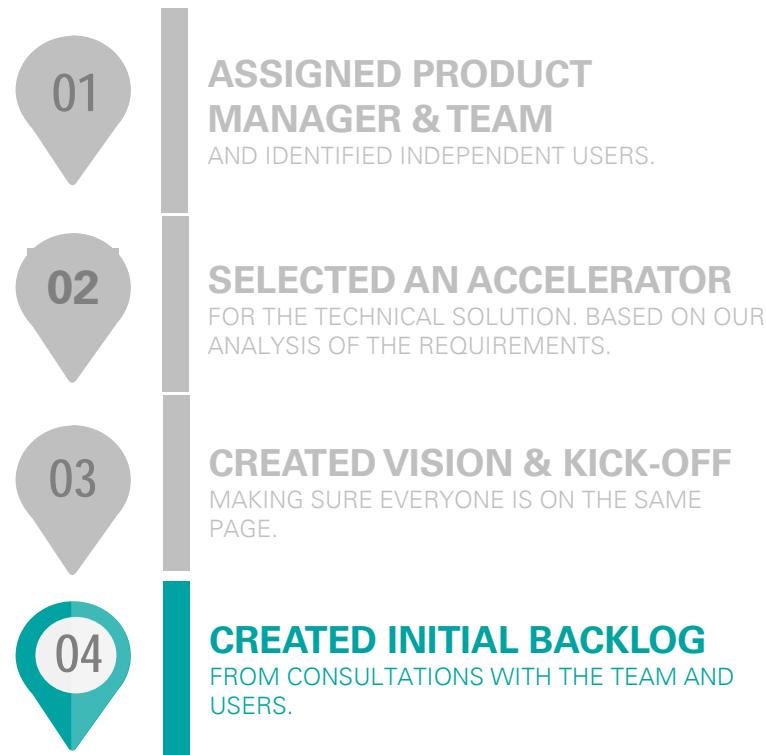
**CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME  
PAGE.

The product manager held an official kick-off meeting that communicated the vision and explained the RFI process and requirements. It was important that the entire team was aware of the big picture and what we were ultimately working towards.



A slide from the kick-off presentation

# Initiation week



The product manager created an initial product backlog. This was a collaborative activity between the product manager, the development team and the independent users.

This backlog represented all of the tasks required to complete the prototype.

This backlog included:

- USER STORIES, CREATED FROM THE PROTOTYPE DESCRIPTION AND INITIAL USER CONSULTATIONS
- DESIGN TASKS (E.G. SCHEDULING USER INTERVIEWS)
- TECHNICAL TASKS (E.G. SETTING UP THE REPOSITORY)
- DEV OPS TASKS (E.G. PROVISIONING MACHINES)

Everyone's ideas and work tasks were considered when creating the initial backlog.

# Initiation week



## 01 ASSIGNED PRODUCT MANAGER & TEAM

AND IDENTIFIED INDEPENDENT USERS.



## 02 SELECTED AN ACCELERATOR

FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.



## 03 CREATED VISION & KICK-OFF

MAKING SURE EVERYONE IS ON THE SAME PAGE.



## 04 CREATED INITIAL BACKLOG

FROM CONSULTATIONS WITH THE TEAM AND USERS.

We used GitHub Issues to represent items in the backlog, and labels to differentiate between the types of item or work:

The screenshot shows a GitHub Issues page for the repository 'kpmg-agile / ca-pqvp'. The interface includes a header with repository information, a navigation bar with links like 'Code', 'Issues 61', 'Pull requests 4', 'Projects 5', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. Below the header is a search bar with the query 'is:open is:issue label:story'. There are also 'Filters' and 'New issue' buttons. A 'Clear current search query, filters, and sorts' button is present. The main area displays a list of 28 open issues, each representing a user story. The stories are categorized by color-coded labels: yellow for 'As an authorized user', green for 'As an Admin User', and blue for 'As an Admin User' (continued). Each story includes a point value (e.g., 2 points, 3 points), a 'story' link, and details about who opened it and its sprint status.

Story Type	Point Value	Description	Opened By	Sprint Status
As an authorized user	2 points	I want to be able to adjust quantities and remove items from my cart	robertlevy	
As an Admin User	3 points	I want to add labels, meta data and other custom fields to an item, which can appear in search results so that I can help Authorized Users make better decisions. I can extend the solution without developer hours.	mattkwong-kpmg	Sprint 3
As an Admin User	2 points	I want to create a new contract which can be linked to specific items so that I can extend the solution without the need to pay for developer hours.	mattkwong-kpmg	Sprint 3
As an Admin User	2 points	I want to view and configure dashboards and visualizations of order data so that I can quickly get the information I need to plan and react accordingly.	mattkwong-kpmg	Sprint 3
As an Admin User	2 points	I want to view and search through all of the orders across all departments so that I can audit and investigate specific orders.	mattkwong-kpmg	Sprint 3

In our experience, reducing the number of tools used to run an agile project keeps things simple and easier to track. Having everything in GitHub allowed us to stay on top of things, and provided complete transparency to the Product Manager.



# Initiation week

- 01 **ASSIGNED PRODUCT MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.
- 02 **SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.
- 03 **CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME PAGE.
- 04 **CREATED INITIAL BACKLOG**  
FROM CONSULTATIONS WITH THE TEAM AND USERS.

We used GitHub Issue labels to assign “points” to each story, based on their relative complexity. These represented an estimate of the effort required to complete that item.



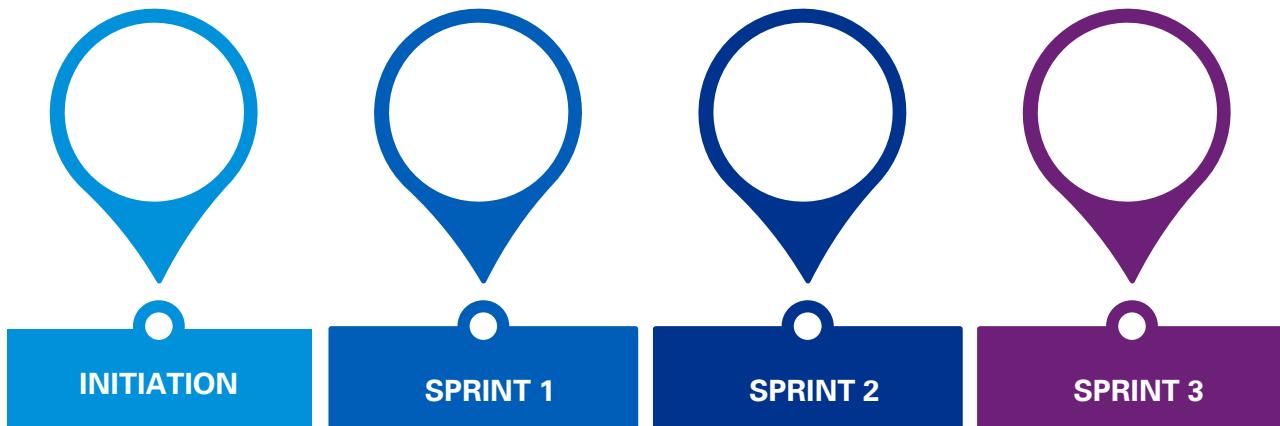
Points were assigned in collaboration with the development team, since they were the ones who knew how to do the work and how long it would take. These points were useful in the release and sprint planning sessions, and helped us balance work across the 3 sprints.

# Initiation week

- 01 **ASSIGNED PRODUCT MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.
- 02 **SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.
- 03 **CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME PAGE.
- 04 **CREATED INITIAL BACKLOG**  
FROM CONSULTATIONS WITH THE TEAM AND USERS.
- 05 **PLANNED THE RELEASE & DEFINED “DONE”**  
AND CREATED THE BACKLOG FOR SPRINT 1

We used the initial product backlog and deadline date to plan how we would develop the prototype.

In order to incorporate as much feedback as possible, we decided that 3 one-week sprints would allow enough opportunities for the independent users to review a working product.



The one week duration would also be long enough for the development team to produce something that was “done” and meaningful for users to interact with.

# Initiation week

- 01 **ASSIGNED PRODUCT MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.
- 02 **SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.
- 03 **CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME PAGE.
- 04 **CREATED INITIAL BACKLOG**  
FROM CONSULTATIONS WITH THE TEAM AND USERS.
- 05 **PLANNED THE RELEASE & DEFINED “DONE”**  
AND CREATED THE BACKLOG FOR SPRINT 1

We assigned high-level goals for each sprint, shown below. In order to build the highest quality product, we invested in the technical architecture and DevOps early.

## SPRINT 1

- DELIVER A WORKING LOG IN PAGE FOR THE AUTH. AND ADMIN USER
- COMPLETE USER INTERVIEWS, WIREFRAMES, DESIGN STYLE GUIDE
- COMPLETE MANDATORY TECH AND DEV OPS WORK
- *THIS WILL RAPIDLY INCREASE QUALITY AND VELOCITY FOR SPRINTS 2 & 3.*

## SPRINT 2

- INCORPORATE USER FEEDBACK AND FIX ISSUES FROM SPRINT 1
- FINISH THE CORE PRODUCT USER STORIES
- REFINE DEV OPS PROCESSES AS REQUIRED

## SPRINT 3

- INCORPORATE USER FEEDBACK AND FIX ISSUES FROM SPRINT 2
- FINISH REMAINING CORE PRODUCT STORIES AND AS MANY ADDITIONAL VALUE-ADDS AND FEATURES AS POSSIBLE



# Initiation week



01

## ASSIGNED PRODUCT MANAGER & TEAM

AND IDENTIFIED INDEPENDENT USERS.



02

## SELECTED AN ACCELERATOR

FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.



03

## CREATED VISION & KICK-OFF

MAKING SURE EVERYONE IS ON THE SAME PAGE.



04

## CREATED INITIAL BACKLOG

FROM CONSULTATIONS WITH THE TEAM AND USERS.



05

## PLANNED THE RELEASE & DEFINED “DONE”

AND CREATED THE BACKLOG FOR SPRINT 1

Based on the sprint goal and the story estimates, the product manager in agreement with the development team created a Sprint Backlog for sprint 1. We created a GitHub “project” in the repository to organize this backlog.

The screenshot shows a GitHub repository named 'kpmg-agile / ca-pqvp' which is private. The 'Projects' tab is selected, showing 6 projects. A search bar at the top has 'sort:name-asc' entered. A green 'New Project' button is visible on the right. The projects listed are:

- CA-PQVP**: General project work, updated 8 days ago.
- RFI Administrative Requirements**: Complete the supporting administration documentation, updated 4 days ago.
- Sprint 1 Backlog**: Sprint Goal: to deliver a shippable log in page for the auth user and admin user, to complete designs and all mandatory DevOps/FrontEnd/BackEnd work. This investment now will rapidly increase our velocity of user stories delivered for Sprint 2 & 3. Updated 5 days ago.

# Initiation week

- 01 **ASSIGNED PRODUCT MANAGER & TEAM**  
AND IDENTIFIED INDEPENDENT USERS.
- 02 **SELECTED AN ACCELERATOR**  
FOR THE TECHNICAL SOLUTION. BASED ON OUR ANALYSIS OF THE REQUIREMENTS.
- 03 **CREATED VISION & KICK-OFF**  
MAKING SURE EVERYONE IS ON THE SAME PAGE.
- 04 **CREATED INITIAL BACKLOG**  
FROM CONSULTATIONS WITH THE TEAM AND USERS.
- 05 **PLANNED THE RELEASE & DEFINED “DONE”**  
AND CREATED THE BACKLOG FOR SPRINT 1

As part of the Sprint 1 planning, the team created an initial definition of “done” and published this on the wiki. This is the criteria by which a developer’s code would be accepted into the master branch.

The screenshot shows a GitHub repository named 'kpmg-agile / ca-pqvp' with a 'Private' status. The top navigation bar includes links for Code, Issues (61), Pull requests (4), Projects (6), Wiki (selected), Pulse, Graphs, and Settings. The 'Wiki' tab is active, showing the 'Definition Of Done (DOD)' page. The page was last edited by Ben Rogers just now, with 7 revisions. The main content area contains the following text:

In order for work (work means an item in the Sprint Backlog) to be considered "done", it must pass the following conditions:

- All code and unit tests checked in to GitHub
- Automated unit testing passes
- Code has been reviewed and certified by Robert (via a Pull Request)
- User story acceptance criteria test passes (certified by Ben)
- Technical Debt has not increased (certified by Robert)

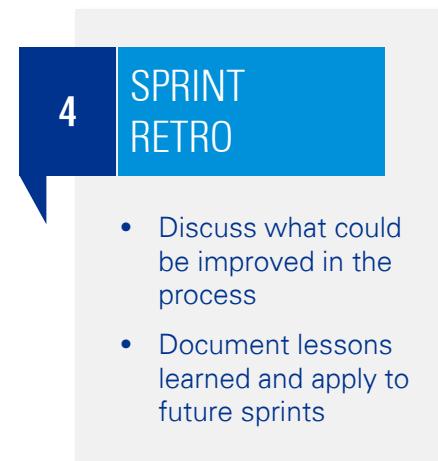
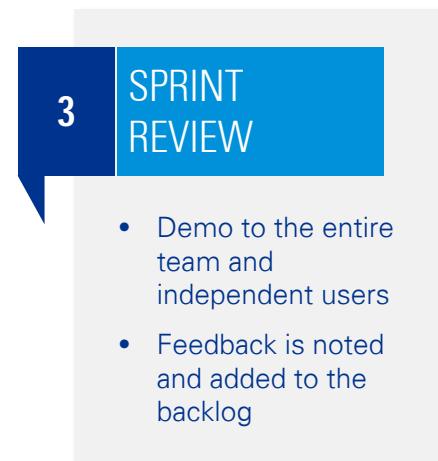
On the right side of the page, there is a sidebar with a 'Pages (32)' section containing a search bar and links to 'Home', 'Adherence to the US Digital Services Playbook', and 'Competitive Analysis'.

Instead of publishing long, elaborate checklists that developers must follow, we kept this definition simple for easier evaluation at the end of the sprint.

# During a Sprint: Meetings

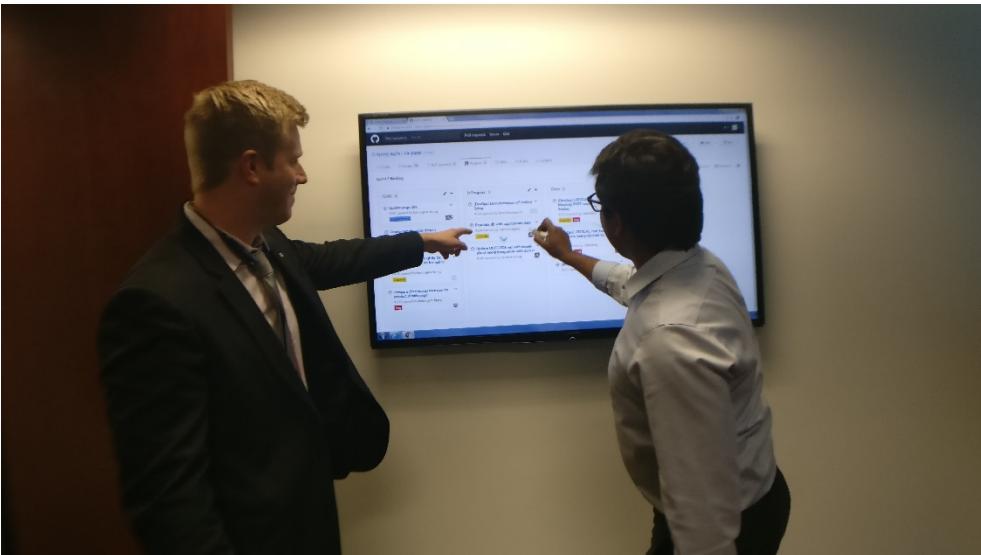
In each sprint, we held four types of official project meeting, in accordance with the Scrum framework. The frequency and days on which these meetings occurred is shown below:

Meetings	Sprint Day Number							
	1	2	3	4	5	6	7	1
✓ SPRINT PLANNING								
✓ DAILY STAND-UP								
✓ SPRINT REVIEW & DEMO								
✓ SPRINT RETROSPECTIVE								



# During a Sprint: Meetings

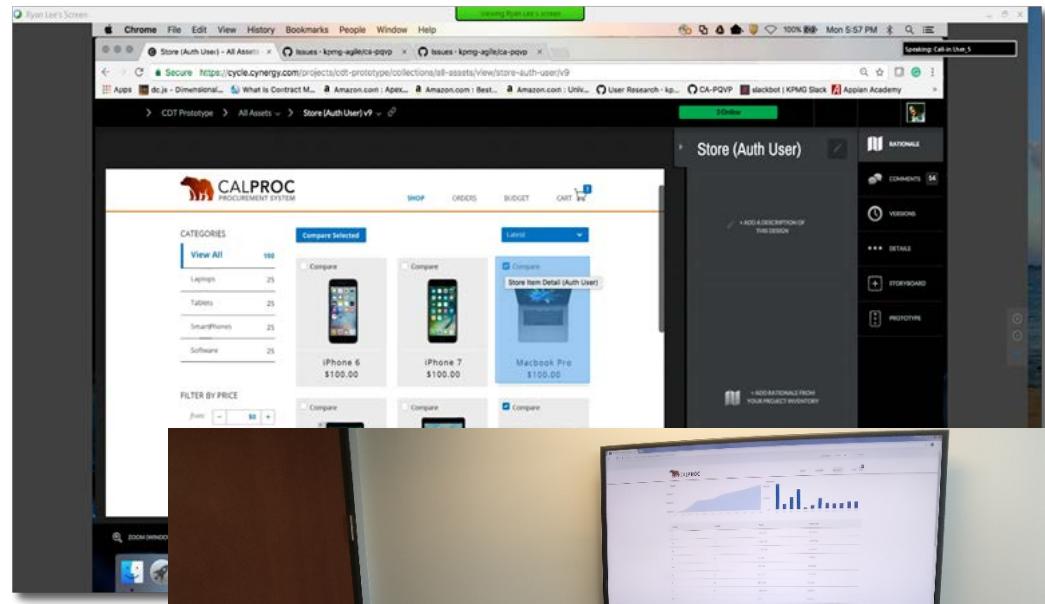
## Daily standup, Release and Sprint Planning meetings



Using GitHub Issues and Project Boards to plan and manage the release and sprints.

