

# Bill DeRouchey

Designer, information architect, design leader

September 2022

# 25+ Years in Design and UX

Focusing on systems, patterns, connections, structures, and language.

Product Designer  
Information Architect  
Design Director  
Creative Director  
Interaction Designer  
Writer

2022 (contract)	 COMPASS
2018–2021	 zendesk
2012–2018	 GE Digital
2010–2012	 SIMPLE
2004–2010	 ziba
2007–2010	 IXDA <small>INTERACTION DESIGN ASSOCIATION</small>



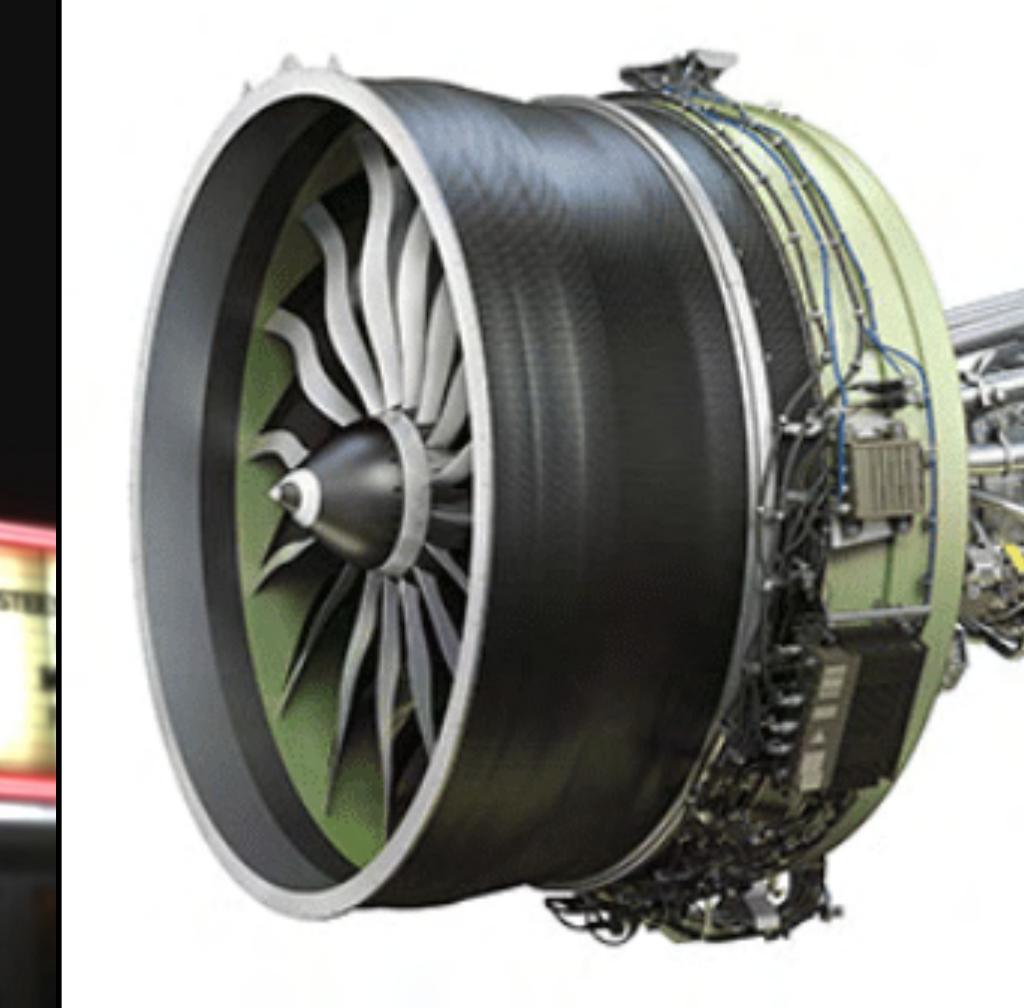
Led experience at a banking startup



Designed interfaces for physical products



Co-chaired a design conference



Directed design for jet engine maintenance



Presented on design topics at 30+ events

# Three Stories



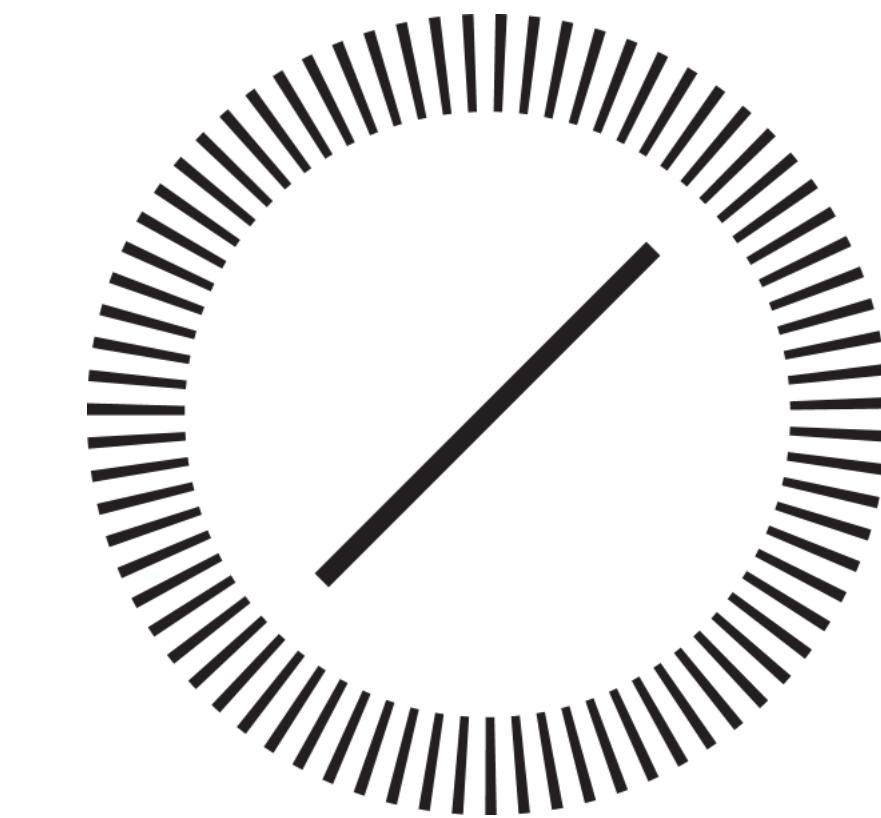
## GE Digital

Design Director  
2012–2018



## Zendesk

Principal Designer  
2018–2021



## Compass Realty

Experience Architect  
2022 (contract)

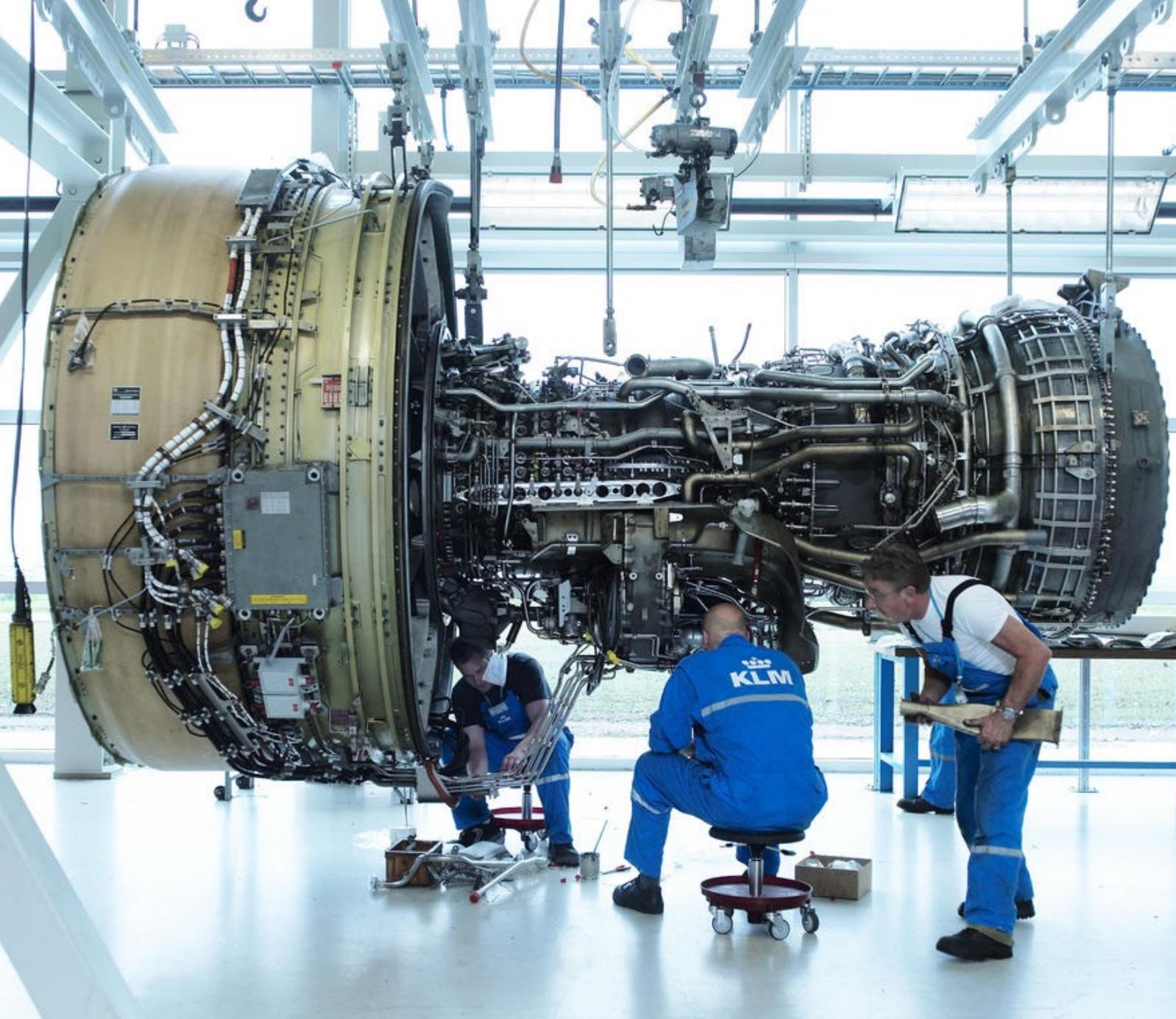


**GE Digital**  
Jet Engine Management

# About

Historic corporate giant with the mission to build the Industrial Internet.

Aimed to use sensor data to better predict engine maintenance in wind farms, power plants, locomotives, jet engines, and other things that spin.



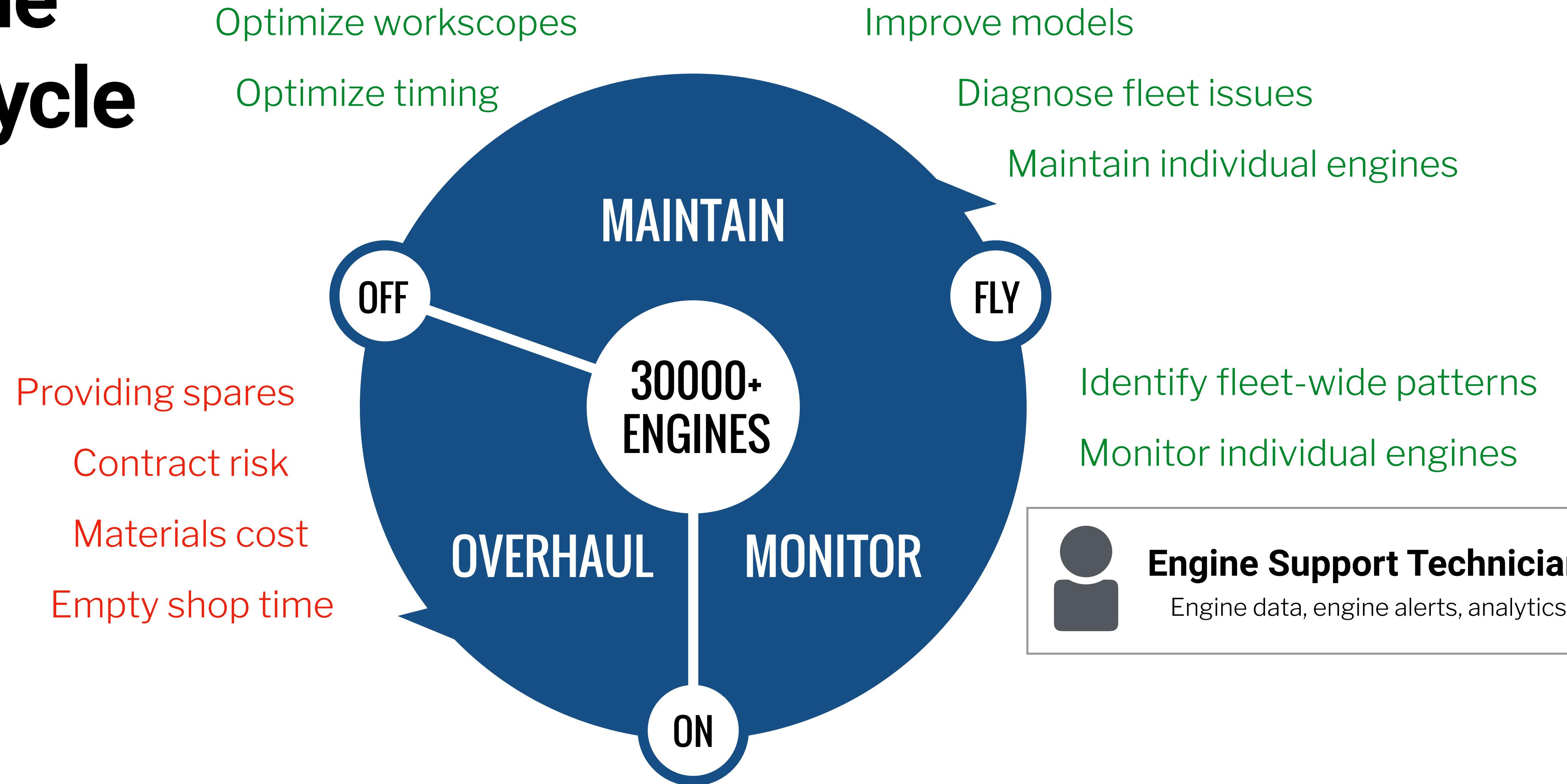
# My Role

Help **GE Aviation engineers** use data to  
diagnose engine issues faster to better  
help **Airlines** maintain aircraft availability

# Directed 20+ projects/products for GE Aviation



# Engine Lifecycle



# Led Research Programs



GE Aviation Fleet Support outside Cincinnati, Ohio



Engine Support, Fleet Managers, Propulsion Engineers

# Key Insight

## Talk was all about alerts

“The alert queue.” “This is how we’ve been doing it for 20 years.” The ancient system required them to think this way.

## Behavior was about engines

They immediately accessed contextual data about the engine in order to diagnose whether a problem existed.

# Key Insight



**Engine  
Alerts**

**Aircraft  
Availability**

**Airlines**

Focusing on alerts created a disconnect with customer airlines.  
We needed to shift how they thought and acted.

# Key Insight



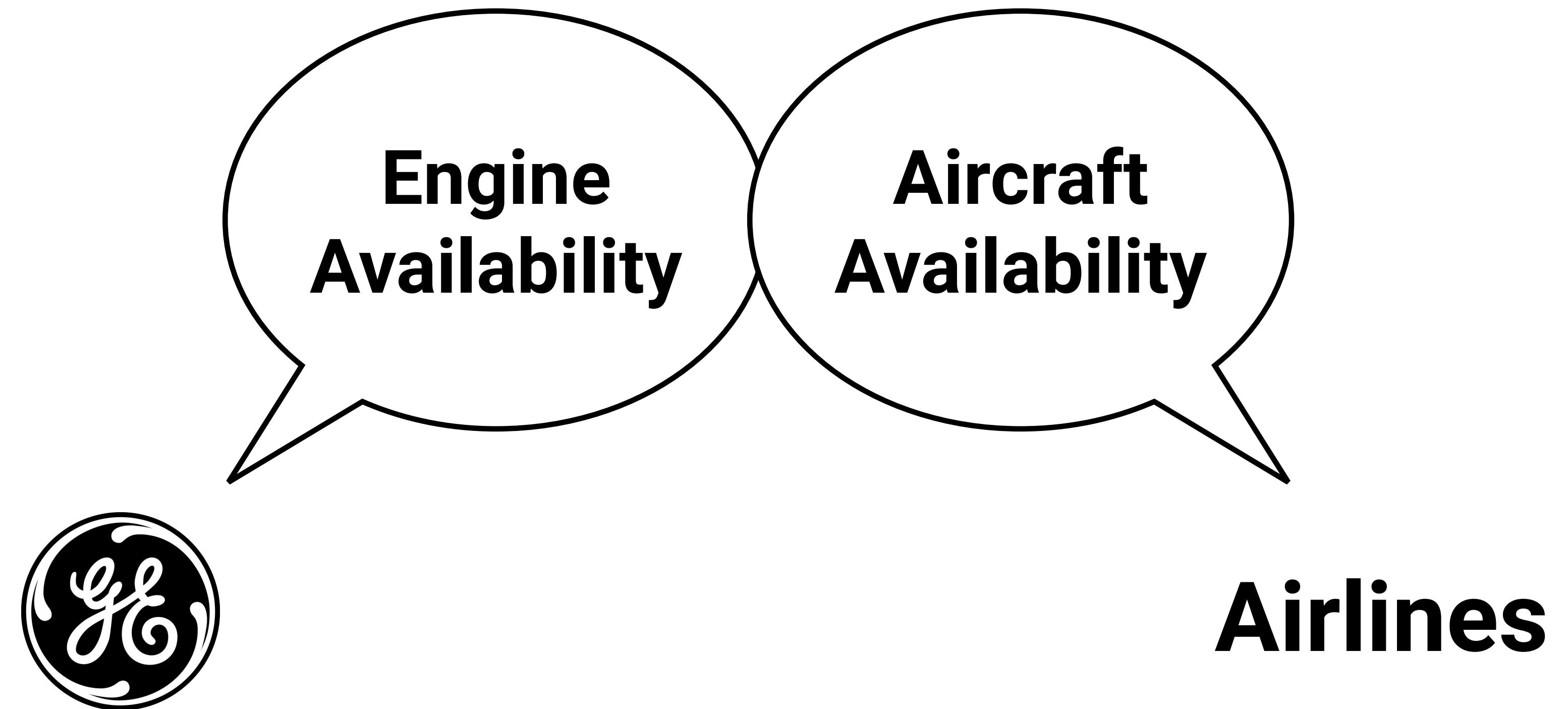
**Engine  
Health**

**Aircraft  
Availability**

**Airlines**

Focusing on alerts created a disconnect with customer airlines.  
We needed to shift how they thought and acted.