Inviting users to Sprint Review sessions.

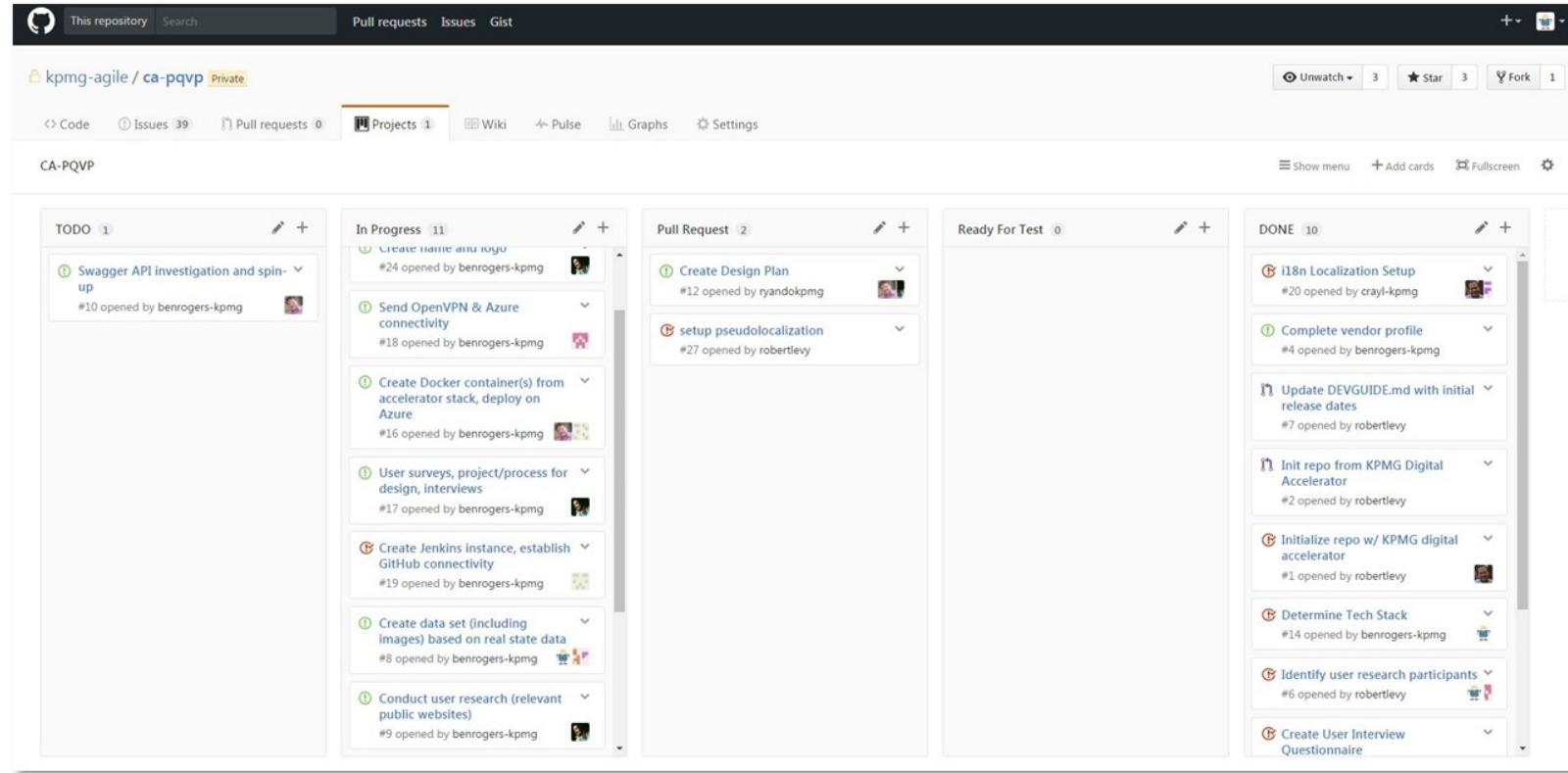
## Sprint Review meetings – in person and broadcast over WebEx



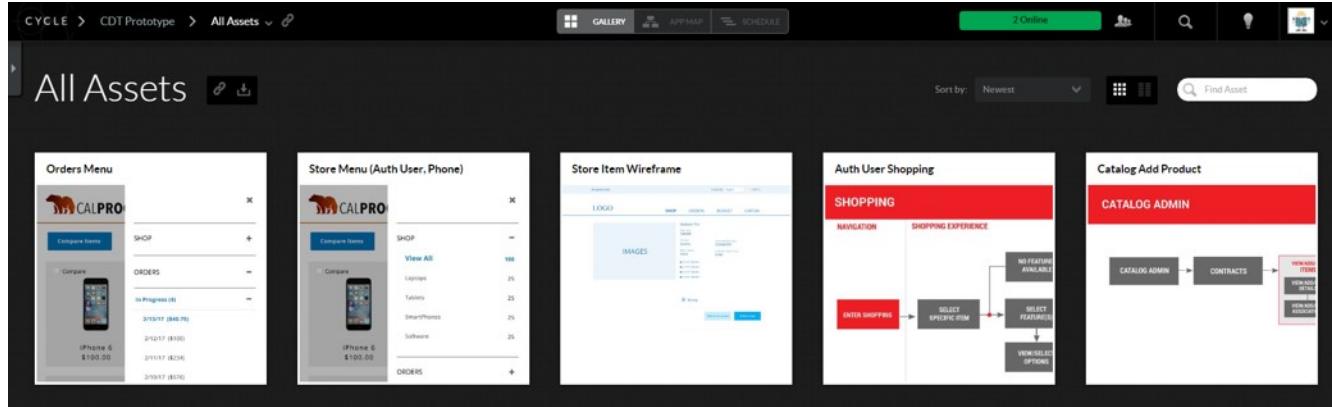
# During a Sprint: Managing work

Key to our agile approach was allowing the development team to self-organize. The development team are the most qualified to decide the order at which items in the sprint should be completed first. As such, they had full authority to pull items from the sprint backlog in any order that was necessary.

Again, minimizing the number of tools we had to use, the team leveraged GitHub's built-in project Kanban board to manage work throughout the sprint:



# During a Sprint: Design to development



We followed a user-centric design methodology to create and refine wireframes, screen flows and the site map. These assets were managed in KPMG's Cycle portal.

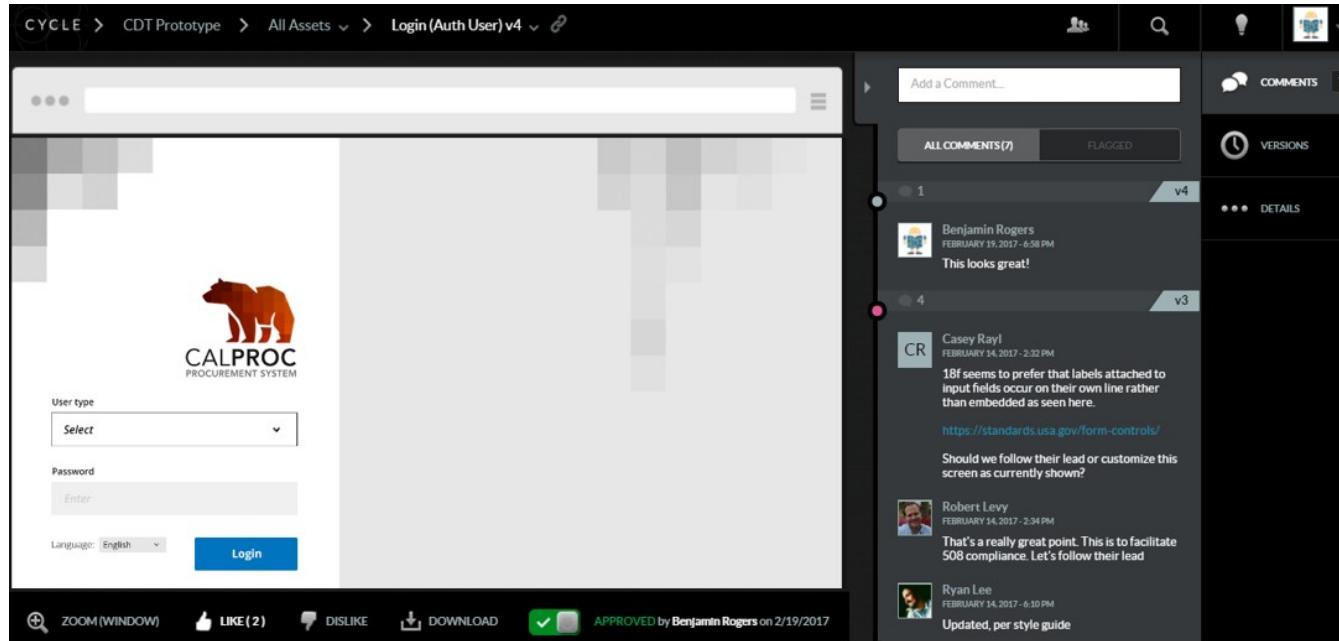
Cycle allowed the developers, product manager and designer to quickly and easily collaborate.

This was an essential step to make sure everyone was aligned, and that ultimately the finished product would match the design.



# During a Sprint: Design to development

In Cycle, team members could leave comments on each wireframe. The designer modified the wireframes based on feedback from user review sessions, developers and product manager.



Ultimately, the product manager would approve a wireframe once users were satisfied, and developers verified it was possible to build. Once approved, the development team would start building the screen (the front end and back end services).

This process was essential to promoting transparency, minimizing rework and incorporating as much user feedback as possible into the final product.

# During a Sprint: Design to development

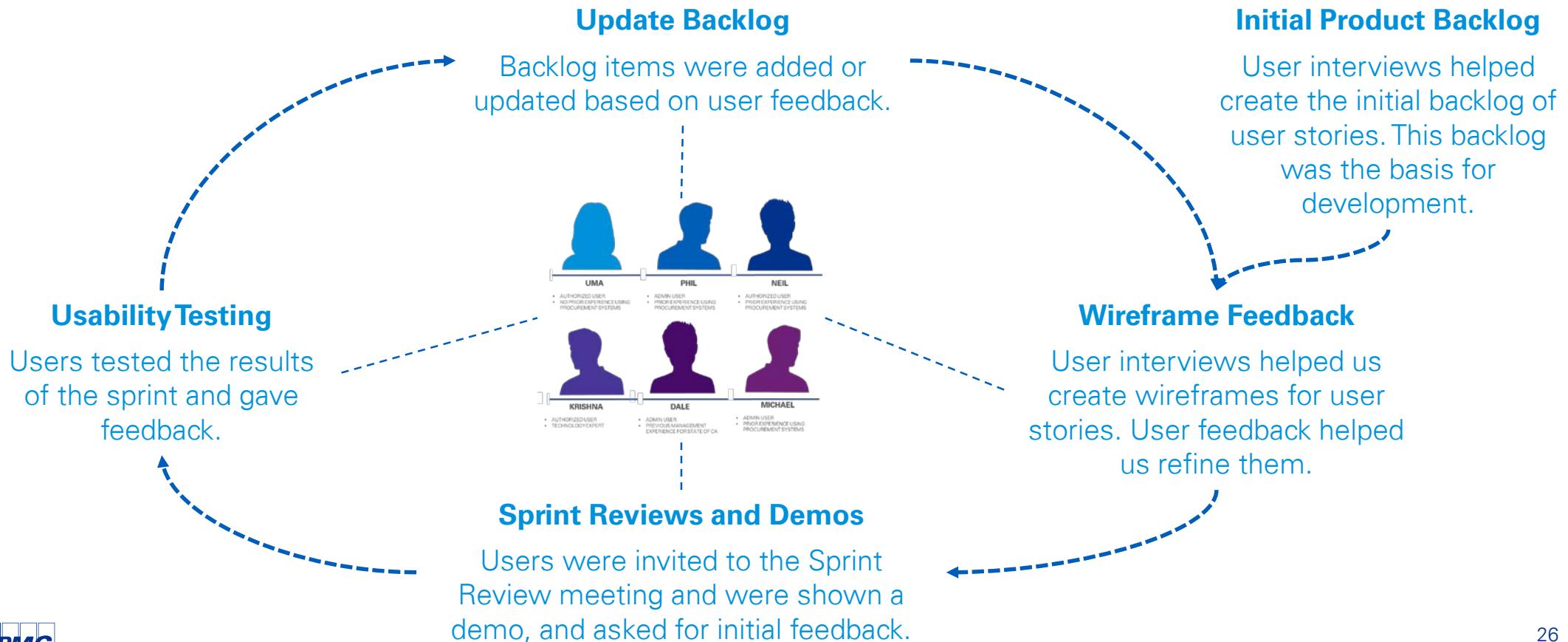
For more information about our methodology and Cycle, please watch the following short video:



<https://www.youtube.com/watch?v=ExHIOI7m0U0>

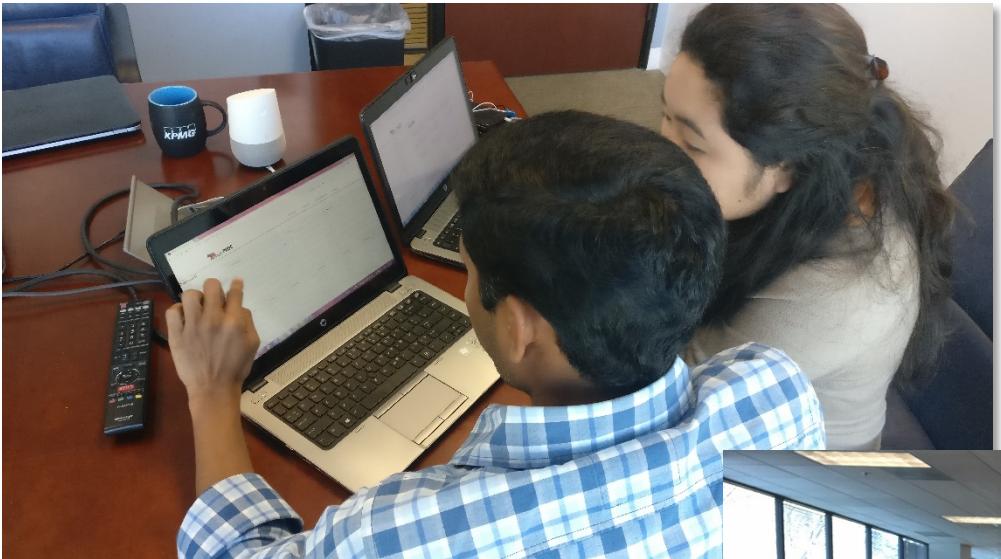
# During a Sprint: User interaction

We involved the independent users ("users") early and often. In our experience, users should be involved in helping the product manager create the initial backlog and user stories. This provides a solid foundation for the early sprints and minimizes later rework. Below is an overview of how the users were involved throughout our development.

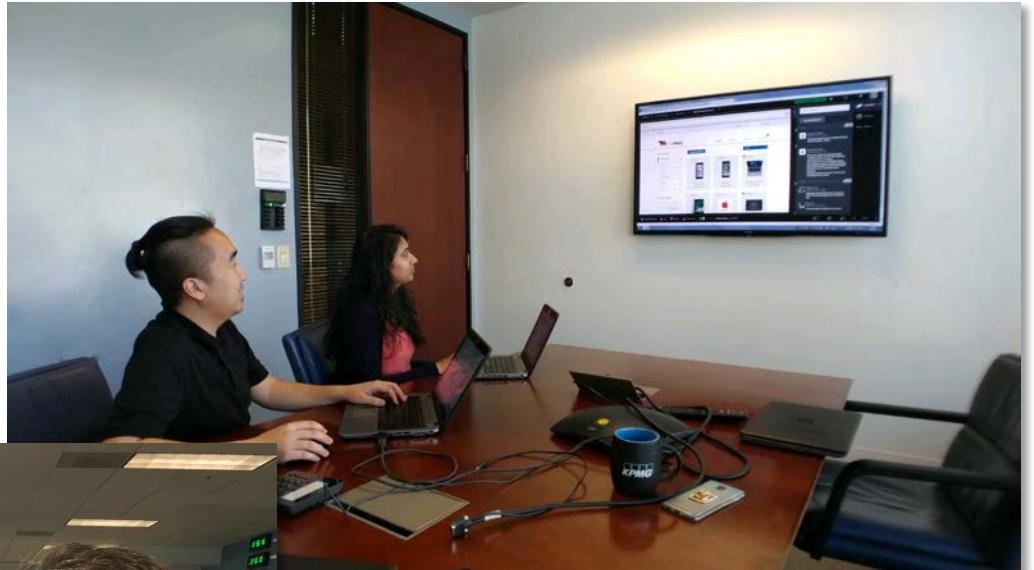


# During a Sprint: User interaction

**Usability Testing and Interviews**



**Wireframe Feedback**



**Former state employee reviews**

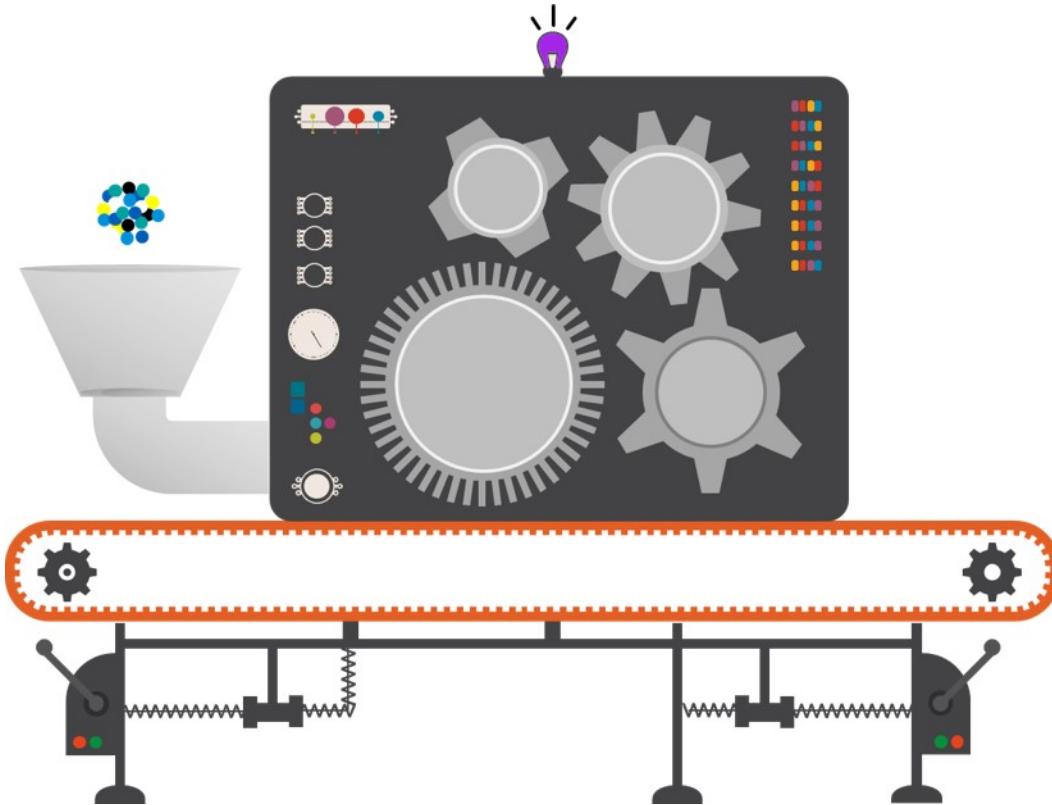
# During a Sprint: Development to production (1 of 3)

We achieved Continuous Integration using Jenkins and GitHub.

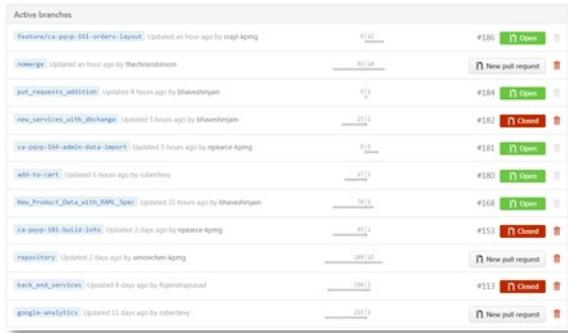
We maintained quality with automated unit tests and tools such as Karma, Protractor, Istanbul and ESLint.

Key to our approach was a clearly defined and understood process amongst the product manager, code development and DevOps teams.

The next two slides show a summary of this process end-to-end, and how a code check-in will ultimately reach the production environment.



# During a Sprint: Development to production (2 of 3)



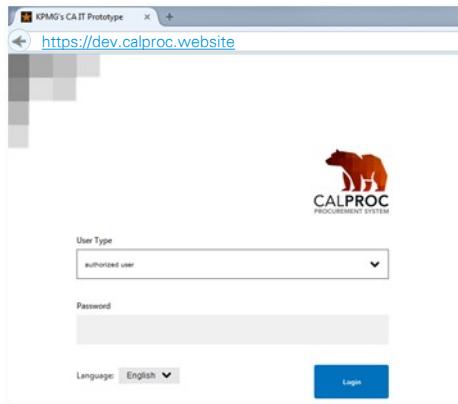
DEVELOPERS WORK ON FEATURE BRANCHES WITHIN GITHUB.



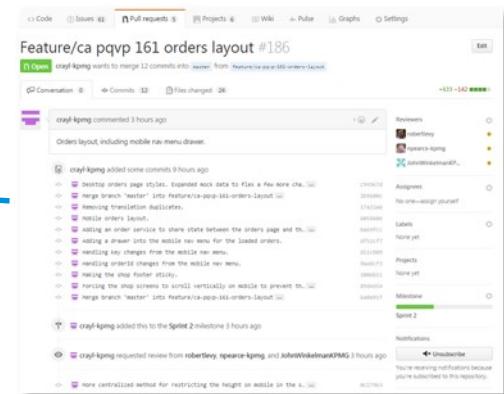
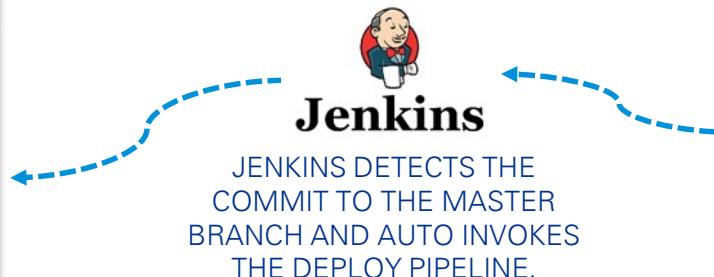
JENKINS, THROUGH GITHUB INTEGRATION, DETECTS THE COMMIT AND AUTO INVOKES THE UNIT TEST PIPELINE.



JENKINS BUILDS THE PROJECT AND REPORTS THE RESULTS OF THE UNIT TESTS.



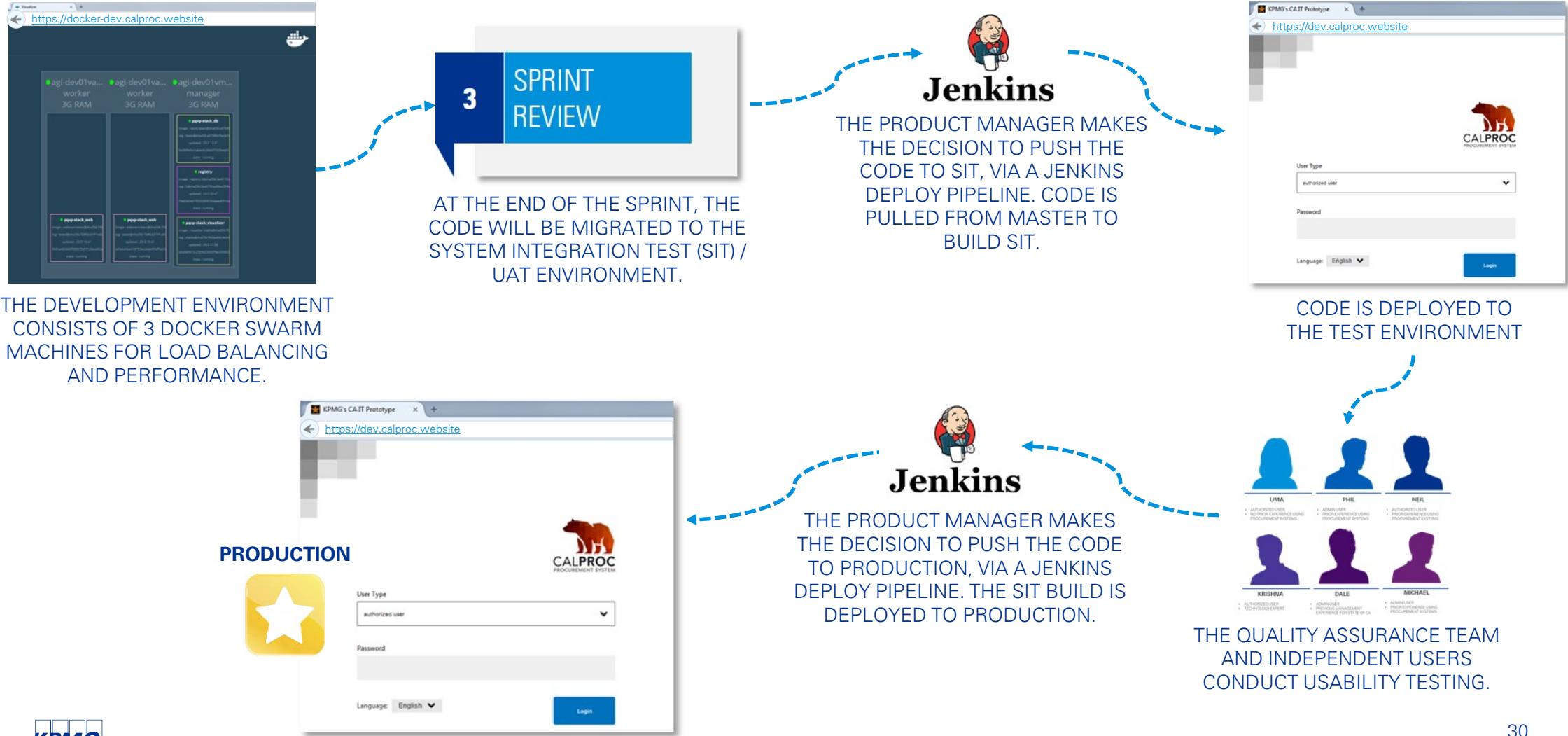
CODE IS DEPLOYED TO THE DEVELOPMENT ENVIRONMENT



IF THE UNIT TESTS ARE SUCCESSFUL, THE DEVELOPER SUBMITS A "PULL REQUEST".

THEIR CODE IS REVIEWED BY AT LEAST THE TECHNICAL ARCHITECT BEFORE IT IS MERGED INTO THE MASTER BRANCH. THIS KEEPS QUALITY HIGH WITHOUT SACRIFICING VELOCITY.

# During a Sprint: Development to production (3 of 3)





# The team



# Our team

## 1. Product Manager

Ben Rogers, CSM

## 2. Technical Architect

Robert Levy

## 3. Interaction Designer/User Researcher/Tester

Ryan Lee

## 4. Writer/Content Designer/Strategist

Cory Fritzsching

## 5. Front End Developers

Nick Pearce (1)  
Casey Rayl (2)

## 6. Backend Developer

Sandeep Pedditi

## 7. DevOps Engineers

Chris Robinson (1)

## 8. Agile Coach

Matt Kwong, PSM

## 9. Business Analyst

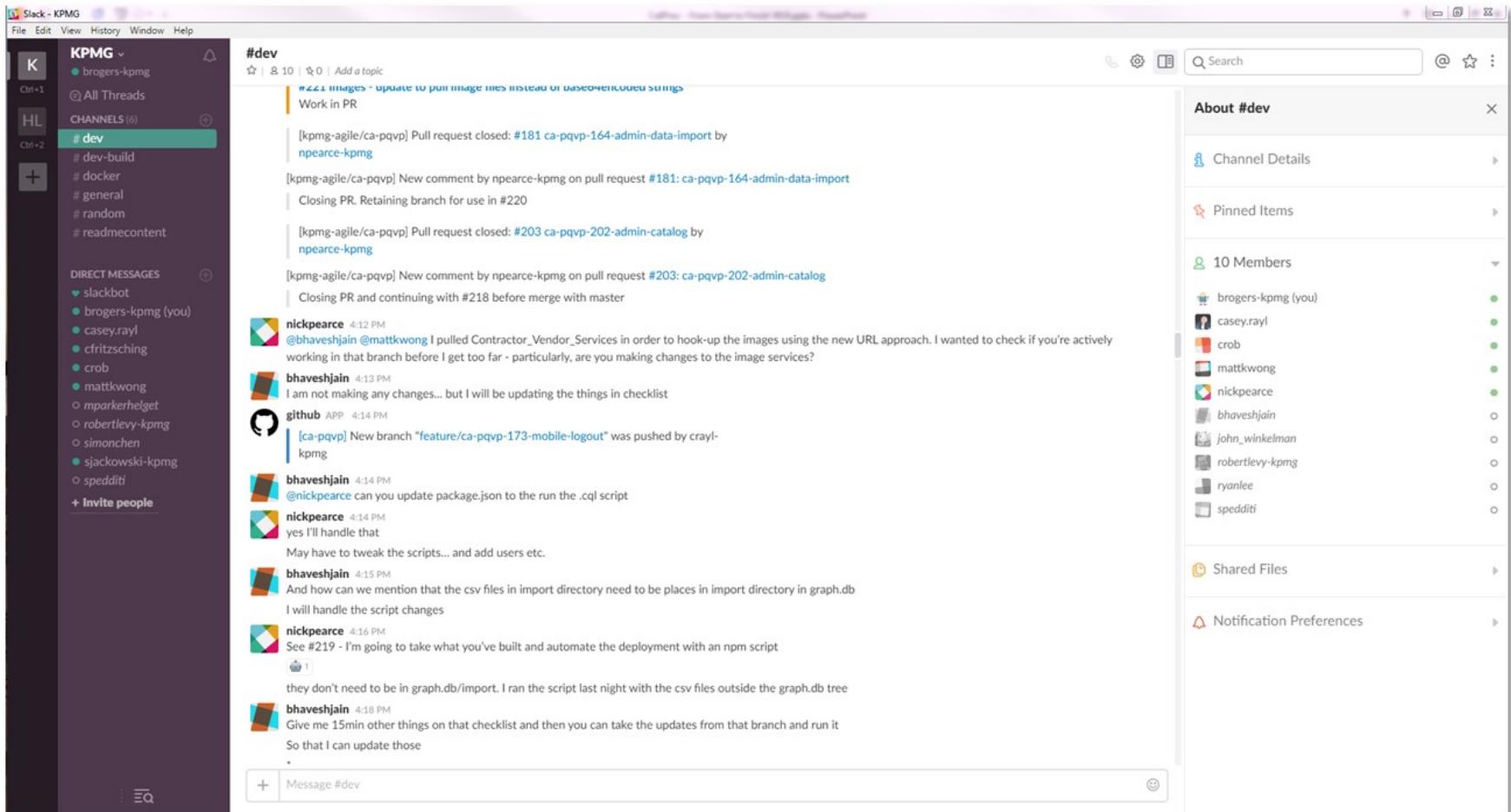
Stacy Lee, CSM

## 10. Business Analyst

Simon Chen, CSM

# Team communication

The team used Slack to stay in constant communication. We integrated Slack with Jenkins and GitHub which allowed us to rapidly collaborate on issues, builds, testing and other tasks.





# User-centric design and usability



# Where human insights meet digital thinking

To thrive in the digital world, organizations need to:



**Focus on the problem first**



**Design for people, not processes**

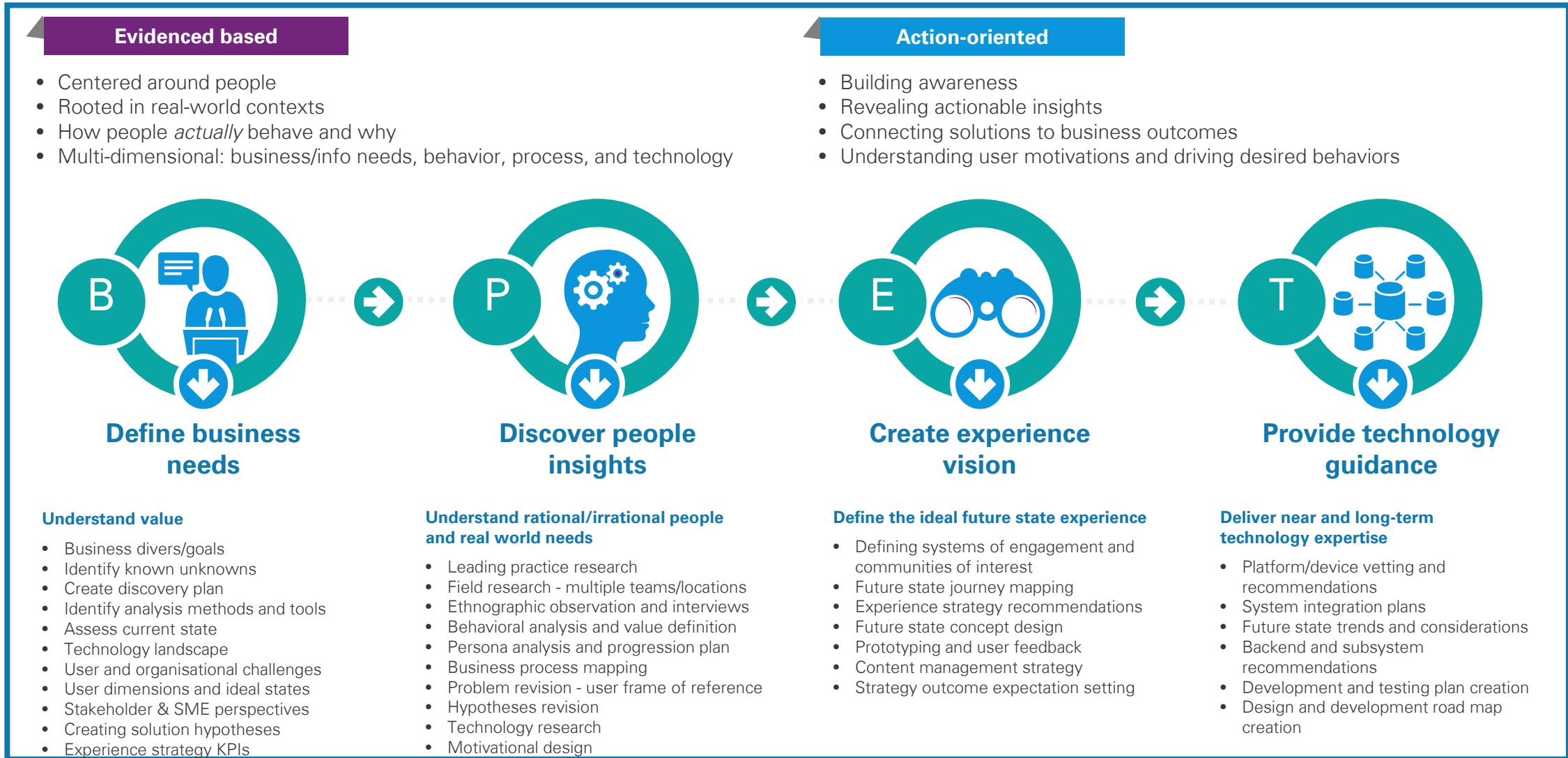


**Deliver small, fast, and often**



**Always look for improvements**

# Our approach and method

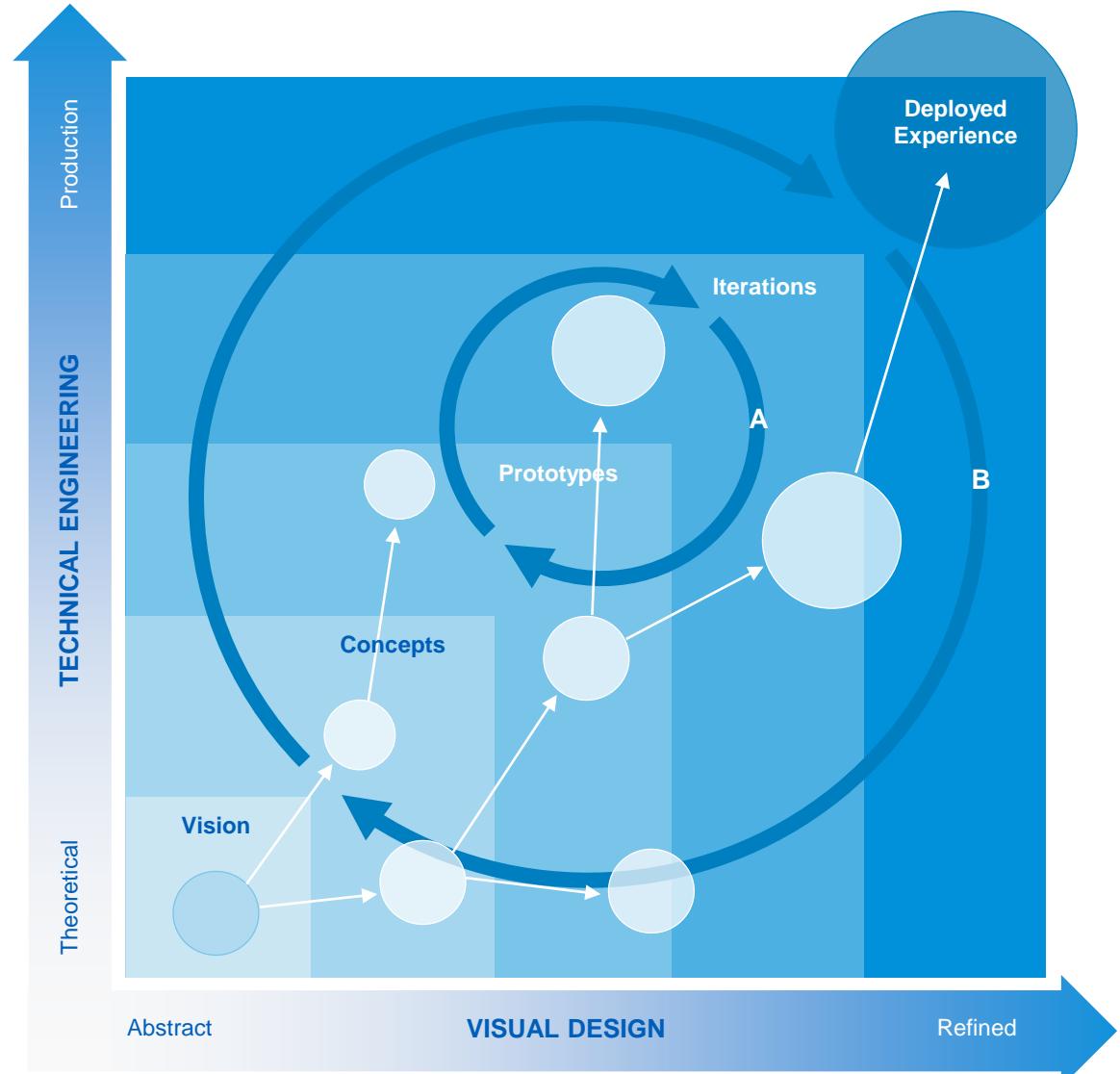


# Progressive Realism

While many organizations are familiar with the concept of agile thinking and the importance of innovation, few understand Progressive Realism and the benefits it brings to the transformation process. Progressive Realism is a crucial mindset to cultivate when approaching any project, especially at the point where concept becomes prototype. **If Design Thinking is about looking broadly at potential problems and solutions, Progressive Realism is about evolving solutions with a more pragmatic approach to progress.**

Progressive Realism acknowledges that businesses sometimes fail to execute ground-breaking initiatives, not because they lack ideas, but because of the difficulty of achieving internal buy-in. **The availability of higher-fidelity prototypes at an early stage makes buy-in from stakeholders and partners easier to achieve.**

Progress is realized quickly and agile drift is minimized by **iterating in both small and large cycles**—implementing user insights while still targeting a deployed experience, which evolves over time.



# Creating a human-centered digital experience

## VISION STATEMENT

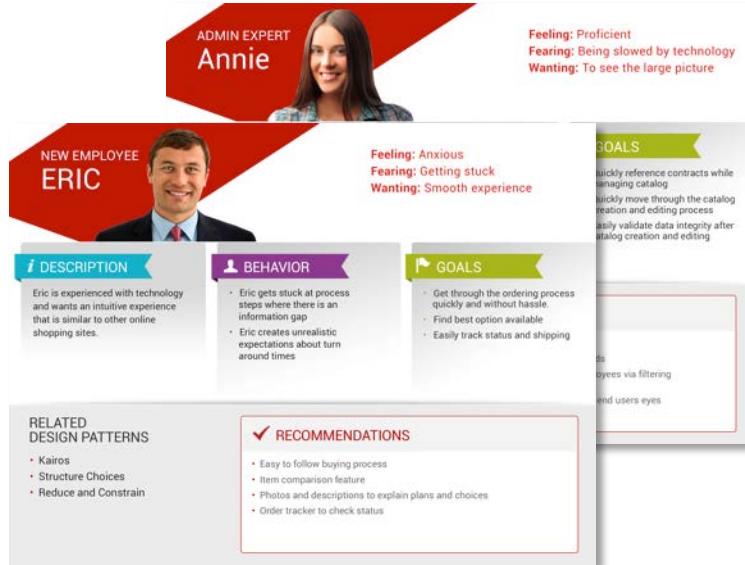
For government employees within California who need to procure computing hardware, software and related services from pre-established state contracts, the "CalProc" procurement system provides ordering, tracking and analysis features. Also, it allows employees of the state's leading purchasing organization the ability to publish product and service information and then track and analyze order data.

Unlike other services, our product provides a very simple and intuitive platform for users.



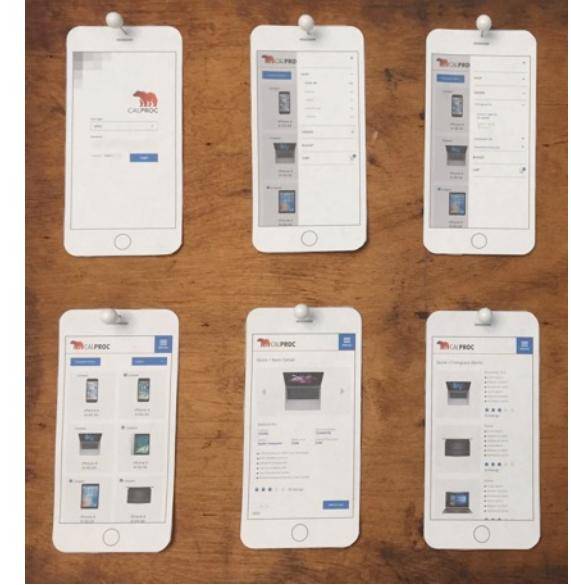
## Contextualize

- Business needs/goals
- Product strategy
- Technical assessment
- Value definition



## Empathize

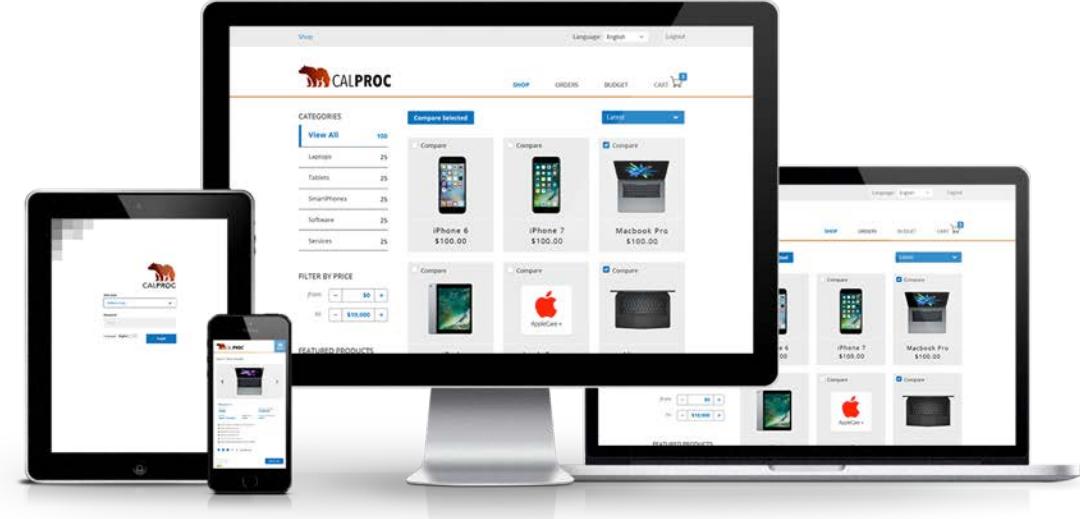
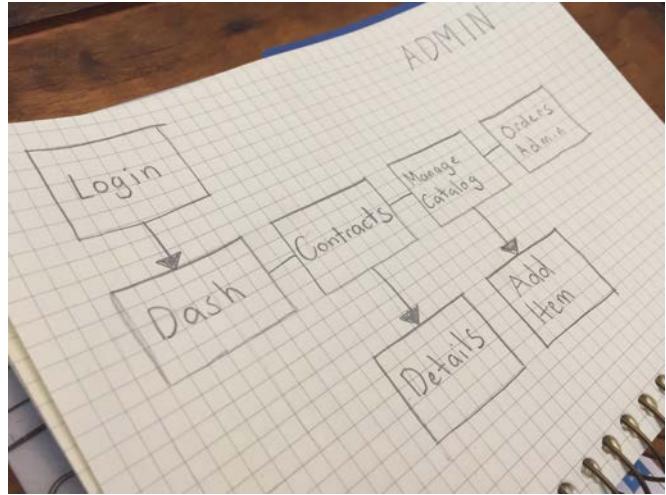
- User research
- Behavioral analysis
- Journey mapping
- Persona creation
- Feature ideation



## Prototype

- Conceptual design
- Motivational design
- Interactive prototyping
- User validation
- Stakeholder alignment

# Creating a human-centered digital experience



Expand

Information architecture  
Workflow mapping  
Visual design  
Style guides  
Road mapping



Implement

Technical architecture  
Agile development  
Experience-driven development



Evolve

Continuous improvement  
Iterative releases  
Analytics-based refinement  
Feature prototyping



# Competitive Analysis

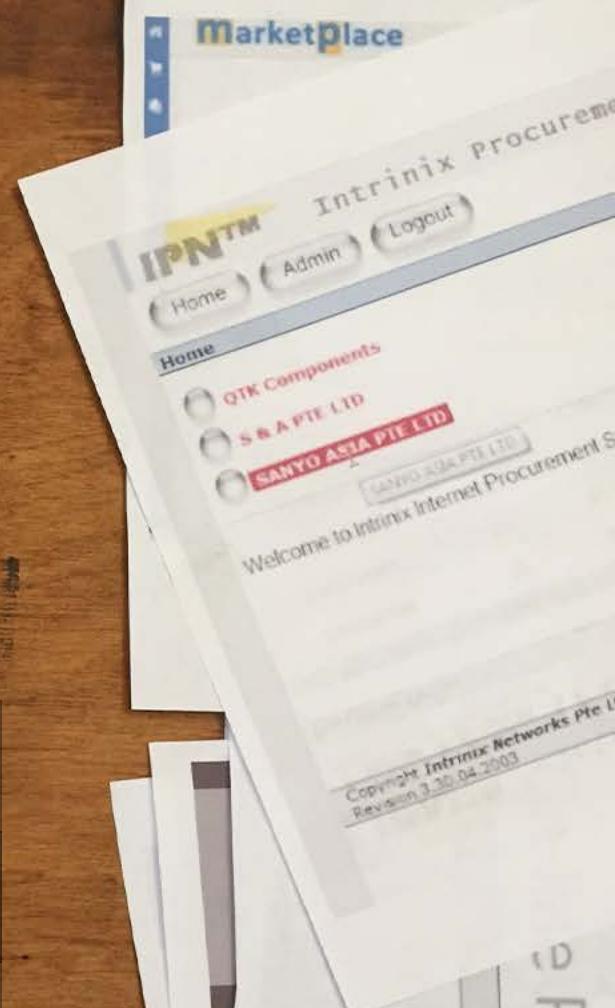
Our initial user research focused around looking at other procurement systems in the market.

After analyzing other systems, we came to the conclusion that users had suggestions to improve the user experience, and wanted them to feel more like consumer buying sites.

	Images/ Descriptions	User Experience	Functionality
Ariba	2	2	2
IPN	1	1	2
Jaggaer SciQuest	1	1	3

1=Poor, 2=Average, 3=Excellent

**Experience Design & Engineering**



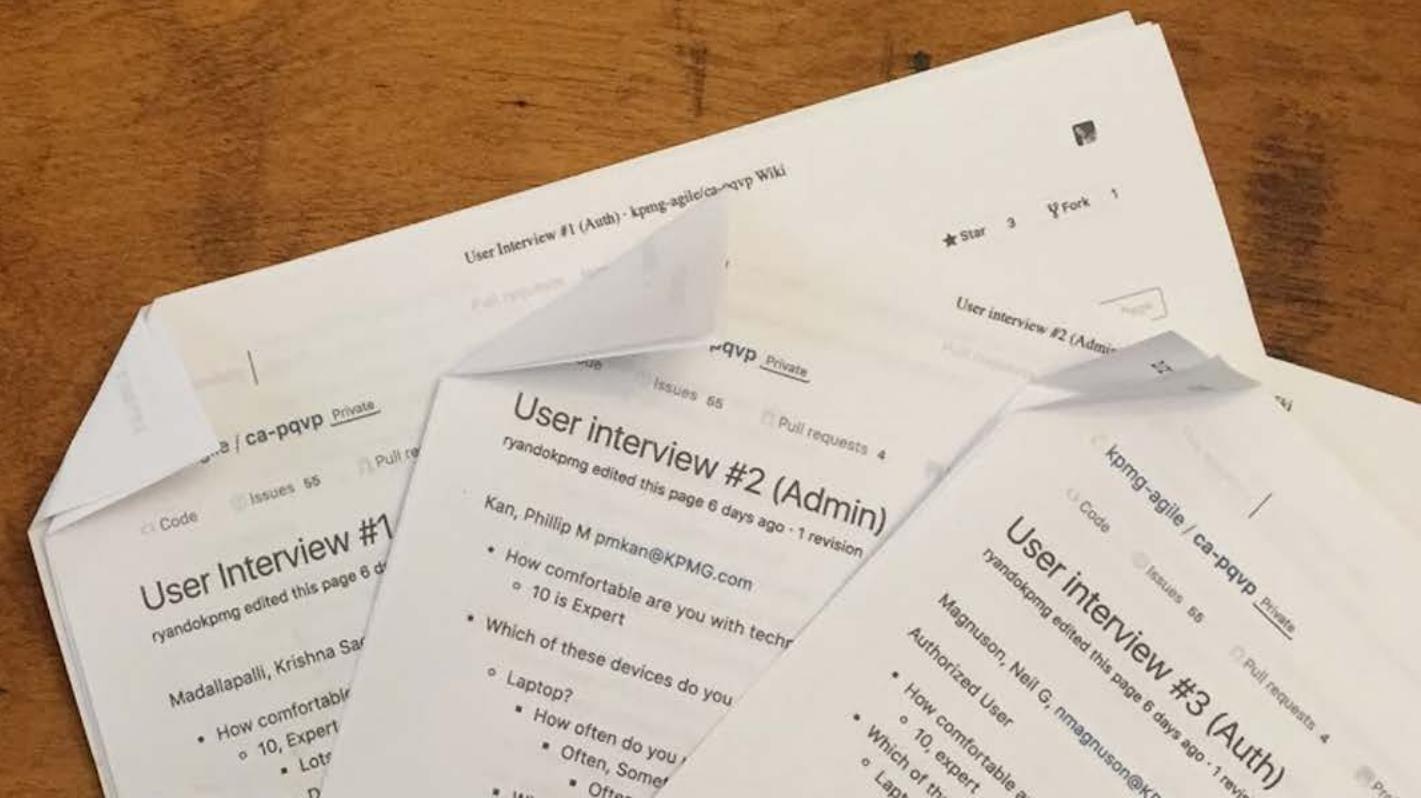


# User Interviews

Our initial user research focused around user interviews. We began this process by creating a loose script that would help us establish user types, device preferences, habits, behaviors, skills, fears, and processes.