# Key Insight



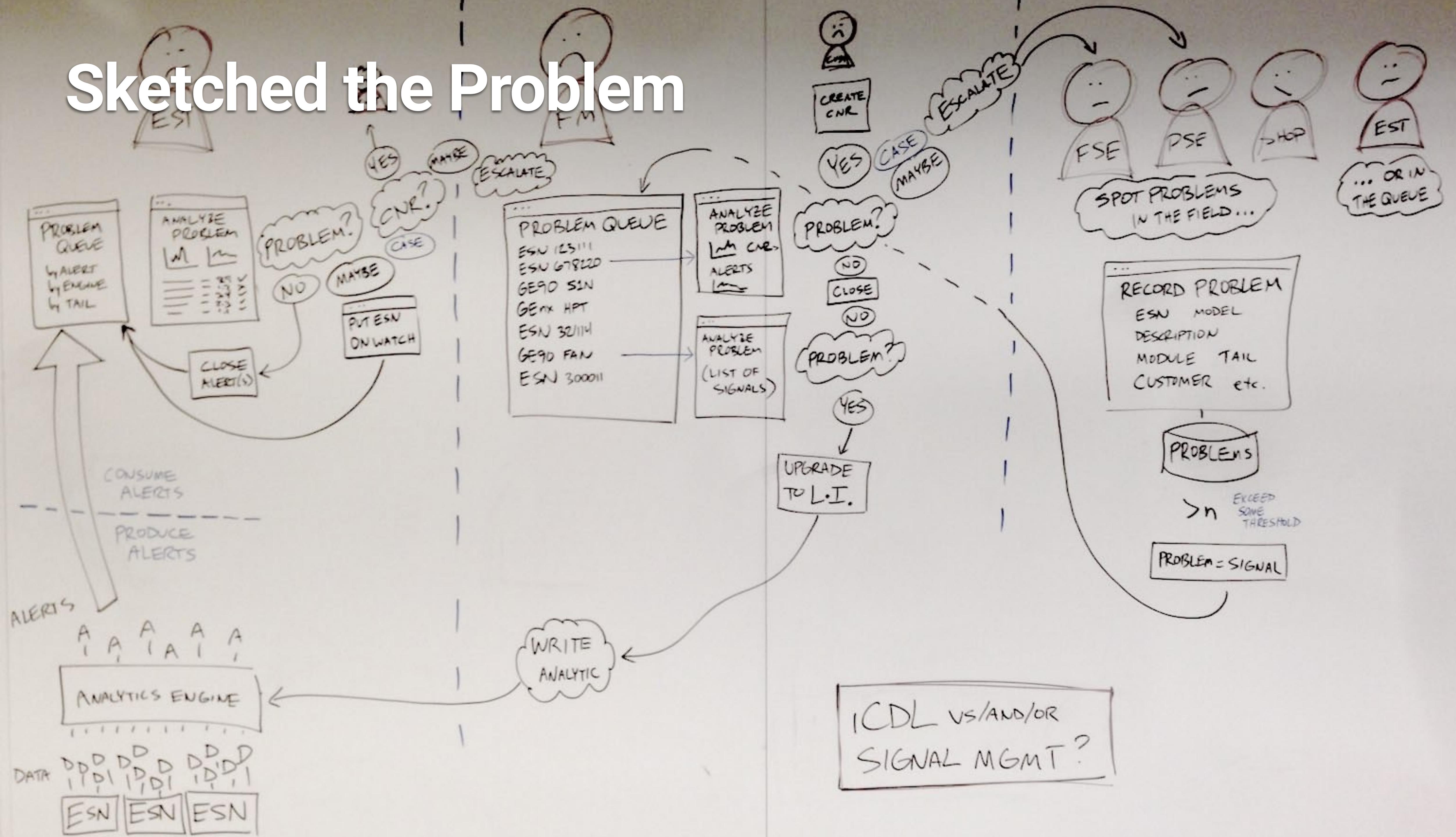
Focusing on alerts created a disconnect with customer airlines.  
We needed to shift how they thought and acted.

# Key Insight



Focusing on alerts created a disconnect with customer airlines.  
We needed to shift how they thought and acted.

# Sketched the Problem



Alerts Cases Engines



Family



Model



Aircraft



Operator



1

2

4

5

3

LanChile Open

Alert Status Open Severity All Type All Alert Date Last 7 days Flight Date Last 7 days

Classification 2 selected

Engines Alerts Faults &amp; Exceeds Sort By Severity Export

**706963** LanChile  
CF6-80C2F  
Position 2  
CC-CWY  
B767-300ER 6 Alerts  
NEWEST ALERT  
an hour ago  
25 Apr 2018 - 14:04 UTC**706741** LanChile  
CF6-80C2F  
Position 2  
LV-CDQ  
B767-300ER 2 Alerts  
NEWEST ALERT  
5 days ago  
20 Apr 2018 - 08:12 UTC**706961** LanChile  
CF6-80C2F  
Position 2  
CC-CZZ  
B767-300ER 2 Alerts  
NEWEST ALERT  
4 days ago  
21 Apr 2018 - 17:57 UTC**706725** LanChile  
CF6-80C2F  
Position 2  
CC-CWF  
B767-300ER 2 Alerts  
NEWEST ALERT  
5 days ago  
20 Apr 2018 - 14:34 UTC**707118** LanChile  
CF6-80C2F  
Position 1  
N420LA  
B767-300ER 1 Alert  
NEWEST ALERT  
7 days ago  
18 Apr 2018 - 15:41 UTC**704644** LanChile  
CF6-80C2F  
Position 1  
CC-CWF  
B767-300ER 1 Alert  
NEWEST ALERT  
5 days ago  
20 Apr 2018 - 14:33 UTC

# Fleet Monitor

We created a system that encouraged flexibility.

- 1 Filter by engine type if you have a technical role.
- 2 Or by airline if you have a customer-facing role.
- 3 Default the view to group alerts by engine.
- 4 Filter and sort flexibly.
- 5 Prioritize engines with the most and/or severe alerts.



6 ESN 706741

Aircraft Information

Aircraft Type	B767-300ER
Tail Number	LV-CDQ
Operator	LanChile
Operator Code	LAN

Engine Information

ESN	706741
Engine Series	CF6-80C2B6F
Engine Position	2
Last Installed	14 Apr 2011
Last Removed	Not Available
Sister Engine	704729
Engine Status	Installed

Times and Cycles

CSN	8371 cycles
TSN	45673 hours
CSSV	3890 cycles
TSSV	24219 hours
CSI	Not Available
TSI	Not Available

Last 90 Days

Cycles	96
Hours	813
Alerts	6

9



Alerts Cases Trends History Faults & Exceeds

Status Open Severity All Type All Sort By Severity Alert Date Last 7 days Flight Date Last 7 days Classification 2 selected

10 MEDIUM OPEN ALL\_ALL\_CF6-80C2F\_GRADUAL\_INCREASE\_DEGT  
 5 days ago 20 Apr 2018 - 08:12 UTC

MEDIUM OPEN ALL\_ALL\_CF6-80C2F\_SUDDEN\_INCREASE\_DEGT  
 5 days ago 20 Apr 2018 - 08:12 UTC

MEDIUM OPEN ALL\_ALL\_CF6-80C2F\_GRADUAL\_INCREASE\_DEGT  
5 days ago 20 Apr 2018 - 08:12 UTC

Detected By: SEA  
Fault Confidence: NA  
Flight Phase: CRUISE  
Parameter: NA  
Parameter Value: NA  
SEA007828

▼ Trends  
Alerting Parameters

# Fleet Monitor

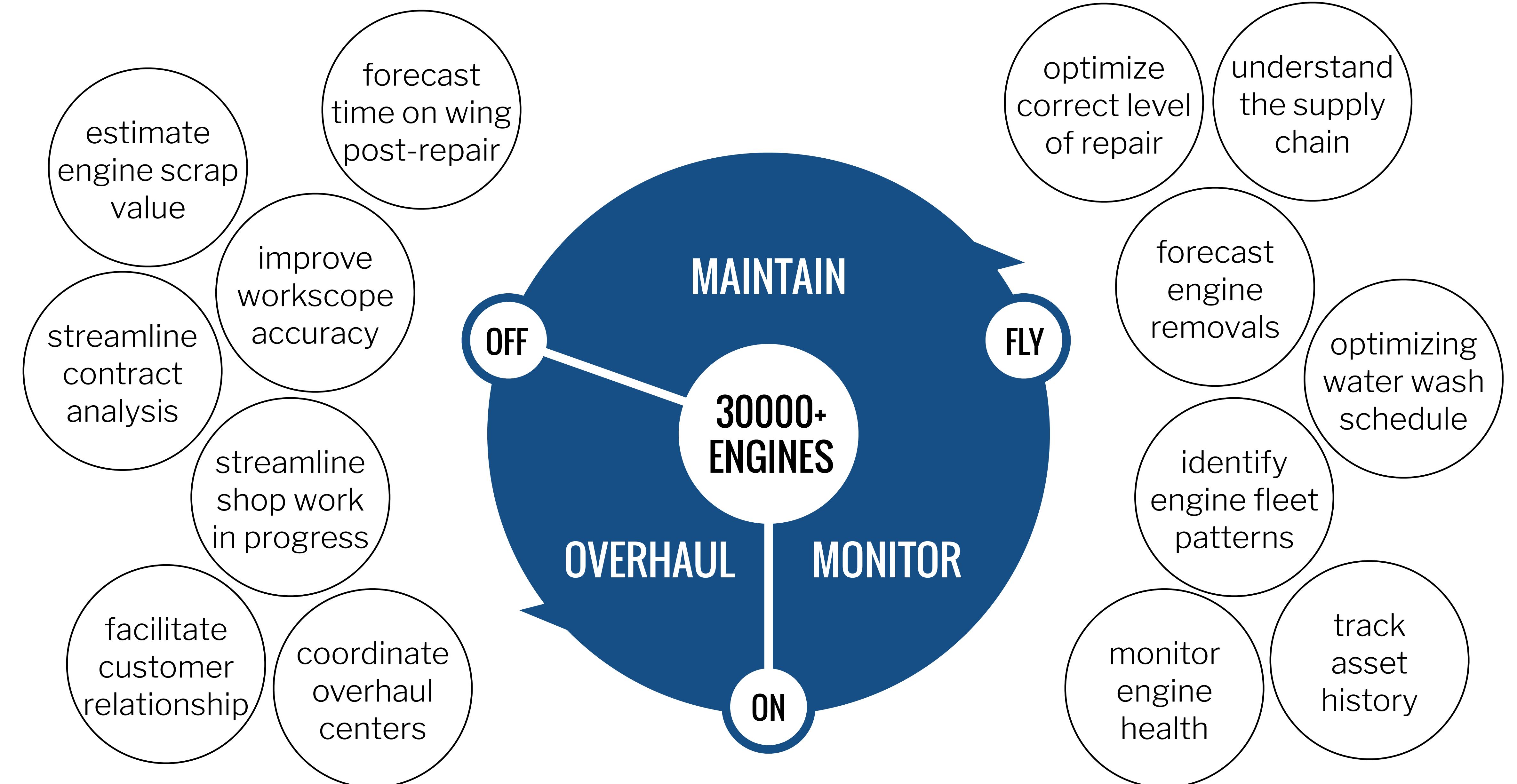
We presented the engine as a whole.