We then used this script to conduct multiple interviews. This allowed us to identify two main user types and establish process flows.

**Experience Design & Engineering**

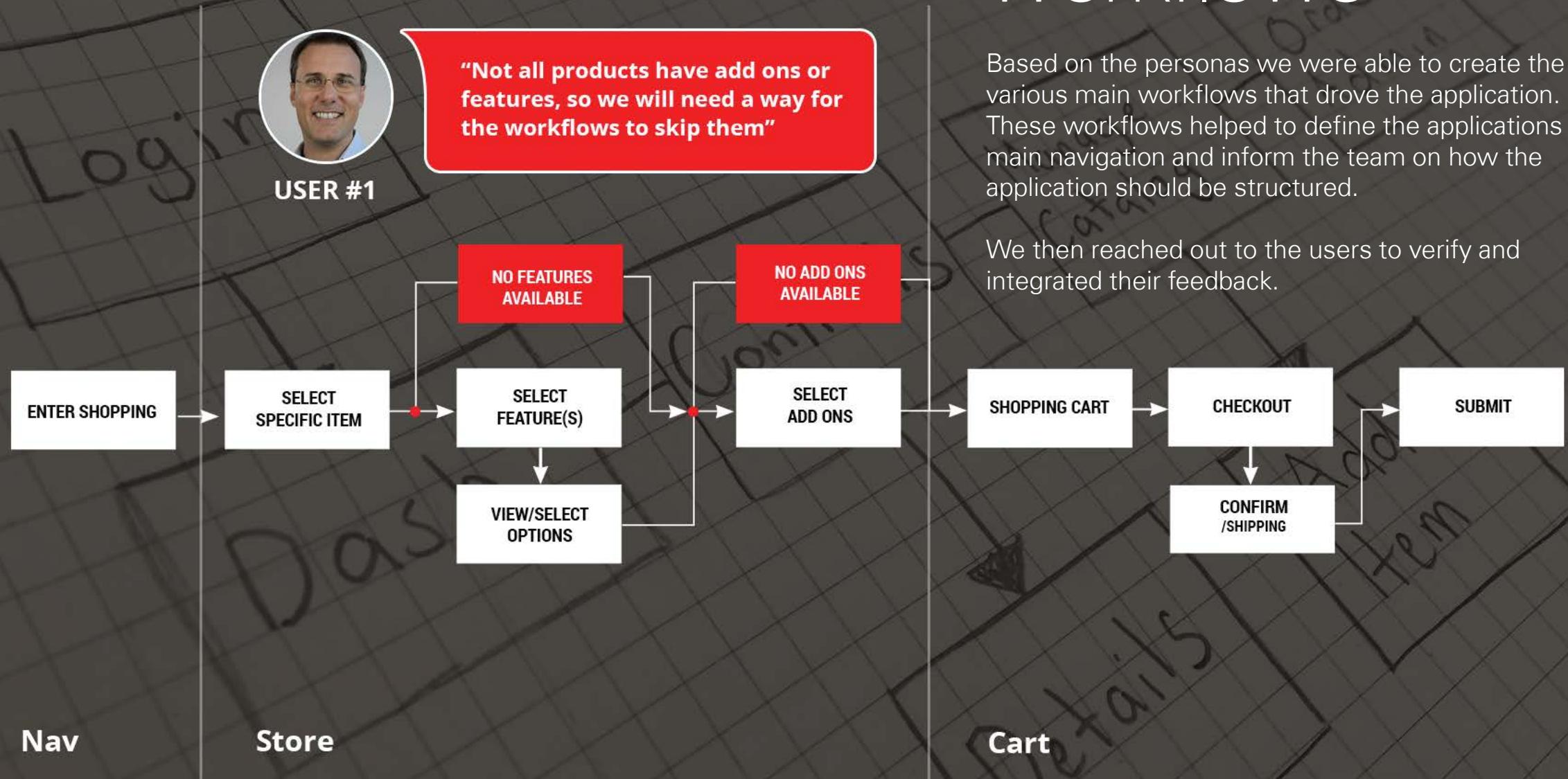


# Personas

After conducting the interviews, we created fictional users that were a summation of the 2 main types of users. These 2 users consisted of an admin and employee. These were then referenced throughout the design process for both workflows and user features.



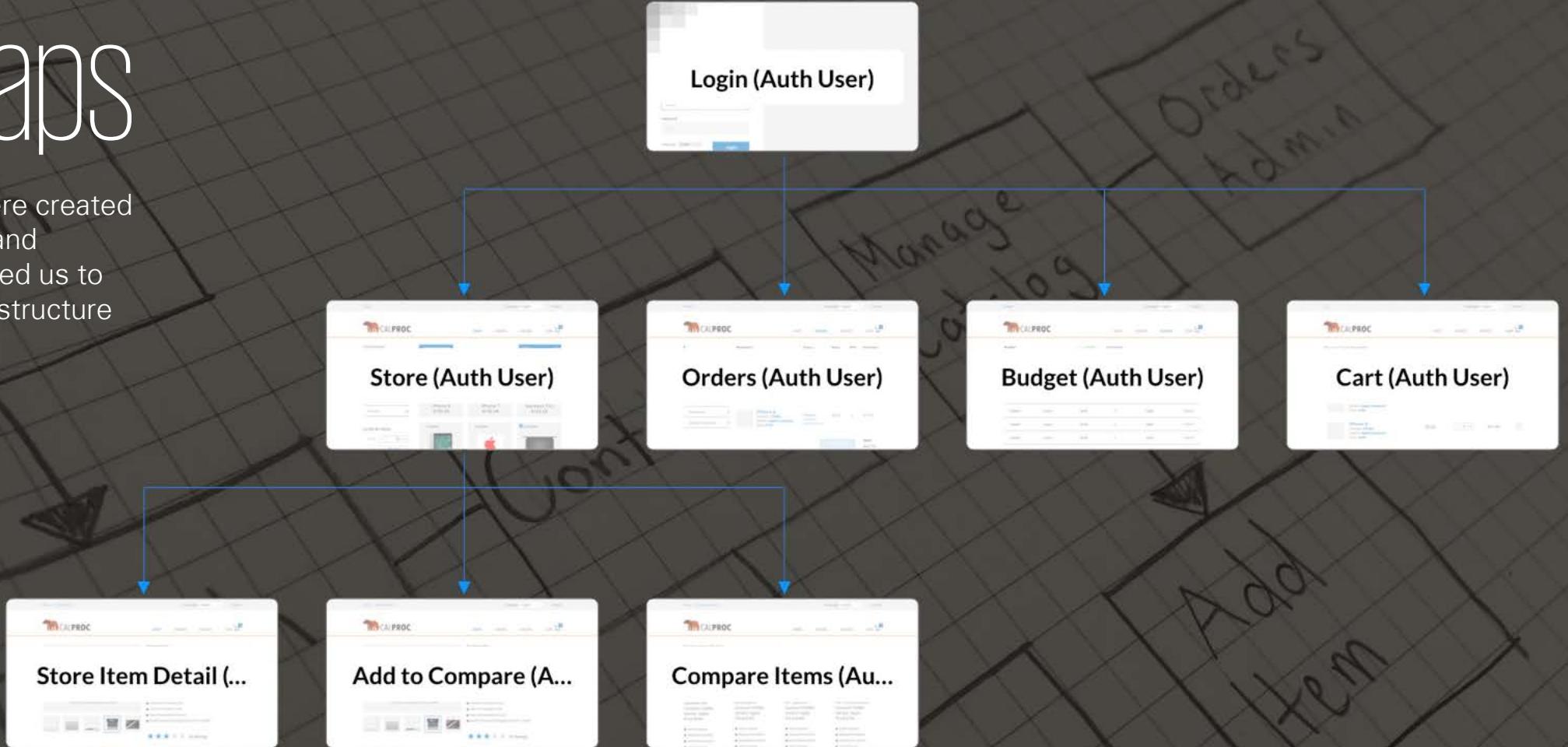
# Workflows





# App Maps

Application maps were created after defining users and workflows. This helped us to form the application structure and cover all of the functionality.

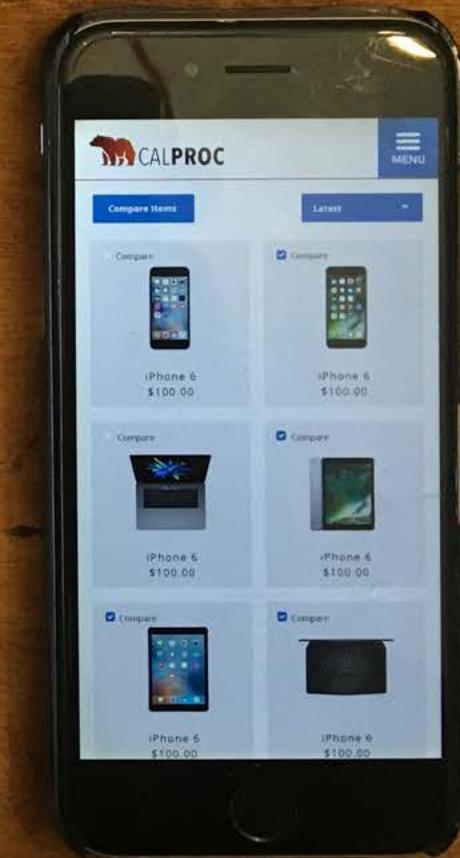
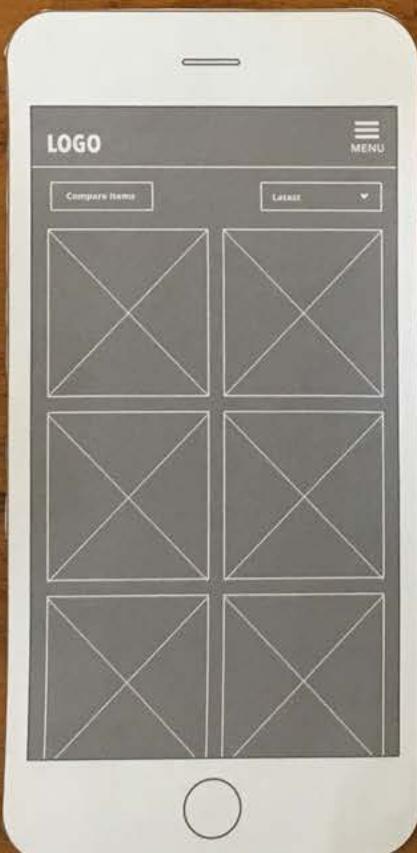
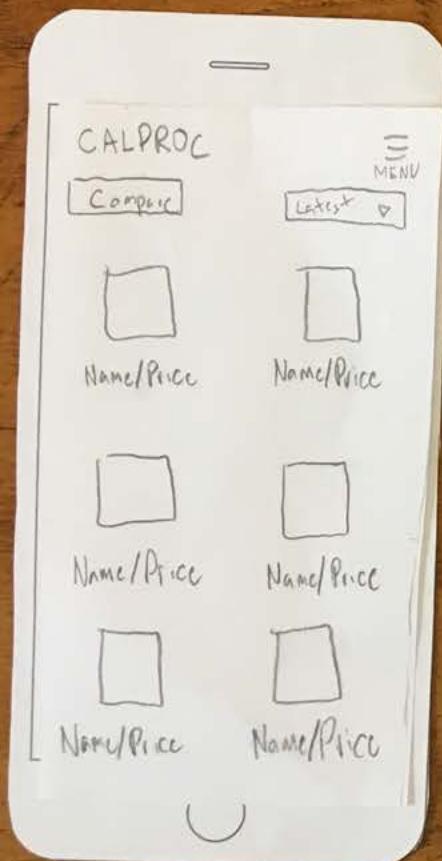




Experience Design & Engineering

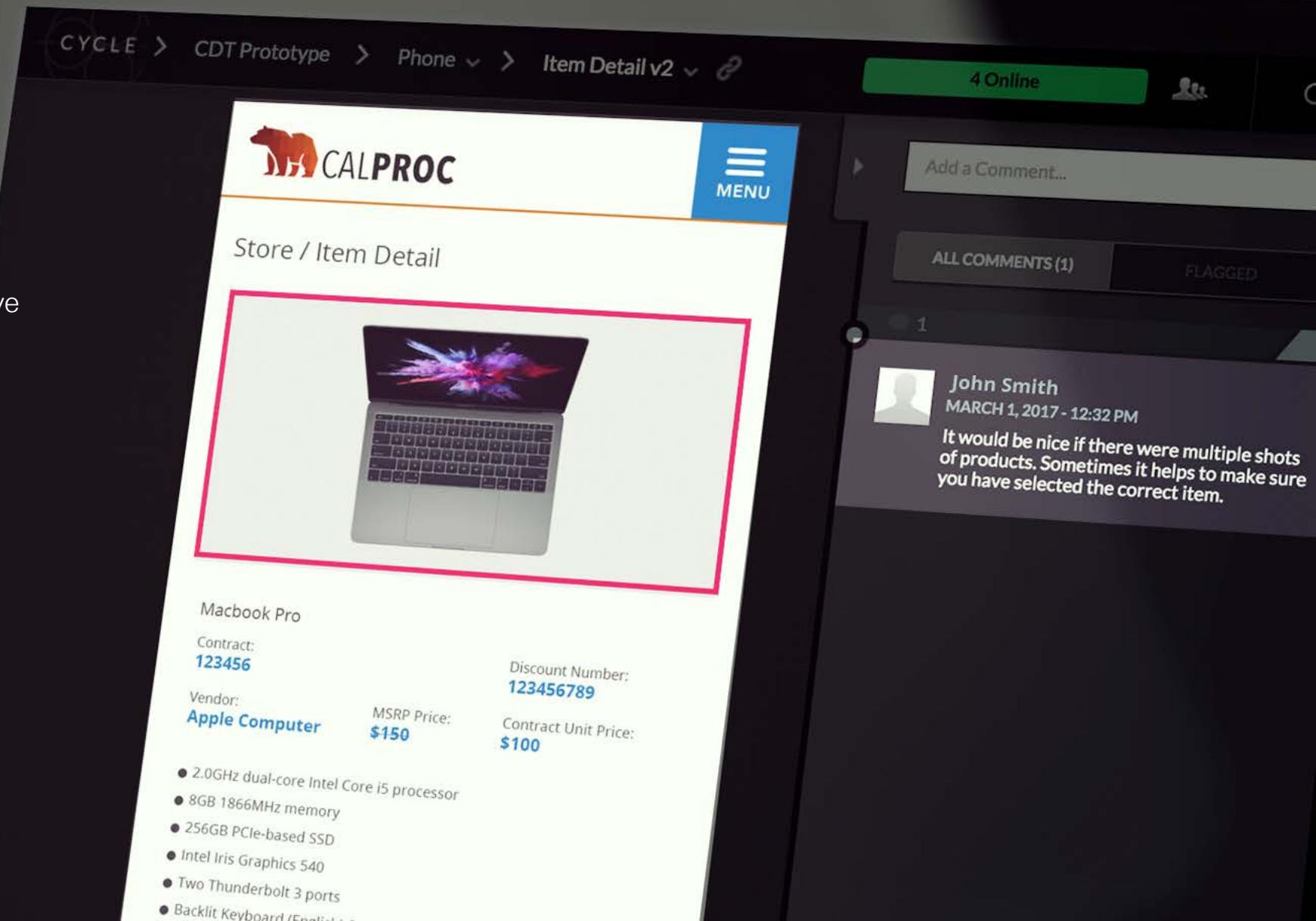
# Rapid Prototyping

Rapid Prototyping allowed us to evolve our work into a polished design quickly.



# Feedback

Cycle was utilized throughout this entire process. We were able to build an iterative prototype based on feedback from users and our team.



The screenshot shows a mobile application interface for 'CALPROC'. At the top, the navigation bar includes 'CYCLE > CDT Prototype > Phone > Item Detail v2'. On the right side of the header, there is a green button labeled '4 Online' and a user icon. Below the header, the main content area is titled 'Store / Item Detail' and features a large image of a Macbook Pro with a red border around it. To the right of the image, the product name 'Macbook Pro' is displayed, along with its contract number '123456', vendor 'Apple Computer', MSRP price '\$150', discount number '123456789', and contract unit price '\$100'. A bulleted list of product specifications follows: 2.0GHz dual-core Intel Core i5 processor, 8GB 1866MHz memory, 256GB PCIe-based SSD, Intel Iris Graphics 540, Two Thunderbolt 3 ports, and Backlit Keyboard (English). On the right side of the screen, there is a comment section with a message from 'John Smith' dated 'MARCH 1, 2017 - 12:32 PM' stating: 'It would be nice if there were multiple shots of products. Sometimes it helps to make sure you have selected the correct item.'

A close-up photograph of a person's hands writing in a spiral-bound notebook. The person is using a pencil to draw a grid and write some notes. The notebook has a dark cover and is open to a page with horizontal lines. The background is slightly blurred.