- 6 Focus analysis on the engine, not the alerts.
- 7 Introduce the concept of viewing by aircraft.
- 8 Provide access to the sister engine to compare operational performance.
- 9 Incorporate related data from other systems.
- 10 View all currently active alerts for this engine.

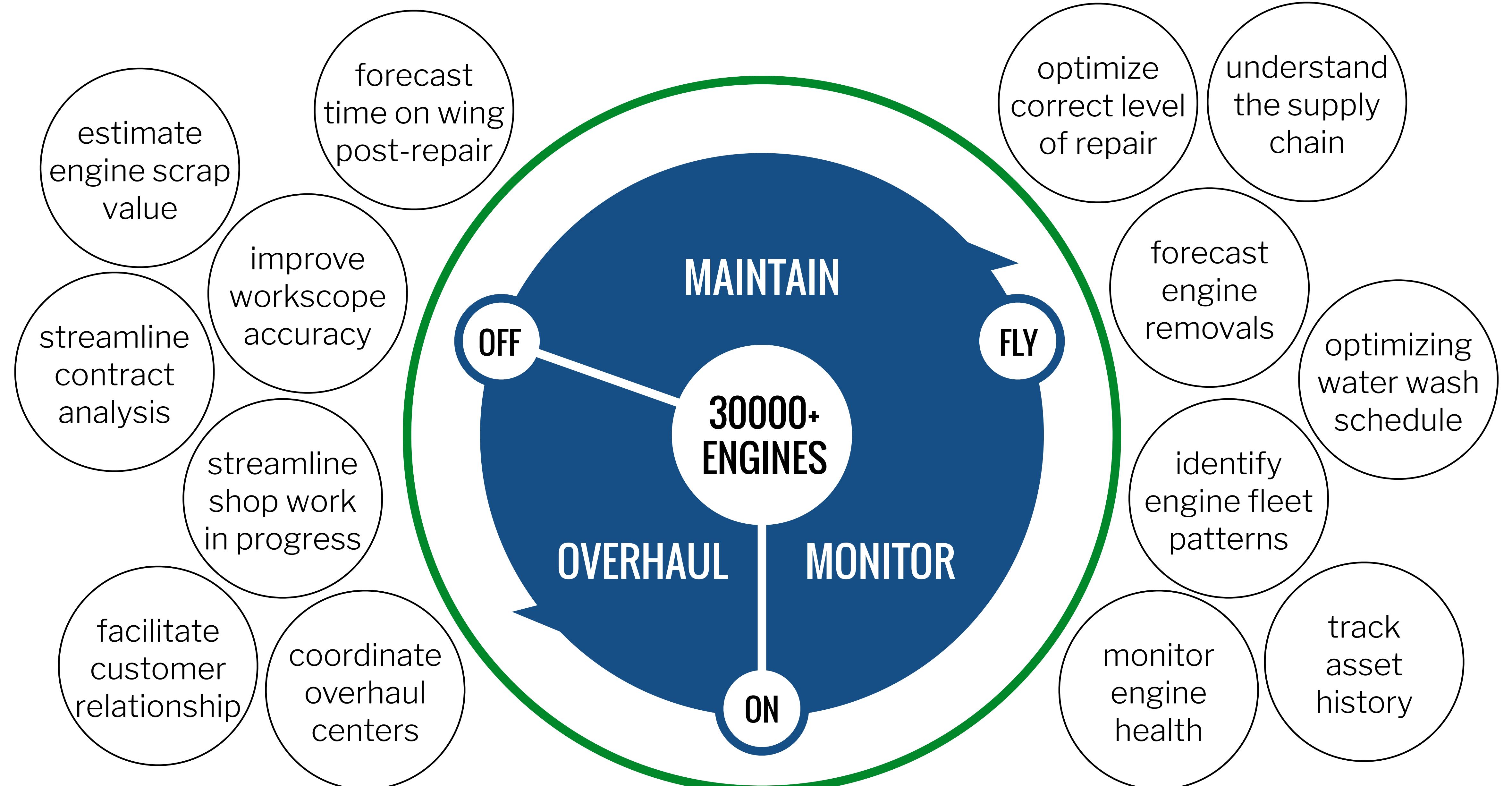
# Outcomes

Evolved how GE Aviation conceived of monitoring, from alert queue to fleet health. They even rebranded the group to be Fleet Support.

Simpler system reduced dependence on decades of expertise, which allowed opening more support centers globally and reducing response time.



**Built up a solid portfolio across the Aviation business...**



**...and the system started to sink in**

# **Proposed a new approach to planning**

## **From reactive planning...**

A new collection of projects every year that address the current hot topics.

## **...to proactive planning**

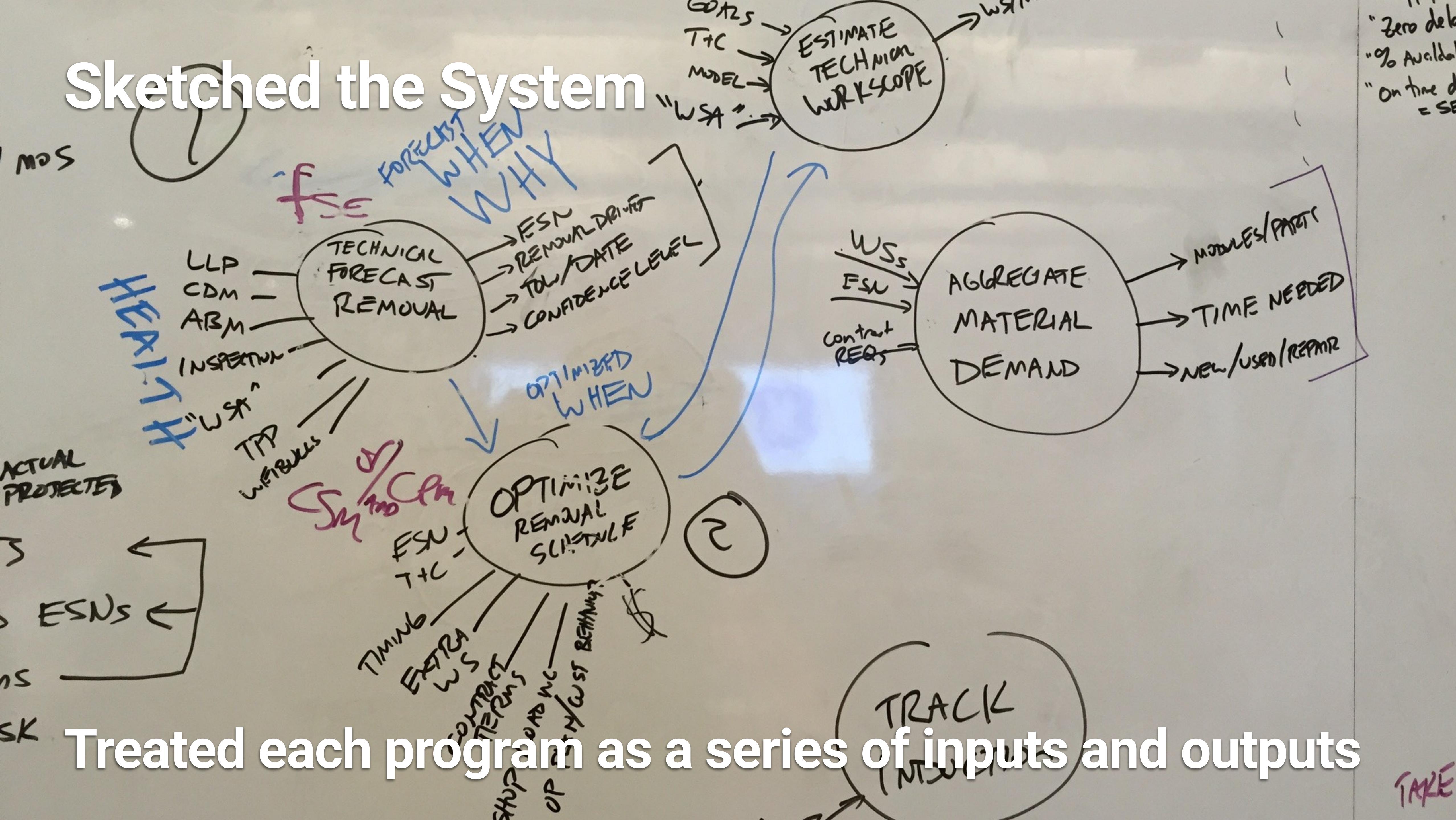
A systemic approach of how all these projects add up together to form a whole.



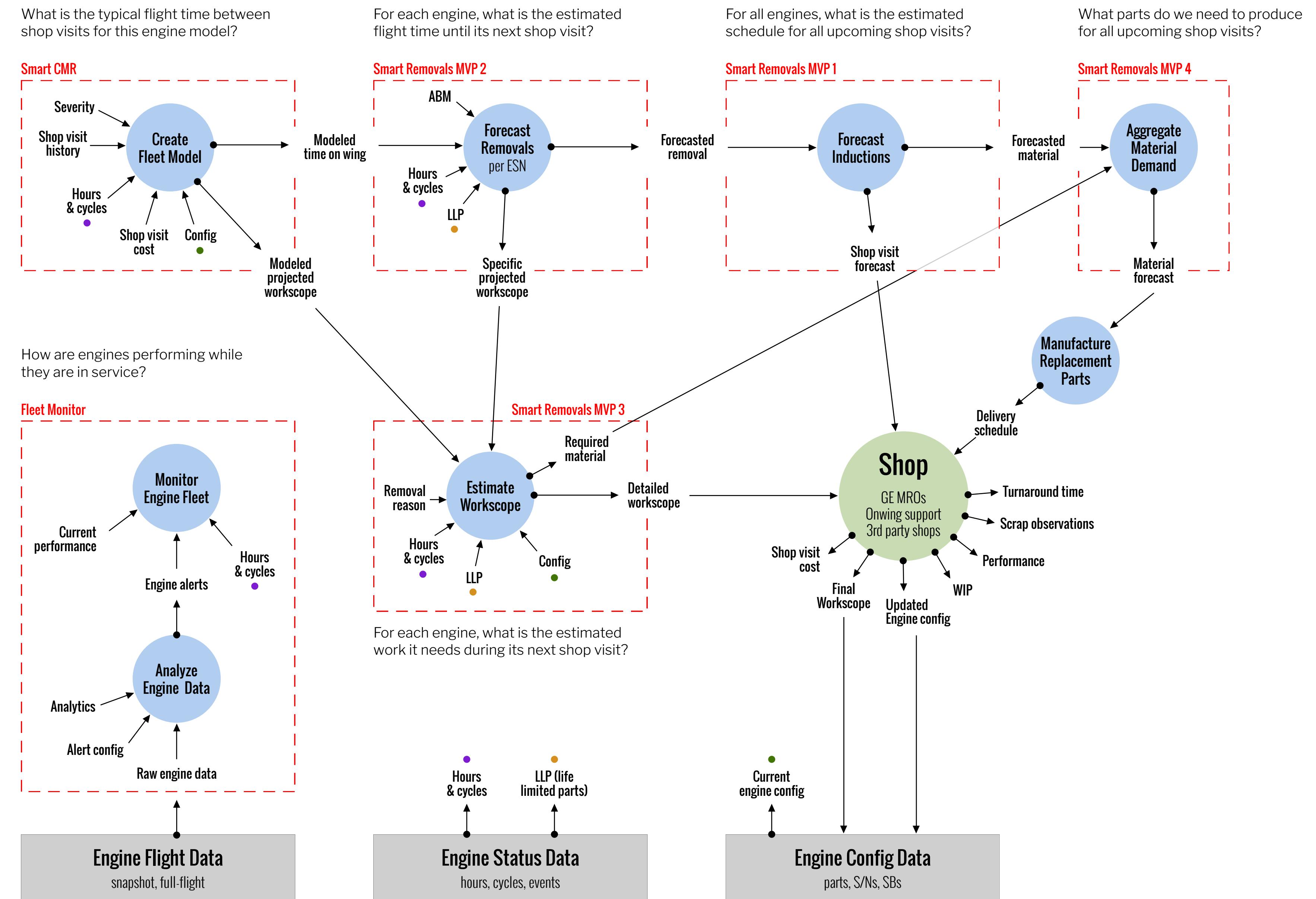
GE Aviation stakeholders in GE Design Center

# Stakeholder Workshops

# Sketched the System



# Distilled



=CRM(CAL)

FINANCIAL

# Sketched the Concept

=REPUBLIC

ONE PLAT

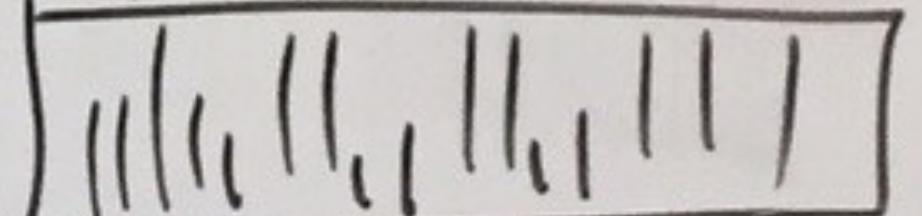
=CF34-8E

## REPUBLIC AIRLINES

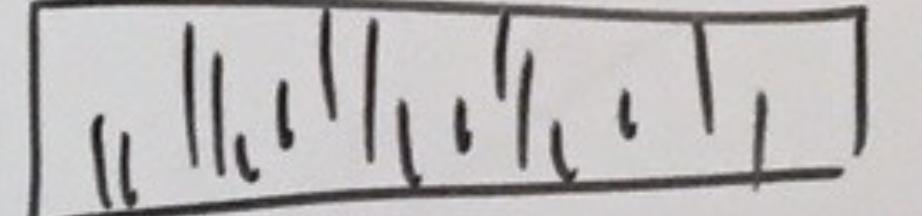
GLOBAL FLEET

129 TAILS    278 ENGINES    20 SPARES    3 OPERATORS

Cycles/month



hours/month



IMPACT \$ OM%? BASE?	ON WATCH 42	WIP 12	Available Spares 8	TOW "AUG" HIGHEST	COST RETRO 1.6 BILLING RETRO .45 DEFERRED BALANCE \$1.2M	Monthly Billing \$4.5M Avg Supplied Up \$1.2M w/o book
----------------------------	----------------	-----------	-----------------------	-------------------------	--	---

ESN	MARGIN	WATCH	Δ PREDICTION (150)	CSV 8500	RECOMMENDATION "STR-FWZ2"	CONFIDENCE 20% (0.2%)
123456 <span style="color:red">▲3</span>	8.9					
463111 <span style="color:green">▼1</span>	14.2		233		CEL-FW31	100% 2.32

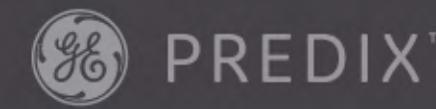
RATINGS

MOVARS

ORT BY  
PRIORITY:

RISK

# Designed the Vision



REPUBLIC AIRLINES ▾  
CF34-8E CSA

OVERVIEW ▾

GE34-8E	Op Parameters (Byr avg)	Fleet hours per month
176 Tails	Hours 2652	
380 Engines	Cycles 1890	
15 WIP	Flight Leg 1.40	
13 Spares		

CONTRACT	2015 CMR	FINANCIAL IMPACT	ON WATCH	43	FLEET ACTIONS
OM%	73.3%	Cost Retro 1.64	CSA Base \$xxMM	CDM / ABM / CNR 26	2X WW 100%
DB MM	\$159	Billing Retro .41	2015 FM, TOW \$xxMM	Reduced BSI 14	Climb Derate 85%
Past Dues	\$3MM	Shop Visits 23/61	2015 FM, Cost \$xxMM	Performance 3	Hot Spares 70%

REMOVAL RECOMMENDATIONS

ENGINES	RISK ▾	PLAN ▾
ESN 193729 Status: Off-wing LLP Life: 7166	ECSN: 17834 ECSV: 12349 FINANCIAL OPERATIONAL	Recommended Remove: IND - Sep 18, 2015 Induct: STR - Sep 25, 2015
ESN 193943		Scheduled

# Outcome: A Holistic Initiative

GE Aviation created a multi-year initiative: **DMRO**.  
(Digital Maintenance, Removal, and Overhaul)

Four of the applications that we worked on are still in operation today: Global Shop Visit Forecasting, Intelligent Slot Tracker, Enterprise Workscoping Tool, and Material Optimizer



**Zendesk**  
Workflow Experience

# About

Customer support SaaS pioneer.

Transitioning from multiple SMB products to an enterprise platform.

Workflow was the original heart of Zendesk, but also the deepest technical and experience debt.



# The Deepest Technical Debt



## THE ORIGINAL ZEN DESK

It's true that Zendesk was created to bring a sense of calm to the often chaotic world of customer service.

It's also true that Zendesk was born on a literal desk. Well, actually it was a door. And really, it was more of a kitchen table. A lot of beer was spilled on it. Innocence was lost here, direction and purpose found.

You probably don't want to touch it, but if you look close, you'll see the exact spot where Zendesk began.