Disruption happens by design.

*The value of success is greater than the cost of failure.*

*With the clockspeed of innovation, you can't afford to move slowly.*

*To keep up you must: learn early, test assumptions, and pivot.*

# Design Thinking and Rapid Prototyping

can help organizations move faster, make decisions quicker that meet customer needs, bring transparency and purpose to agile methods, and align organizations on vision.

Validate  
Desirability



Gain  
Alignment

## **Customer Expectations**

Heightened customer expectations from consumer to enterprise, raise the bar and deliver at the speed of the market

## **Speed of Agile Development**

As agile methods become more deeply integrated, problems need to be aligned with the end to end agile delivery model, from sprint to DevOps

## **Industry Standards and Regulations**

While the speed increases, policies and security needs for auditing and risk in regulated industries are not decreased

## **Need for Rapid Evolution**

As the clock speed increases there becomes more a demand for competing needs – evolutions & time

## **Organizational Constraints**

Alignment and constraints of an organization often kill ideas before they can be tested or evolved

# What does Rapid Prototyping look like?

## 1. Hypothesize

Think fast. Dream big.  
"How might we..."  
(1-2 days)

## 2. Make

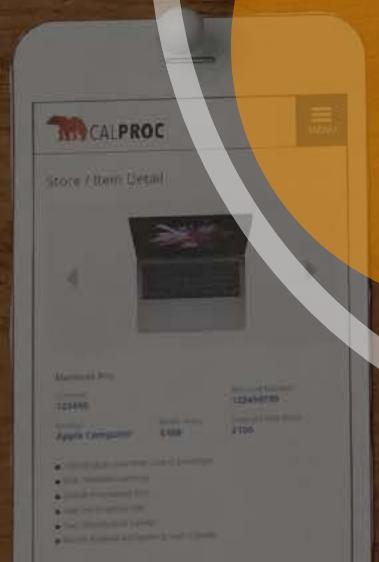
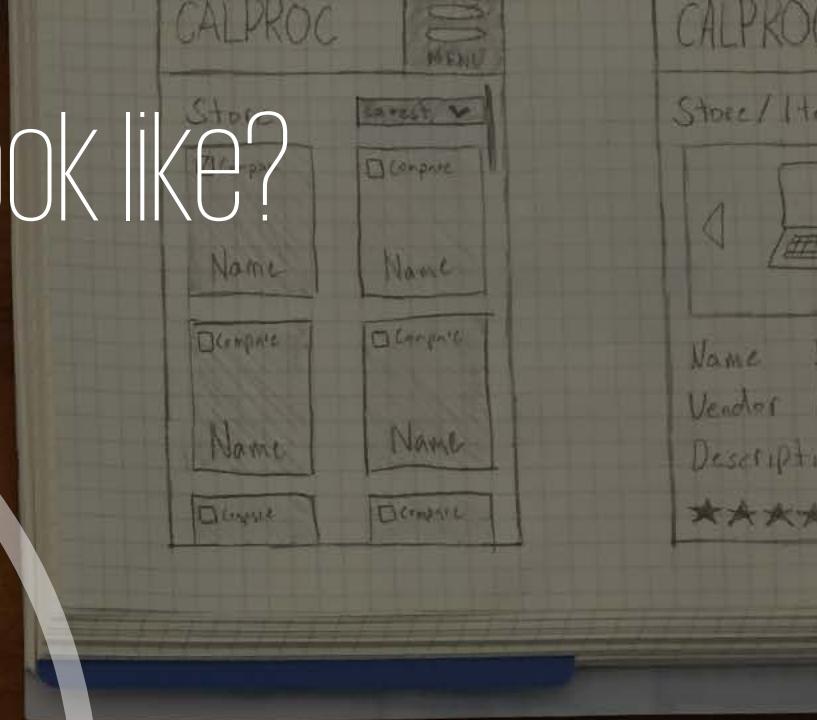
Fidelity over function.  
Pixels and coffee.  
(2-3 weeks)

## 3. Share

Gain feedback.  
Sell the dream.  
(1-2 days)

## Align

Gain agreement  
on vision and  
team buy in.





# Technical Architecture



# Technical Architecture overview

## Docker Containers Deployed to Azure

## KPMG Infrastructure

### Web App

- Multiple load balanced nodes
- [Swagger](#) / [SwaggerHub](#) / [RAML](#) API specs
- UI layer
  - Angular 2
  - [USWDS](#) UI components
- Webpack configured for ES6 and hot reload
- i18next for localization
- Custom RAML webpack loader for auto-generated wrappers around API specs
- Service Layer
  - NodeJS & Express
  - Osprey for mocking and validation of API requests/responses against API specs

### Neo4J Database

- Behind firewall, only accessible via the service layer
- Sample data converted to CSV format, supplemented with additional data, denormalized, and then imported into this open source graph DB

### Docker Visualizer

### GoAccess

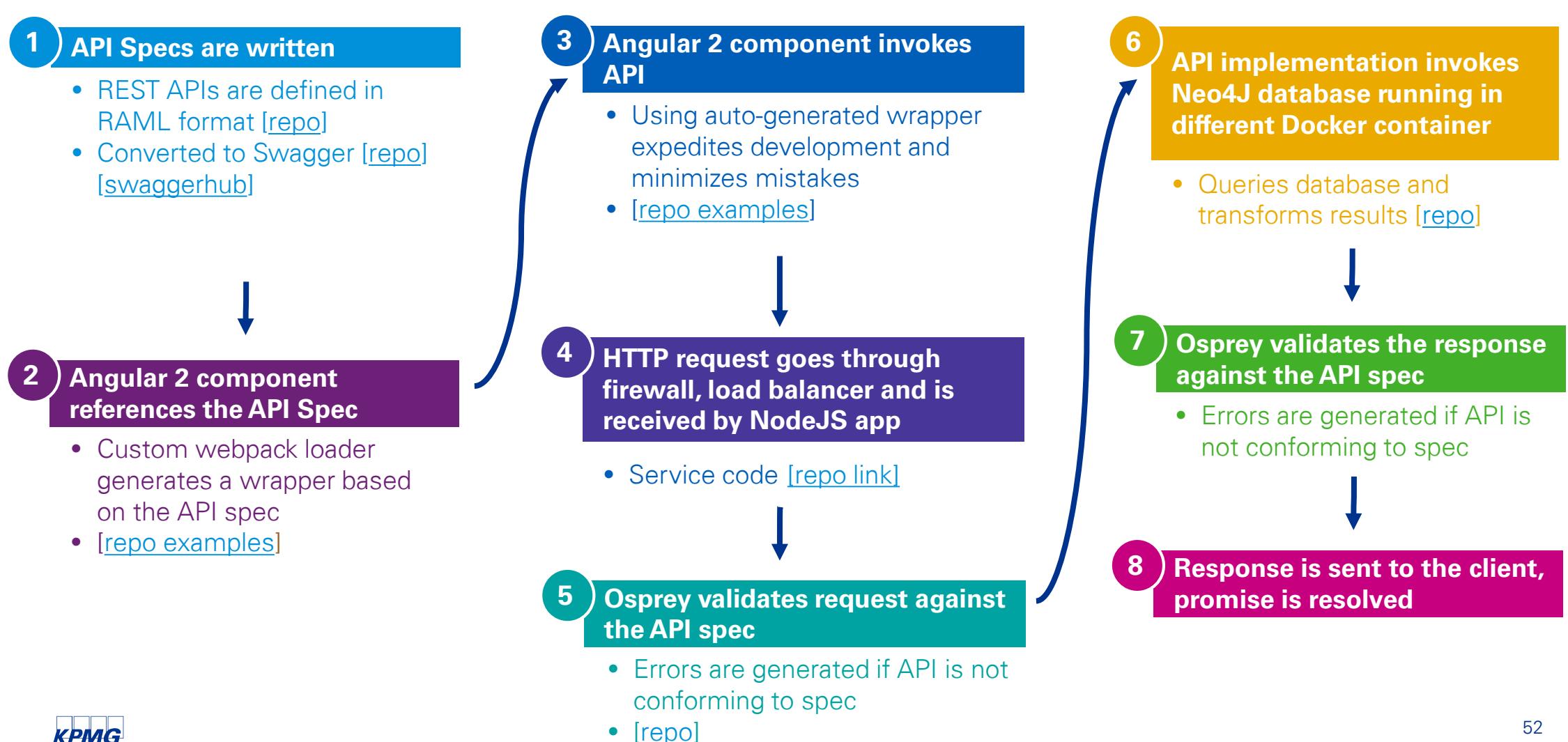
### Nagios

- Monitoring tools used to track and alert on the health and usage of the application
- [Production Docker Visualizer](#)
- [Production Nagios](#) (login: ca-pqvp-guest / kpmgagile#2017)
- [Production GoAccess](#)

### Automated Testing

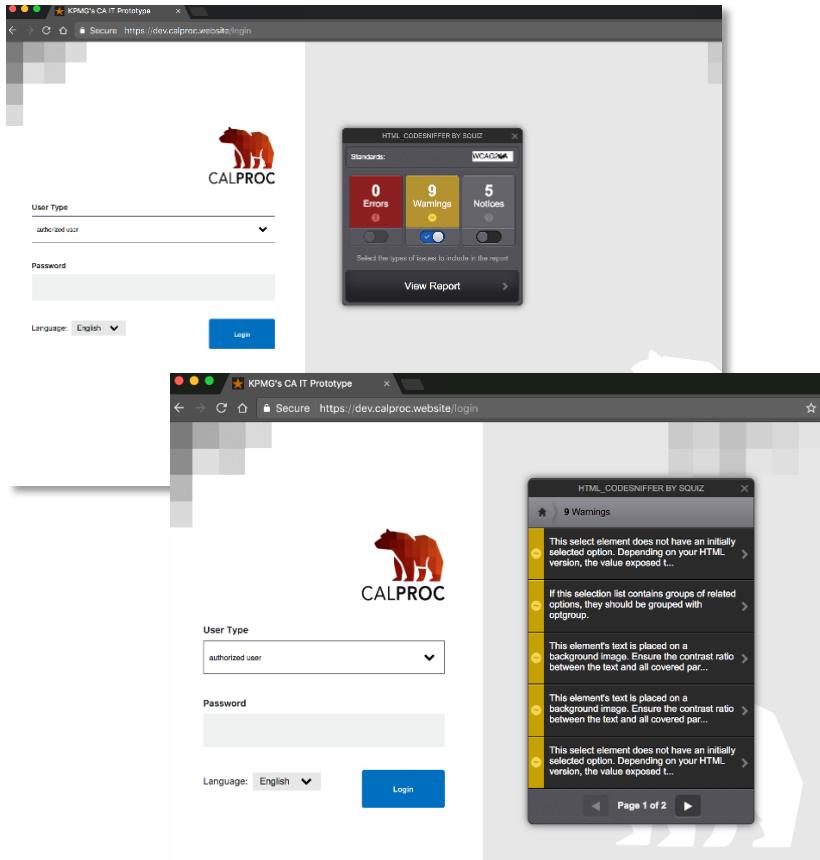
- Integrated with Jenkins CI process. Builds that don't pass all tests do not get deployed.
- ESLint for static analysis
- Karma + Jasmine for unit tests
- Protractor for automated UI tests
- Istanbul for instrumenting/reporting code coverage
- Custom "[pseudoloc](#)" webpack loader for localization testing
- Scaffolding system ensures new components start with tests and quality code by default
- [Latest test reports from Jenkins](#)

# UI → Server → DB flow

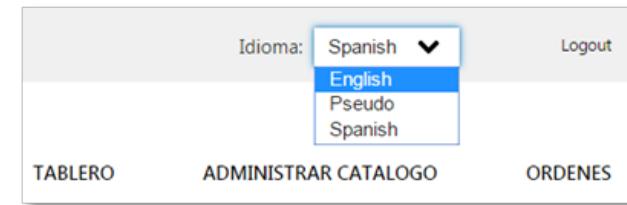


# Testing and Quality Assurance (1 of 2)

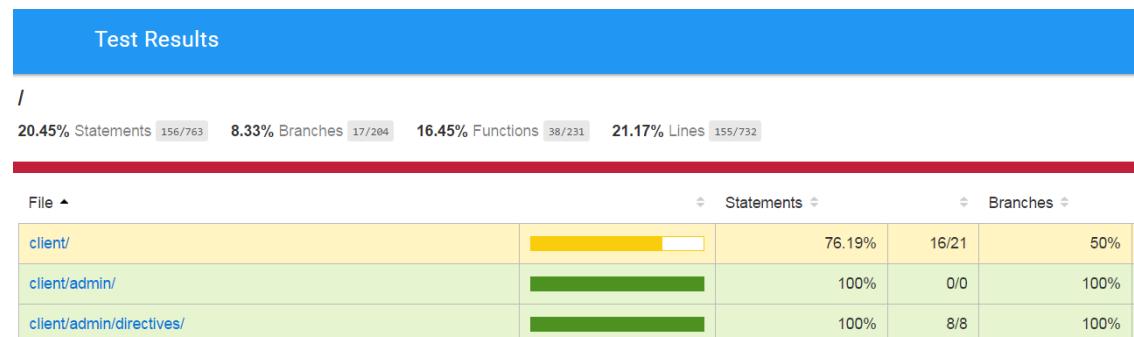
We performed HTML code and 508 compliance testing using HTML\_CodeSniffer and Chrome Accessibility Tools.



We added Pseudo Localization early, which allowed developers to identify localization issues before adding in additional languages.



We used Istanbul for code coverage and analysis. We would improve the score in future iterations.



# Testing and Quality Assurance (2 of 2)

We used ESLint for static code quality analysis.

Test Results      Karma    Protractor    Istanbul    ESLint

### ESLint Results

**Summary**

	Files with errors	0	
Errors	0	Files with only warnings	0
Warnings	0	Clean files	134
Total alerts	0	Total files linted	134

**File Breakdown** (expand all)

File	Errors	Warnings
/u01/app/jenkins/workspace/Development/1_Build_Application/client/admin/directives/catalog-item/CatalogItem.js	0	0
/u01/app/jenkins/workspace/Development/1_Build_Application/client/admin/directives/catalog-item/CatalogItem.spec.js	0	0
/u01/app/jenkins/workspace/Development/1_Build_Application/client/admin/directives/catalog-item/index.js	0	0
/u01/app/jenkins/workspace/Development/1_Build_Application/client/admin/directives/contract-details/ContractDetails.js	0	0
/u01/app/jenkins/workspace/Development/1_Build_Application/client/admin/directives/contract-details/ContractDetails.spec.js	0	0

We used Karma for automated unit testing:

Test Results      Karma    Protractor    Istanbul    ESLint

Tested in Chrome 56.0.2924 (Linux 0.0.0) on Fri Mar 03 2017

**39 specs, 0 failed, 1 pending**

**CatalogItem**  
should be creatable

**ContractDetails**  
should be creatable  
should initialize default name to heading  
should initialize custom name to heading

**Contracts**  
should be creatable  
should initialize default name to heading  
should initialize custom name to heading

**DashboardByContract**  
should be creatable  
should initialize default name to heading  
should initialize custom name to heading



We used Protractor for automated User Interface testing:

Test Results      Karma    Protractor    Istanbul    ESLint

**KPMG CA Prototype - 1.899s**  
Tests: 1 Skipped: 0 Failures: 0

page should have a title - 1.896s  
• Passed. ✓

**crawling the application - 1.853s**  
Tests: 1 Skipped: 0 Failures: 0

should crawl '/' as {"userId":"123"} @ [1280,720] and not find any errors - 1.853s  
• Passed. ✓