# My Role

Help **Admins** to create workflows that  
Help **Customer Service Agents** to  
Help **End Customers** solve problems

# Context

- 1 “Triggers” were atomic rules with limited logic capabilities
- 2 ... thus requiring 100s or 1000s of triggers to replicate necessary business logic
- 3 ... and the only management tool offered was searching by title

## Triggers

Triggers take action when a ticket is created or updated. For example, use a trigger to send email notifications when an agent adds a comment to a ticket. The order of your triggers is important because triggers can act on changes made by other triggers. [Learn more](#)

Name	Last updated	Usage (last 7 days)	⋮
Notify group of assignment	2017-11-10	9	
Notify assignee of reopened ticket	2018-01-15	8	
Notify all agents of received request	2017-11-02	7	
Notify requester of comment update	2018-01-14	6	
Notify assignee of assignment	2018-01-15	5	
Example Answer Bot trigger for Nicolette Robichaud	2018-01-15	4	
Tonkean System Trigger	2017-02-12	3	
Ticket Form thing	2017-11-10	2	
Set high priority for Nicolette Robichaud	2017-11-10	1	

# Result

Admins were suffering.

Unsurprisingly, they avoided editing workflows because it was so difficult.

Worse? Nothing had changed in 10+ years.

# Customer Research

24-30 admins walked us through their workflows and challenges in managing them

A screenshot of a Zendesk support ticket list. The left sidebar shows navigation links for Apps, Internals, Workin' It, Brain Dev, and Know. The main area has a search bar for 'Search alex pruz' and a user profile for 'Alex Pruzinsky'. The ticket list is filtered by 'Agent' and shows 2265 tickets from Oct 14, 2019. The list includes various ticket categories such as [Ticket Category] Ticket Form is Finding or Purchasing Courses, [Ticket Category] Ticket Form is Student: Payments, charges and receipts, and [Ticket Category] Ticket Form is Refund. The ticket descriptions provide details like 'Set General SF Hours Schedule on Tickets with No Schedule', 'Issue with a charge', and 'Boleto question or issue (Brazil only)'. The ticket count for each category ranges from 2 to 3622.

Ticket Category	Date	Count
[Ticket Category] Ticket Form is Finding or Purchasing Courses → Ticket Category is S Discovery/Enrollment	Oct 14, 2019	2
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Issue with a charge	Oct 14, 2019	3622
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Payment error	Oct 14, 2019	7105
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Coupon or promotion issue	Oct 14, 2019	2730
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Currency	Oct 14, 2019	1174
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Missing course	Oct 14, 2019	916
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Boleto question or issue (Brazil only)	Oct 14, 2019	670
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Gift or redemption code issue	Oct 14, 2019	210
[Ticket Category] Ticket Form is Student: Payments, charges and receipts → Receipts and taxes	Oct 14, 2019	3031
[Ticket Category] Ticket Form is Issues while Taking a Course → Ticket Category is S Course Taking	Oct 14, 2019	4674
[Ticket Category] Ticket Form is Refund → Ticket Category is S Refund	Oct 14, 2019	127597
[Ticket Category] Ticket Form is Account/Profile Questions → Ticket Category is S Account/Profile	Oct 14, 2019	8801
[Ticket Category] Ticket Form is Account/Profile (Instructor) → Ticket Category is Ins Account/Profile	Oct 14, 2019	901

# Data Analysis

Performed the first textual analysis of triggers. First designer to ask for, and get, access to the data lake.

27,000+ accounts, 1,000,000+ triggers

Relearned python to build an analysis tool

```

binning = ', '.join(str(e) for e in bin)
print(str(i) + " : " + str(bin[0:5]) + "... " + str(bin[-5:]))
rules_sql = """SELECT id, account_id, definition FROM data_lake_classic.rules WHERE
deleted_at is null"""
sql_str = rules_sql.format(f_ids=bin_string)
rules_df = pd.read_sql(sql_str, conn)
tr1 = rules_df.shape[0]
trt += tr1
print("{} triggers, {} total".format(tr1,trt))
acts_l = []
alls_l = []
anyss_l = []
defs_packed = []
fireOn_l = []

definition_list = rules_df['definition'].tolist()
for definition in definition_list:
    temp_def = process_rule(definition)
    temp_acts = temp_def['acts']
    temp_all = temp_def['alls']
    temp_anys = temp_def['anyss']
    if "onCreate" in temp_all:
        fireOn_tmp = "onCreate"
    elif "onChange" in temp_all:
        fireOn_tmp = "onChange"
    else:
        fireOn_tmp = "onEvery"
    defs_packed.append(item)

```

## Tiers of complexity

Triggers	# Accounts	% Total
10-24	18,618	67.3%
25-49	5,093	18.4%
50-99	2,370	8.6%
100-249	1,173	4.2%
250-499	286	1.0%
500-999	101	0.4%
1000+	38	0.1%
Total	27,679	
Or, in terms of thresholds.		
Triggers	# Accounts	% Total
10+	27,679	100.0%
25+	9,061	32.7%
50+	3,968	14.3%
100+	1,598	5.8%
250+	425	1.5%
500+	139	0.5%
1000+	38	0.1%

## Highest customer usage

account_id	account_name (not necessarily customer name)	# triggers
2144605		33665
1470967		4820
1908154		3383
888110		3114
34989		3063
9618821		3030
2184366		3026
2046910		2935
9519405		2774
9163788		2679
2363060		2674
1208524		2574
2097762		2192
9114058		2124
761341		2106
5194		1989
2201375		1981

## Fields used most often

Source	Count	% Usage
Total	6,046,092	
Custom ticket field	1,050,433	17.4%
Tags	623,491	10.3%
Ticket	598,545	9.9%
Status	532,930	8.8%
Group	459,427	7.6%
Comment text	311,798	5.2%
Assignee	304,887	5.0%
Email user	275,791	4.6%
Subject text	212,875	3.5%
Received at	199,321	3.3%
Target	188,495	3.1%
Form	177,688	2.9%
Organization	139,924	2.3%
Comment is	136,559	2.3%
Update via	105,570	1.7%
Priority	102,183	1.7%

## Field usage by tier

Trigger Tiers	Total	10-24	25-49	50-99	100-249	250-499	500-999	1000+
# accounts	27,679	67.3% 18,618	18.4% 5,093	8.6% 2,370	4.2% 1,173	1.0% 286	0.4% 101	0.1% 38
# triggers	1,052,494	25.3% 265,923	16.5% 173,431	15.5% 163,619	16.6% 174,564	9.3% 97,596	6.7% 70,390	10.2% 106,971
<b>Ticket status</b>								
Fire on new tickets only	36.3%	30.7% 81707	34.3% 59423	37.5% 61335	40.4% 70439	43.3% 42231	32.2% 22699	41.5% 44436
Fire on updated tickets only	13.7%	13.5% 35858	11.6% 20132	11.2% 18398	13.2% 23040	11.4% 11141	12.6% 8870	24.7% 26447
Ticket status doesn't matter	50.0%	55.8% 148358	54.1% 93876	51.3% 83886	46.5% 81175	45.3% 44224	55.2% 38821	33.7% 36088
<b>Group (group_id)</b>								
in Actions	22.4%	12.1% 32245	22.1% 38302	24.8% 40539	26.1% 45548	31.8% 31080	28.8% 20286	25.7% 27528
in Conditions All	11.2%	8.4% 22225	10.3% 17889	11.7% 19098	14.0% 24421	16.3% 15886	12.5% 8806	9.1% 9689
in Conditions Any	2.8%	3.4% 8966	2.3% 4045	2.5% 4066	3.1% 5358	3.7% 3606	2.7% 1923	1.2% 1241
<b>Assignee (assignee_id)</b>								
in Actions	9.1%	8.2% 21844	10.7% 18582	11.2% 18381	10.3% 18042	9.0% 8745	6.0% 4195	5.8% 6195
in Conditions All	11.1%	22.1% 58757	11.8% 20413	8.3% 13614	7.0% 12190	6.4% 6293	3.5% 2460	2.6% 2743
in Conditions Any	1.5%	3.2% 8429	1.3% 2260	1.0% 1607	0.8% 1434	1.2% 1170	1.0% 690	0.1% 69
<b>Priority (priority_id)</b>								
in Actions	8.0%	5.9% 15814	9.8% 16942	10.5% 17176	9.5% 16507	7.4% 7258	6.7% 4748	5.2% 5605
in Conditions All	1.3%	1.3% 3565	1.8% 3195	1.6% 2699	1.6% 2785	1.0% 983	0.7% 484	0.3% 356
in Conditions Any	0.2%	0.2% 554	0.2% 419	0.2% 388	0.3% 457	0.4% 371	0.5% 364	0.0% 12
<b>Status (status_id)</b>								
in Actions	9.8%	6.6% 17622	10.2% 17635	11.3% 18494	10.9% 19024	10.4% 10153	11.8% 82771	11.6% 12408
in Conditions All	33.1%	38.5% 102422	34.3% 59567	31.7% 51787	28.4% 49644	27.9% 27190	24.4% 17165	37.6% 40203
in Conditions Any	2.0%	2.8% 7344	2.5% 4292	2.3% 3726	1.7% 2920	1.4% 1399	1.0% 723	0.1% 138

# Key Insight: Clones

The most interesting insight was discovering “cloned triggers,” triggers that were logically identical and followed the same pattern.

**40% of all triggers were clones.**

**Cloned triggers**

account_id	account_name (not necessarily customer name)	# triggers	# unique triggers	% clones
2144605		33665	13	100.0%
1908154		3383	15	99.6%
2019803		152	1	99.3%
874624		999	9	99.1%
9853087		996	9	99.1%
729218		215	2	99.1%
998695		105	1	99.0%
2201375		1981	20	99.0%
2046910		2935	30	99.0%
9660190		708	10	98.6%
9165692		186	3	98.4%
515664		1614	32	98.0%
9242597		396	8	98.0%
2153462		97	2	97.9%

For example,

Company A has 2932 triggers, 99% of which are logically:

```
if (status == X) and (organization == Y)  
    then custom_field.vip = Z
```

Company B has 850 triggers, 40% of which are logically:

```
if (ticket_form == X) and (theatre_visited == Y)  
    then group = Z
```

Company C has 683 triggers, 79% of which are logically:

```
if (recipient == X)  
    then group = Z
```

**In these common examples, changing one piece of business logic would require editing hundreds of triggers.**

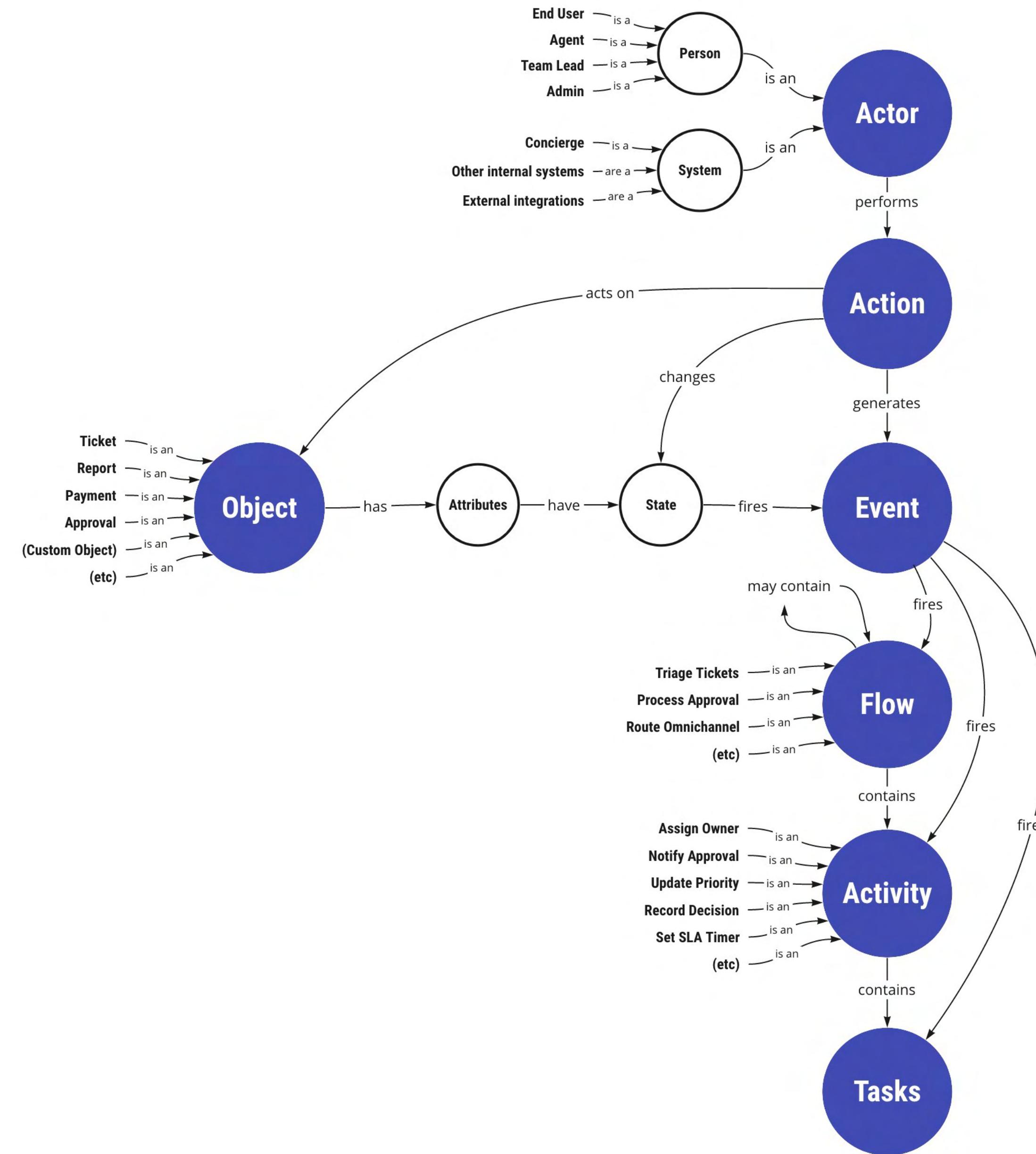
# Core Research Insight

We needed to provide **modularity**.

To upscale from SMB to enterprise,  
to handle more complex workflows,  
to handle more varied channels (messaging, chat, social media...),  
to reduce challenges for admins,  
and frankly to catch up to the competition,  
Zendesk needed to build modularity into its workflow offerings.

# Concept Model

Partnered with system architects to create a concept model of their future architecture.



# Proposed Initiatives

## Events



available event types are populated in

## Admin Center

Access Sunshine Workflows from the Admin Center.



EXISTING

## Event Maps

Configure how to handle incoming events and determine which workflows they fire.



PHASE 1

## Workflow Dashboard

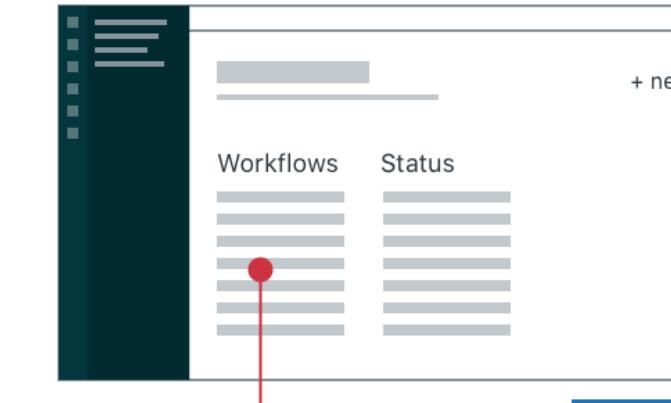
Dashboard displaying current operational metrics and status of active workflows.



PHASE 4

## Workflows

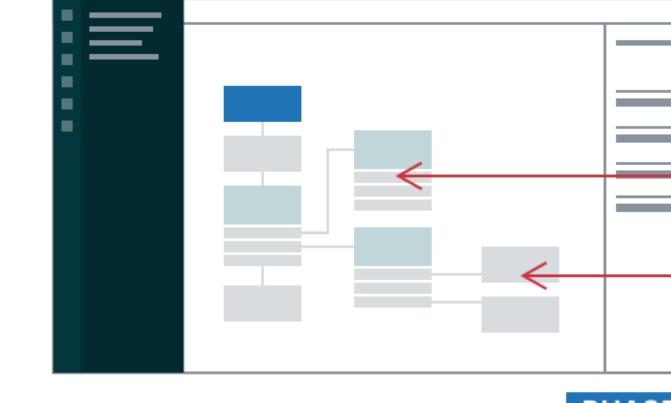
View the list of workflows created, their status, and access to create new workflows.



PHASE 2

## Workflow Builder

Visually build workflows by combining data, conditions and actions.



PHASE 2

## Workflow Library

Access components pre-built (by Zendesk or partners) to use in creating workflows.