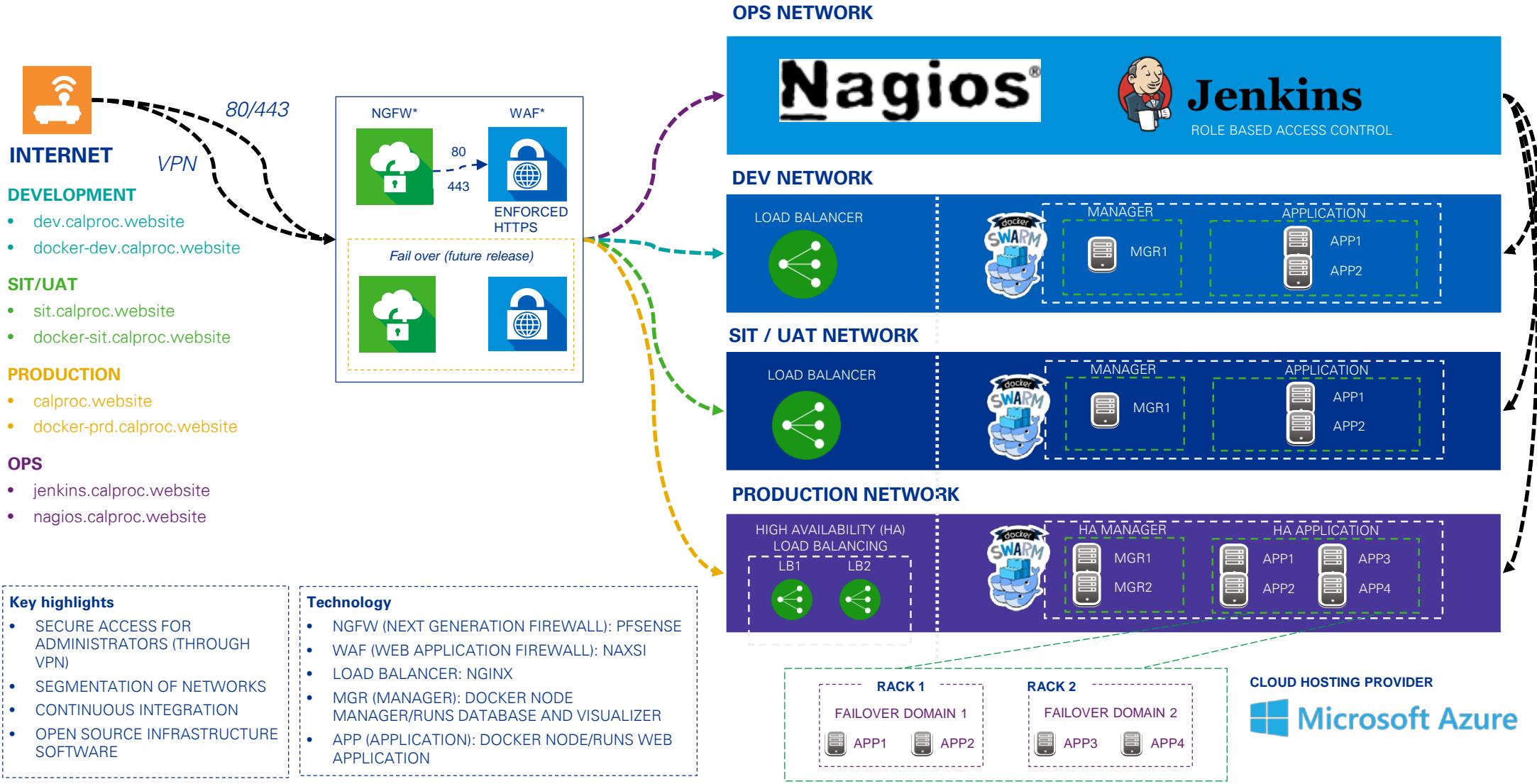




# DevOps



# Infrastructure diagram



# Virtual machines

## Dev, Jenkins, Nagios

19 items	
NAME	STATUS
agi-dev01vapp01	Running
agi-dev01vapp02	Running
agi-dev01vjenk01	Stopped (deallocated)
agi-dev01vjenk02	Running
agi-dev01vmgr01	Running
agi-dev01vnag01	Running
agi-dev01vweb01	Running

## WAF

1 items	
NAME	STATUS
ops01rvprox01	Running

## NGFW

1 items	
NAME	STATUS
ops01vngfw02	Running

## SIT/UAT and Production

19 items	
NAME	STATUS
agi-prd01vapp01	Running
agi-prd01vapp02	Running
agi-prd01vapp03	Running
agi-prd01vapp04	Running
agi-prd01vmgr01	Running
agi-prd01vmgr02	Running
agi-prd01vweb01	Starting
agi-prd01vweb02	Running
agi-sit01vapp01	Running
agi-sit01vapp02	Running
agi-sit01vmgr01	Running
agi-sit01vweb01	Running



# High availability sets

Essentials ^

Resource group (change) [REDACTED]  
Location [REDACTED]  
Subscription name (change) [REDACTED]  
Subscription ID [REDACTED]

Fault domains 2  
Update domains 2  
Virtual machines 2  
Managed No

Search virtual machines

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
agi-prd01web01	Running	0	0
agi-prd01web02	Running	1	1

Essentials ^

Resource group (change) [REDACTED]  
Location [REDACTED]  
Subscription name (change) [REDACTED]  
Subscription ID [REDACTED]

Fault domains 2  
Update domains 2  
Virtual machines 4  
Managed No

Search virtual machines

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
agi-prd01app01	Running	0	0
agi-prd01app02	Running	1	1
agi-prd01app03	Running	0	1
agi-prd01app04	Running	1	0

Essentials ^

Resource group (change) [REDACTED]  
Location [REDACTED]  
Subscription name (change) [REDACTED]  
Subscription ID [REDACTED]

Fault domains 2  
Update domains 2  
Virtual machines 2  
Managed No

Search virtual machines

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
agi-prd01vmgr01	Running	0	0
agi-prd01vmgr02	Running	1	1

Note: some details redacted for security purposes.

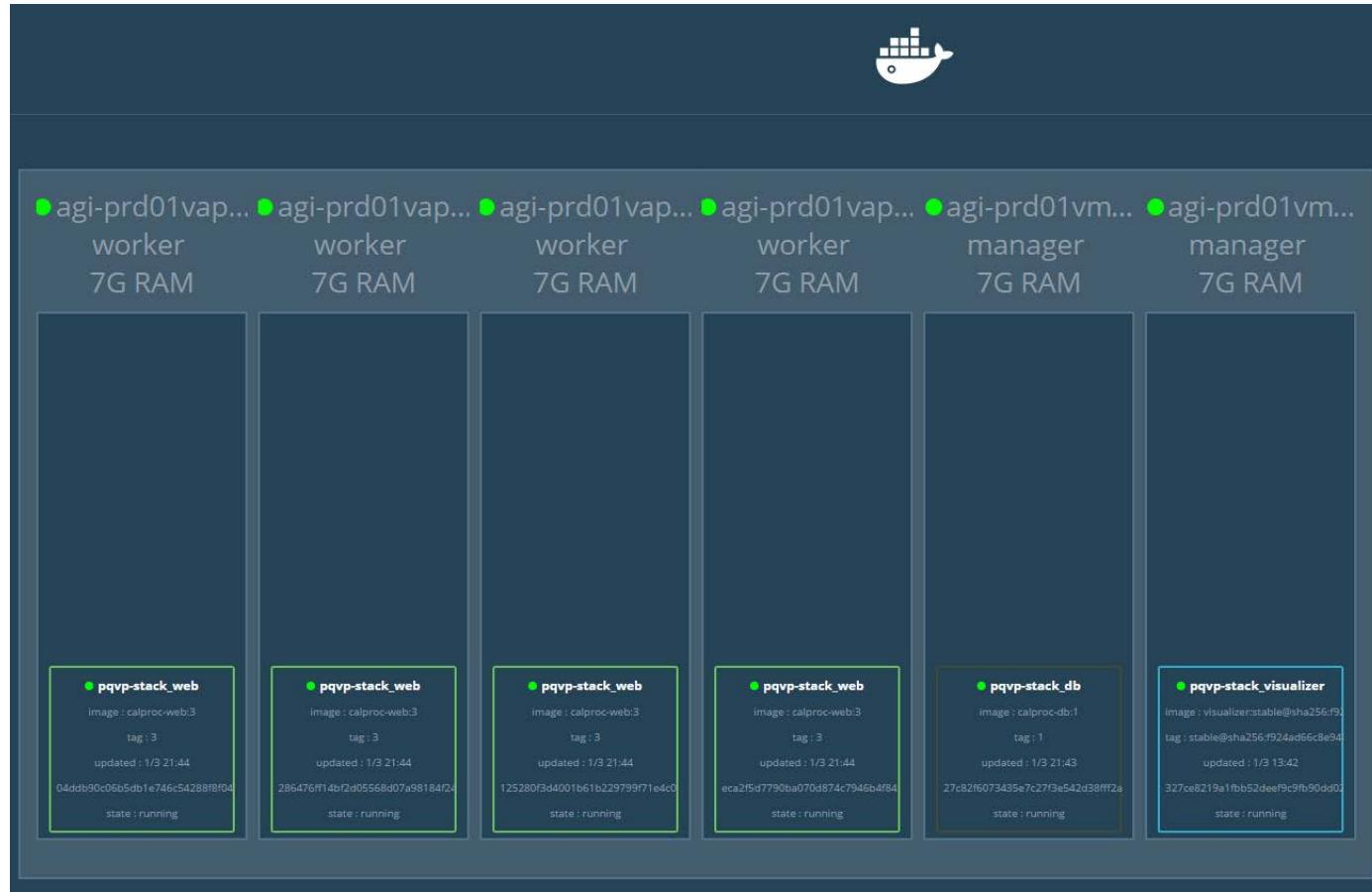
# Network segmentation

The screenshot shows the Microsoft Azure portal interface for managing a virtual network. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, SETTINGS (Address space, Connected devices, Subnets, DNS servers, Peerings, Properties, Locks, Automation script), and a search bar. The 'Subnets' link is currently selected. The main content area displays a table of subnets under the resource group 'AGI'. The columns are NAME, ADDRESS RANGE, AVAILABLE ADDRESSES, and SECURITY GROUP. The subnets listed are:

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
AGI.AOP01-DMZ	172.20.106.0/24	251	NSG_MNG1-DMZ
AGI.AOP01-MGMT	172.20.107.0/24	251	NSG_MNG1-MGMT
AGI.AOP01-APP	172.20.108.0/24	248	NSG_MNG1-APP
AGI.ADV01-DMZ	172.20.109.0/24	251	NSG_MNG1-DMZ
AGI.ADV01-MGMT	172.20.110.0/24	250	NSG_MNG1-MGMT
AGI.ADV01-APP	172.20.111.0/24	248	NSG_MNG1-APP
AGI.ASI01-DMZ	172.20.112.0/24	251	NSG_MNG1-DMZ
AGI.ASI01-MGMT	172.20.113.0/24	250	NSG_MNG1-MGMT
AGI.ASI01-APP	172.20.114.0/24	248	NSG_MNG1-APP
AGI.APP01-DMZ	172.20.115.0/24	251	NSG_MNG1-DMZ
AGI.APP01-MGMT	172.20.116.0/24	249	NSG_MNG1-MGMT
AGI.APP01-APP	172.20.117.0/24	245	NSG_MNG1-APP



# Production Docker Swarm



```
@agi-prd01vmgr01:~$ docker node ls
ID          HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS
2qlvqdzw1gaurcqb64prd72p  agi-prd01vapp03  Ready  Active
3rtbkpxui7hjqyy2hn75oub0 *  agi-prd01vmgr01  Ready  Active  Leader
kz0jc919vy0eyj1oaiswfi4g4  agi-prd01vmgr02  Ready  Active
sp9jnx92eoiiini6ehic0rdlk0  agi-prd01vapp02  Ready  Active
sugfoar9ocu3n5okke2okeqfm  agi-prd01vapp01  Ready  Active
x9xpjuehrjc8m9v2v0irb7s6  agi-prd01vapp04  Ready  Active
```

See more at:

- Dev: <https://docker-dev.calproc.website/>
- SIT: <https://docker-sit.calproc.website/>
- PROD: <https://docker-prd.calproc.website/>



# Continuous Resource Monitoring with Nagios

**Nagios®**

**Current Network Status**  
Last Updated: Wed Mar 1 01:30:09 UTC 2017  
Updated every 90 seconds  
Nagios Core™ 4.3.1 - www.nagios.org  
Logged in as nagiosadmin

**General**  
Home Documentation

**Current Status**  
Tactical Overview  
Map (Legacy)  
Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**  
Availability Trends (Legacy) Alerts History Summary Histogram (Legacy) Notifications Event Log

**System**  
Comments Downtime Process Info Performance Info Scheduling Queue Configuration

**Service Status Details For All Hosts**  
Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
dev01vapp01	Current Load	OK	03-01-2017 01:28:37	2d 9h 6m 36s	1/4	OK - load average: 0.00, 0.02, 0.00
	Current Users	OK	03-01-2017 01:29:00	2d 9h 6m 15s	1/4	USERS OK - 0 users currently logged in
	PING	OK	03-01-2017 01:25:19	2d 9h 29m 53s	1/4	PING OK - Packet loss = 0%, RTA = 1.37 ms
	Root Partition	OK	03-01-2017 01:29:00	2d 9h 6m 13s	1/4	DISK OK - free space: /:22642 MB (78% used=90%)
	SSH	OK	03-01-2017 01:26:43	2d 9h 28m 33s	1/4	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.1 (protocol 2.0)
	TCP	OK	03-01-2017 01:27:33	1d 9h 27m 43s	1/4	TCP OK - 0.003 second response time on 172.20.111.2
	Total Processes	OK	03-01-2017 01:27:33	0d 1h 32m 36s	1/4	PROCS OK: 150 processes
	webstart docker memory	OK	03-01-2017 01:28:37	0d 18h 14m 6s	1/4	OK: pqvp-stack_web.zu08dh415bgf52k02g4fbq.u9c9
	webstart docker status	OK	03-01-2017 01:28:57	0d 5h 0m 48s	1/4	OK: pqvp-stack_web.zu08dh415bgf52k02g4fbq.u9c9
	webstart docker uptime	OK	03-01-2017 01:29:14	0d 18h 24m 59s	1/4	OK: pqvp-stack_web.zu08dh415bgf52k02g4fbq.u9c9
localhost	Current Load	OK	03-01-2017 01:26:24	2d 10h 15m 52s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	03-01-2017 01:25:44	2d 10h 15m 14s	1/4	USERS OK - 0 users currently logged in
	HTTP	OK	03-01-2017 01:28:45	2d 10h 14m 37s	1/4	HTTP OK: HTTP/1.1 200 OK - 11595 bytes in 0.001 second response time
	PING	OK	03-01-2017 01:28:40	2d 10h 13m 59s	1/4	PING OK - Packet loss = 0%, RTA = 0.06 ms
	Root Partition	OK	03-01-2017 01:27:38	2d 10h 13m 22s	1/4	DISK OK - free space: /:27641 MB (93% used=97%)
	SSH	OK	03-01-2017 01:25:41	2d 10h 12m 44s	1/4	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.1 (protocol 2.0)
	Total Processes	OK	03-01-2017 01:28:45	2d 10h 11m 29s	1/4	PROCS OK: 42 processes with STATE = RSZD

**Service Status Details For All Hosts**

**March 02, 2017 13:00**

**Monitoring Overview**  
Last Updated: Wed Mar 1 01:23:19 UTC 2017  
Updated every 90 seconds  
Nagios Core™ 4.3.1 - www.nagios.org  
Logged in as nagiosadmin

**Monitoring Performance**  
Service Check Execution Time: 0.00 / 4.01 / 0.693 sec  
Service Check Latency: 0.00 / 0.00 / 0.00 sec  
Host Check Execution Time: 4.02 / 4.50 / 4.287 sec  
Host Check Latency: 0.00 / 0.00 / 0.00 sec  
# Active Host / Service Checks: 2 / 17  
# Passive Host / Service Checks: 0 / 0

**Network Outages**  
0 Outages

**Hosts**  
0 Down 0 Unreachable 2 Up 0 Pending

**Services**  
0 Critical 0 Warning 0 Unknown 17 Ok 0 Pending

**Monitoring Features**

Flop Detection	Notifications	Event Handlers	Active Checks	Passive Checks
✓ All Services Enabled	✓ 4 Services Disabled	✓ All Services Enabled	✓ All Services Enabled	✓ All Services Enabled
No Services Flapping	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled
All Hosts Enabled	No Hosts Flapping			

## Email alerts:

**All Unread**

Search Nagios (Ctrl+E)

RE: PROBLEM Service Alert: dev01vmgr02/Total Processes is WARNING \*\*  
Fri 3/3/2017 8:04 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 8:02 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 8:02 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 8:01 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is UNKNOWN \*\*  
Fri 3/3/2017 7:37 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 7:04 AM 11 KB

RE: PROBLEM Service Alert: dev01vmpg01/Total Processes is WARNING \*\*  
Fri 3/3/2017 7:01 AM 11 KB

RE: RECOVERY Service Alert: www.calproc.website/HTTPS\_443 is OK \*\*  
Fri 3/3/2017 6:57 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 6:57 AM 11 KB

RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Fri 3/3/2017 6:57 AM 11 KB

RE: PROBLEM Service Alert: www.calproc.website/HTTPS\_443 is CRITICAL \*\*  
Fri 3/3/2017 6:56 AM 11 KB

**Inbox**

To: Rogers, Bergener J  
Subject: RE: PROBLEM Service Alert: sit01vmpg01/calproc-db docker memory is WARNING \*\*  
Date/Time: Fri Mar 3 10:01:51 PST 2017  
Additional info:  
WARNING: pqvp-stack\_db\_1.n9117gw|tf7wwgg3hyqcba4 memory is 713.93359375m; OK: pqvp-stack\_visualizer.1.0ewmhft7t9gaswk4cdpq5o memory is 66.16460623m

See more at: <https://nagios.calproc.website>

Login:

ca-pqvp-guest

Password:

kpmgagile#2017



# Network Traffic Monitoring with GoAccess

GoAccess monitors our network traffic and bandwidth usage. The data is refreshed every 15 minutes.

**Dashboard**

Overall Analyzed Requests (02/Mar/2017 - 02/Mar/2017)

Last Updated: 2017-03-02 19:07:16 +0000

Total Requests 2,006	Valid Requests 2,006	Failed Requests 0	Processed Time 0	Unique Visitors 7	Unique Files 99
Excl. IP Hits 0	Referrers 0	Unique 404 4	Static Files 63	Log Size 455 KiB	Bandwidth 15.8 MiB

Unique visitors per day - Including spiders

Hits  
Visitors

02/Mar/2017

#	Hits	Visitors	Bandwidth	Data
1	2,006 (100.00%)	7 (100.00%)	15.8 MiB (100.00%)	02/Mar/2017

Requested Files (URLs)

Hits  
Visitors

#	Hits	Visitors	Bandwidth	Method	Protocol	Data
1	501 (24.98%)	1 (14.29%)	4.02 kB (0.02%)	GET	HTTP/1.1	/apis/nodes
2	501 (24.98%)	1 (14.29%)	8.11 MB (51.30%)	GET	HTTP/1.1	/apis/tasks
3	497 (24.78%)	1 (14.29%)	11.4 kB (0.07%)	GET	HTTP/1.1	/apis/services
4	17 (0.85%)	3 (42.86%)	1.38 kB (0.01%)	GET	HTTP/1.1	/api/v1/images/138
5	15 (0.75%)	3 (42.86%)	1.29 kB (0.01%)	GET	HTTP/1.1	/api/v1/images/125
6	14 (0.70%)	3 (42.86%)	1.01e+3 B (0.01%)	GET	HTTP/1.1	/api/v1/images/95
7	13 (0.65%)	6 (85.71%)	660 kB (4.08%)	GET	HTTP/1.1	/

**See this working live:**

- Dev: <https://dev-lb.calproc.website>
- SIT/UAT: <https://sit-lb.calproc.website>
- PROD1\*: <https://prd-lb1.calproc.website>
- PROD2\*: <https://prd-lb2.calproc.website>

\*traffic is monitored on each specific load balancer. We have implemented "round robin" load balancing on nginx.



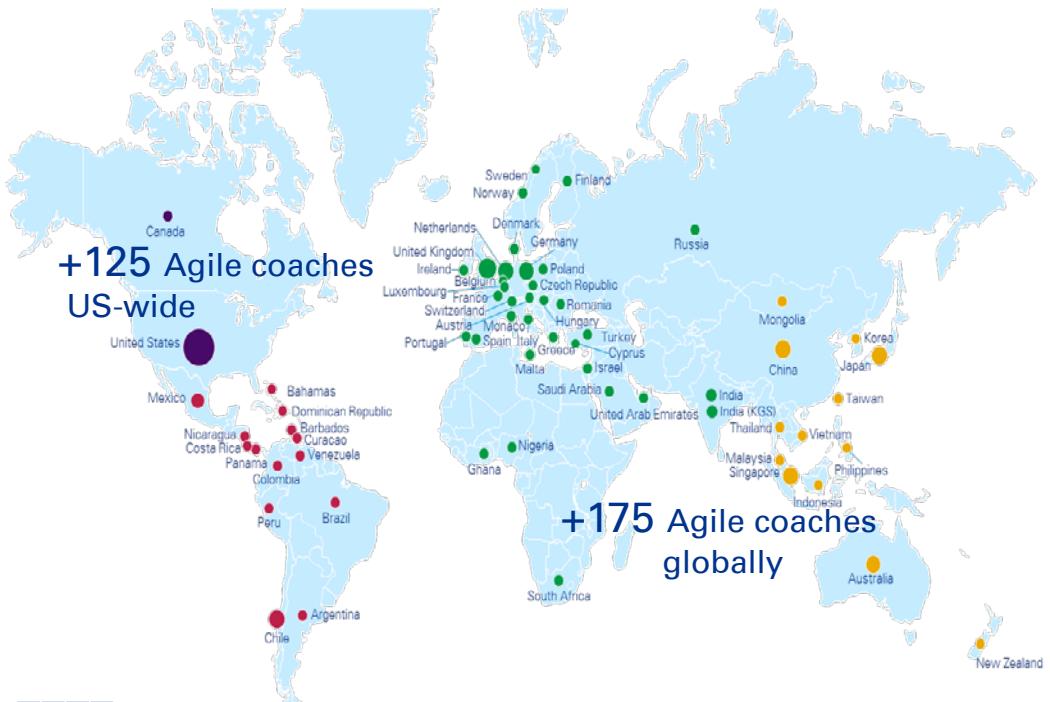
# Our commitment to Agile



# KPMG Agile capabilities U.S. and global coverage market recognition

Our U.S. and global bench of Agile coaches and trainers know core Agile approaches and scaled frameworks, and have practical delivery experience with implementation projects. Resources can be deployed as a Virtual Bench or as a Named Resource model.

## AGILE U.S. and Global Coverage



## Awards and Recognition



Named a “**major player**” by IDC for Worldwide Organizational and Change Consulting Services



**Training Magazine Hall of Fame -Top 10**  
globally for more than four consecutive years



**#6 on Learning Elite**  
KPMG placed higher than any other professional services firm



Recognized as **ALM Intelligence Vanguard Leader** (highest category rating) for behavioral change management



# Conclusion and next steps

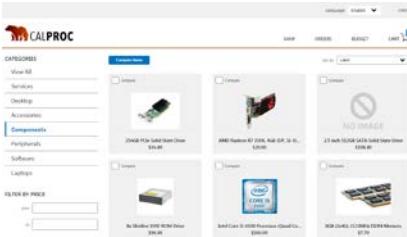


# Release retrospective | Key accomplishments (1 of 3)

The team reflected on the most important features and tasks we achieved in the three sprints. The top three were:

1

Delivered the main features that users wanted.



**Pictures of items** – users really wanted to see what they were buying. The team found generic, royalty-free images to add to the site. Future iterations would add more images, and perhaps purchased images direct from the manufacturer or a professional vendor.

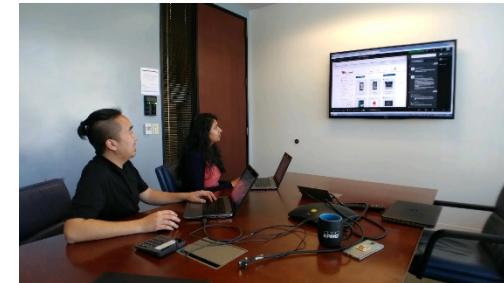
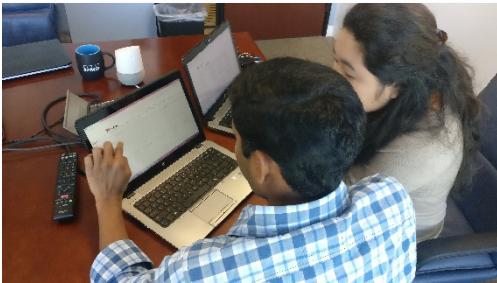
**A working budget page** – users wanted to keep track of their spending and see an overview with easy to understand visuals. Future iterations would add budget validation checks to ordering (e.g. to prevent exceeding a set budget limit)



**A working summary dashboard** – admin users wanted to be able to see total expenditure for the year at a glance. They also wanted to see how this compared to the same time in the previous year. Additionally, they wanted to see a breakdown by item category. Future iterations would allow users to drill into these dashboards and view the source data.

# Release retrospective | Key accomplishments (2 of 3)

## 2 Involved users early and often.



**From the initial user-story and backlog creation, initial wireframe and workflow reviews** through to the final sprint review, we placed the users at the heart of our development.

This helped us plan the release and prioritize the most relevant items for each sprint. The regular feedback and interactions with the users either confirmed we were still on the right track, or forced us to adapt and change.



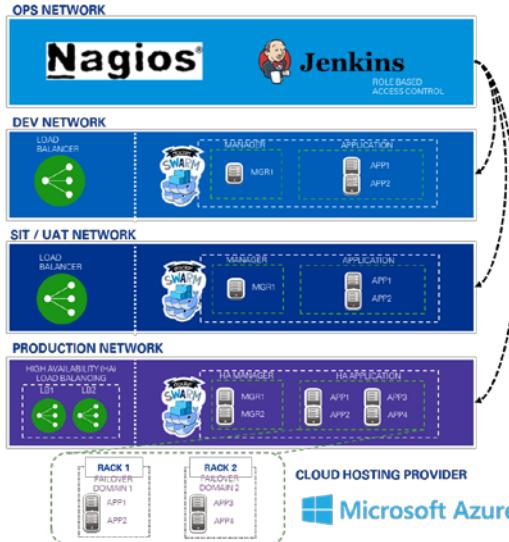
A screenshot of a GitHub issue page. The title of the issue is "Modify logo to remove "procurement system" #162". The status is "Closed" and it was opened by "benrogers-kpmg" 6 days ago with 2 comments. A comment from "benrogers-kpmg" is highlighted, stating: "During usability testing, users commented that they would like the "procurement system" text removed from the logo. Reasoning - they already know it's a procurement system." The GitHub interface includes navigation bars for Code, Issues (64), Pull requests (2), Projects (7), Wiki, Pulse, Graphs, and Settings.



# Release retrospective | Key accomplishments (3 of 3)

## 3

## Minimizing Technical Debt and maximizing quality.



**Robust Infrastructure** with secure VPN access for admins, segmentation of networks, load balancing and CI with rapid build times.



**Docker Containers Deployed to Azure**

Web App	Neo4J Database	Docker Visualizer	GoAccess	Nagios
<ul style="list-style-type: none"><li>Multiple load balanced nodes</li><li>Swagger / SwaggerHub / RAML API specs</li><li>UI layer<ul style="list-style-type: none"><li>Angular 2</li><li>USWDS UI components</li></ul></li><li>Webpack configured for ES6 and hot reload</li><li>tBNext for localization</li><li>Custom RAML webpack loader for auto-generated wrappers around API specs</li></ul>	<ul style="list-style-type: none"><li>Behind firewall, only accessible via the service layer</li><li>Sample data converted to CSV format, supplemented with additional data, denormalized, and then imported into this open source graph DB</li></ul>	<ul style="list-style-type: none"><li>Monitoring tools used to track and alert on the health and usage of the application</li><li>Production Docker Visualizer</li><li>Production Nagios (login: ca-pqv-guest / kpmgagile#2017)</li><li>Production GoAccess</li></ul>		

**Automated Testing**

- Integrated with Jenkins CI process. Builds that don't pass all tests do not get deployed.
- ESLint for static analysis
- Karma + Jasmine for unit tests
- Protractor for automated UI tests
- Istanbul for instrumenting/reporting
- Custom "pseudoloc" webpack loader
- Scaffolding system ensures new
- Latest test reports from Jenkins

Idioma: Spanish English Pseudo Spanish

TABLERO ADMINISTRAR CATALOGO ORDENES



**Scalable technical architecture laying the foundations for the future.** We believe that being agile doesn't mean taking shortcuts. We invested early in code quality and doing things "the right way". This accelerated our velocity in Sprints 2 and 3. This quality will continue to pay off in future iterations – meaning it will take far less time to add new features, such as support for more languages (Spanish and example Pseudo already implemented).

**Strong brand and style guide**  
– users liked that our simple design had a strong brand. We based our style on the U.S. Web Design Standard, 18F Content Guide, and Usability.gov.

# Remaining Backlog (1 of 3)

The screenshot shows a GitHub backlog board titled "Future Sprints - Outstanding Backlog". The board has two columns: "Backlog" and "Priority".

- Backlog:** Contains 26 items.
  - ① as an authorized user, I want to create an automatically repeating order up to the end date I specify so that I do not have to manually make repetitive orders  
#280 opened by mattkwong-kpmg 5 points story
  - ① As an Admin User I want to add labels, meta data and other custom fields to an item, which can appear in search results so that I can help Authorized Users make better decisions. I can extend the solution without developer hours.  
#56 opened by mattkwong-kpmg 3 points story
  - ① As an Authorized User I want to compare software items by release date so that I can make an informed decision about the product I want to order.  
#32 opened by mattkwong-kpmg 3 points story
  - ① As an Authorized User I want to be able to delete one or more multiple items in my shopping basket at checkout and/or at any time so that I do not have to delete an item manually multiple times  
#268 opened by mattkwong-kpmg 3 points story
- Priority:** Contains 9 items.
  - ① As an authorized user I want to be able to adjust quantities and remove items from my cart  
#149 opened by robertlevy 2 points story
  - ① as an authorized user, I want to be able to perform a text search on the catalog so that I can easily find the information I am looking for  
#277 opened by mattkwong-kpmg 3 points story
  - ① "Latest" and "Oldest" sort value details  
#243 opened by mattkwong-kpmg 3 points story
  - ① DB password - implement in ENV variable  
#250 opened by npearce-kpmg 3 points story
  - ① convert header/footer markup to lists  
#269 opened by robertlevy enhancement
  - ① As an Authorized User I want to be able to edit the quantity of an item in my shopping basket at checkout and/or at any time so that I do not have to add or

We iterated through as much of the backlog as we could through three one-week sprints. During these three sprints, the Product Manager prioritized the user stories and functionality related to the core prototype requirements and most pressing user feedback.

Bug fixes, user stories, improvements and other work items that were not completed in the first three sprints remain in the backlog.

In the sprint 3 retrospective meeting, the team decided to organize this backlog in a new GitHub project “Future Sprints – Outstanding Backlog” for transparency. We grouped several into a “Priority” column. These represent the issues we would likely work on in a Sprint 4.

The screenshot shows the GitHub project page for "Future Sprints - Outstanding Backlog".

- Project Summary:** 7 projects, updated just now. Description: This project represents the user stories, defects and work items that we would plan to handle in future sprints.
- Project Metrics:** Unwatch 3, Star 3, Fork 1.
- Project Options:** New Project button.

# Remaining Backlog (2 of 3)

Given the timeframe, the Product Manager had to make several scope compromises – in consultations with users and the rest of the team. Some of these examples and reasoning are provided below:

**Search ([Issue #277](#))** – as the initial data set was relatively small, users felt it was more important to improve navigation, layout and presentation of the “shop” rather than implement a text search feature. However, as the data set grows the ability to search becomes more important and would be top of our list for a future sprint.

**Editing Quantities ([Issues #41, #149](#))** – many users were nervous about being allowed to specify large quantities of items in their carts/orders without any validation against what they are allowed to spend (e.g. per their approved budget/allowance). This validation is tracked in the backlog as issue [#313](#), which is a more complex effort. As users can still achieve their key objectives without having a quantity edit (i.e. they are able to submit multiple orders), this feature was moved to the backlog and other core functionality was prioritized instead.

**Review Functionality ([Issue #271, #282](#))** – users emphasized how important reviews from other buyers were in influencing their decisions. They stressed that they wanted to see reviews from other state employees, and not just general public reviews. In a future sprint, we would build this functionality and include the ability for users to give star ratings, leave text reviews and also be rewarded with points for giving the review – similar to Google “Local Guides” and Yelp.

# Remaining Backlog (3 of 3)

**Amazon (and other commercial site) Integration ([Issue #275](#))** – users mentioned that, in addition to specific reviews of a product/service, it would be helpful to view public reviews and more detailed information by displaying corresponding information from Amazon or similar commercial site. Admin users also mentioned that retrieving some information from a commercial site would be helpful and would reduce the amount of maintenance and manual work they would have to perform on the site. During the initial sprint, our business analysts researched and found several Amazon product IDs for use in an API call, but the Product Manager made the decision to move this to the backlog and prioritize more essential functionality for sprints 1 to 3.



# See also



# Capabilities and Thought Leadership

KPMG provides services in Human Centered Design, Mobile and Web Application Development and Agile transformations. The following links highlight our services, work and Thought Leadership.



The screenshot shows the KPMG Digital and Mobile Solutions page. It features a header with navigation links like Services, Industries, Topics, Insights, Institutes, and Contact. Below the header is a large image of a person working at a desk with a laptop, smartphone, and tablet. A sidebar on the left contains sections for 'Digital and Mobile Solutions' and 'Client Stories'. The main content area has a heading 'Thriving in a digital world' with a sub-section about 'Digital disruption' and a call-to-action button 'Submit RFP'.

## Digital and Mobile Solutions

<https://advisory.kpmg.us/managementconsulting/capabilities/digital-mobile.html>



The screenshot shows the 'Thriving In A Digital World' whitepaper by Rick Wright. It includes a bio, a photo of Rick Wright, and a section titled 'KPMG Can Help'. Below this is an 'Infographic' titled 'Human insights meet digital thinking' with five key points: focus on the problem, deliver small, fast and often, design for people not processes, always look for improvement, and focus on the problem first. At the bottom is a call-to-action 'Read this whitepaper to learn more about the four kinds successful organizations are using to create value and sheet the challenges of this new landscape by redefining the need to think and act digitally. By focusing on the problems, designing for the opportunities, learning in an agile way and continually looking to improve, today's organizations thrive in a digital world—and you can too.'

## Thriving In A Digital World

<https://institutes.kpmg.us/institutes/advisory-institute/articles/2016/07/thriving-digital-world.html>



The screenshot shows the IDC MarketScape: Worldwide Digital Enterprise Strategy Consulting Services 2015 Vendor Assessment report. It features a world map and a grid ranking vendors. The grid has 'Competitiveness' on the vertical axis and 'Strategic Vision' on the horizontal axis. Vendors listed include PwC, Deloitte, KPMG, EY, and others. A call-to-action 'Contact KPMG' is present.

## IDC Report

<https://advisory.kpmg.us/content/dam/kpmg-advisory/PDFs/ManagementConsulting/idc-mark.pdf>



The screenshot shows the Forrester Wave™: Mobile Enterprise App Services Providers — Midsize Providers, Q1 2015 report. It includes a 'KEY TAKEAWAYS' section, a 'Market' analysis, and a 'Mobile Services Providers Bring Speed And Focus To Mobile Enterprise Apps' section. A 'FIGURE 1' chart ranks providers across 'Competitiveness' and 'Strategic Vision'. A call-to-action 'Access The Forrester Wave Model For Overall Insight' is included.

## Forrester Report

<https://advisory.kpmg.us/content/dam/kpmg-advisory/PDFs/ManagementConsulting/forrester-wave.pdf>



Thank you





**Primary contact:**

**Nancy Pinkins**

Manager, Advisory, Service Operations  
500 Capitol Mall, Suite 2100  
Sacramento, CA 95814  
916-554-1737  
[npinkins@kpmg.com](mailto:npinkins@kpmg.com)

**Secondary contact:**

**Tom Keane**

Account Relationship Director  
500 Capitol Mall, Suite 2100  
Sacramento, CA 95814  
916-551-3157  
[tkeane@kpmg.com](mailto:tkeane@kpmg.com)



[kpmg.com/socialmedia](http://kpmg.com/socialmedia)

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

© 2017 KPMG LLP, a Delaware limited liability partnership and the U.S. member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The KPMG name and logo are registered trademarks or trademarks of KPMG International.