PHASE 3

## Support Trigger Admin

admins can use their custom conditions in Trigger admin

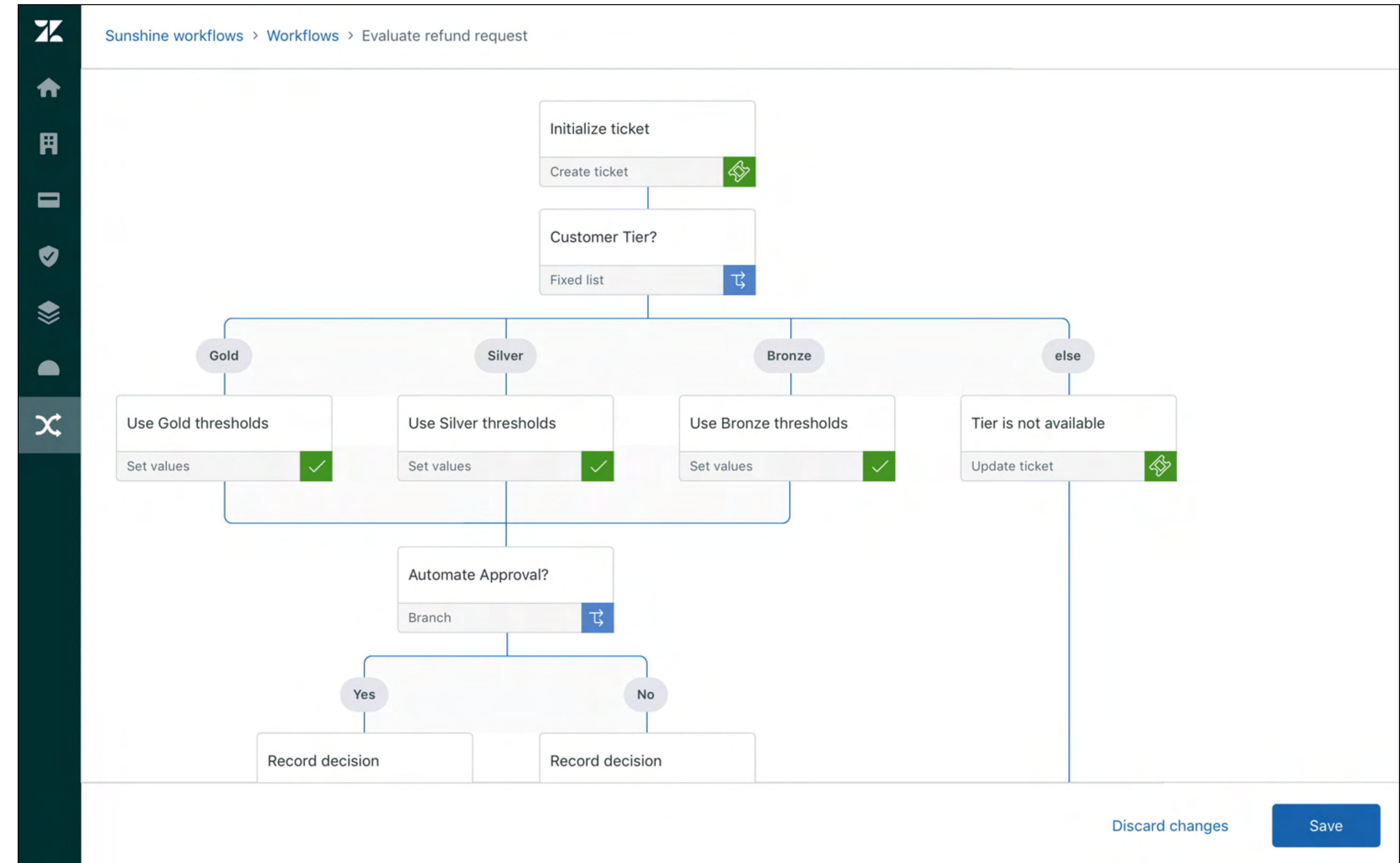


Proposed a series of initiatives that would lead to a competitive workflow system.

# Workflow Builder

Aligned three different product teams on a single approach to a visual workflow builder.

Started with design-only workshops to find common patterns, then worked with product managers to develop a common benefit.



...and then **the pandemic** started.

Every long-term initiative stopped.

It was time to focus on targeted fixes.

# Trigger Filters

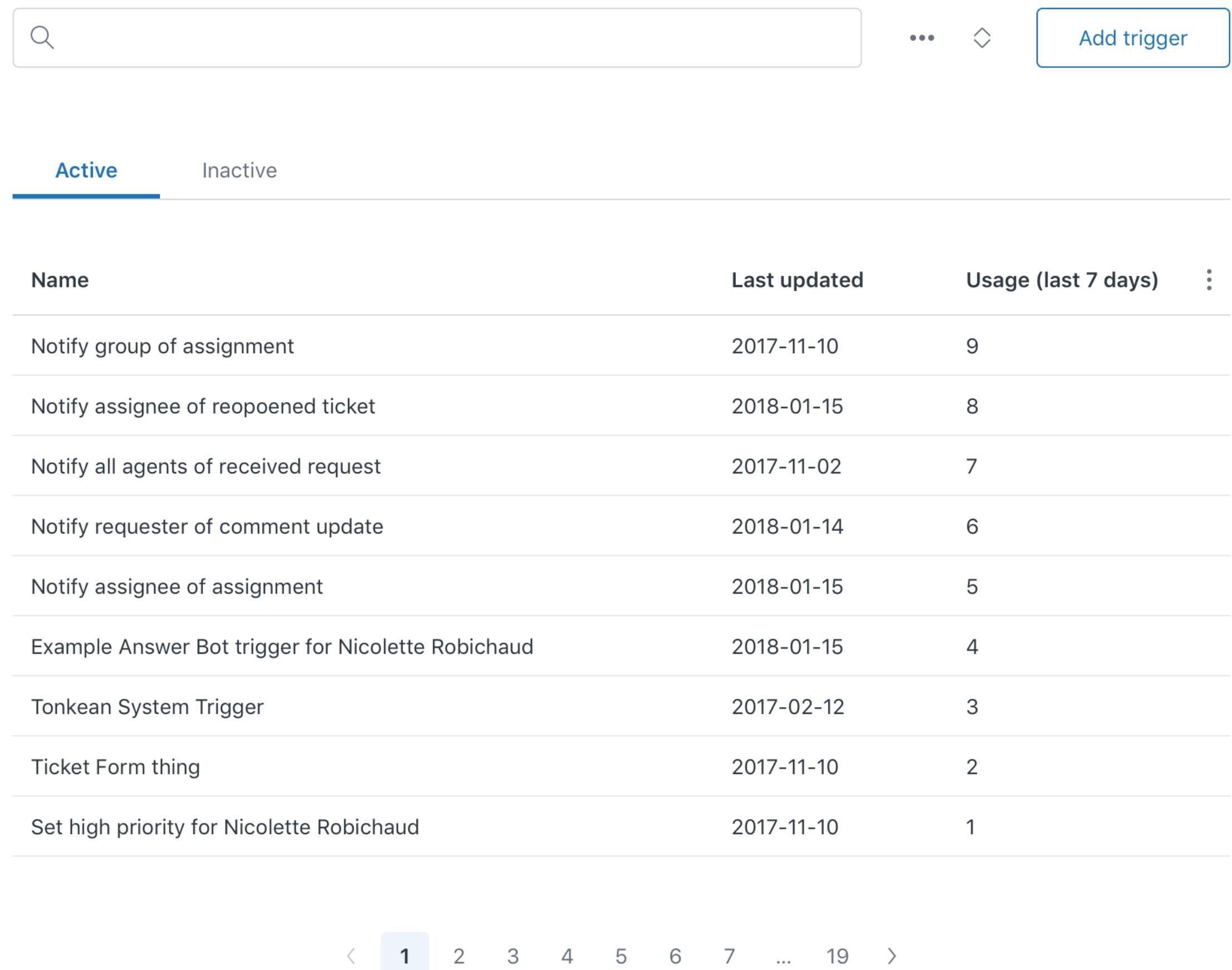
Scaled back to focus on the trigger list.

From our research, we knew admins need to find triggers that check specific fields (conditions) or set specific fields (actions).

Designed and introduced filtering that leveraged existing admin behavior of relying heavily on the search bar.

## Triggers

Triggers take action when a ticket is created or updated. For example, use a trigger to send email notifications when an agent adds a comment to a ticket. The order of your triggers is important because triggers can act on changes made by other triggers. [Learn more](#)



The screenshot shows the Zendesk Triggers list. At the top right is a search bar with a magnifying glass icon, followed by three dots, a dropdown arrow, and a blue 'Add trigger' button. Below the search bar are two filter buttons: 'Active' (which is underlined in blue) and 'Inactive'. The main area is a table with columns: 'Name', 'Last updated', 'Usage (last 7 days)', and a three-dot menu icon. There are nine rows of trigger data:

Name	Last updated	Usage (last 7 days)	⋮
Notify group of assignment	2017-11-10	9	
Notify assignee of reopened ticket	2018-01-15	8	
Notify all agents of received request	2017-11-02	7	
Notify requester of comment update	2018-01-14	6	
Notify assignee of assignment	2018-01-15	5	
Example Answer Bot trigger for Nicolette Robichaud	2018-01-15	4	
Tonkean System Trigger	2017-02-12	3	
Ticket Form thing	2017-11-10	2	
Set high priority for Nicolette Robichaud	2017-11-10	1	

At the bottom center is a navigation bar with page numbers: < 1 2 3 4 5 6 7 ... 19 >

Original text

# Trigger Filters

From this live example, we can see how the feature works.

People can still simply start with the name of the trigger to find matches. However, they can also now enter the text of the description, or the details from the conditions or actions entered.

The feedback from this was huge.

(Categories were also included, although that was a related effort in which I had professional say, yet another designer worked on it.)

## Triggers

Set up event-based rules that run every time a ticket is created or updated. Popular triggers include notifying customers when a new comment is added to their ticket or an out-of-office reply. [Learn about triggers](#)

Edit order

Add trigger

The screenshot shows the Zendesk Triggers list interface. At the top, there is a filter section with a dropdown set to 'Name' and a search input field. Below the filter, it says 'Showing 318 triggers'. There are two tabs: 'Active' (which is selected) and 'Inactive'. The main list is divided into sections: 'Name' (with a dropdown arrow), 'Notifications' (with a dropdown arrow and a count of 7), and 'Categories' (with a dropdown arrow and a count of 34). Under 'Notifications', there are eight items, each with a checkbox and a description: 'Notify assignee of assignment', 'Notify assignee of comment update', 'Notify assignee of reopened ticket', 'Notify requester and CCs of comment update', 'Notify requester and CCs of received request', 'Notify requester of new proactive ticket', 'Notify all agents of received request', and '+ Add trigger'. Under 'Categories', there are two items, each with a checkbox and a description: 'Check if 265/35ZR22' and 'Check if 235/55R20'.

Live example

# Trigger Filters

From this live example, we can see how the feature works.

People can still simply start with the name of the trigger to find matches. However, they can also now enter the text of the description, or the details from the conditions or actions entered.

The feedback from this was huge.

(Categories were also included, although that was a related effort in which I had professional say, yet another designer worked on it.)

## Triggers

Set up event-based rules that run every time a ticket is created or updated. Popular triggers include notifying customers when a new comment is added to their ticket or an out-of-office reply. [Learn about triggers](#)

Edit order

Add trigger



Filter Field Operator

Conditions Comment text (any) X

Add another filter Showing 8 triggers

Active Inactive

Name :

Check if 225/60R18	Categories
<input type="checkbox"/>	

Check if 235/55R19	Categories
<input type="checkbox"/>	

Check if 235/55R20	Categories
<input type="checkbox"/>	

Check if 245/45R20	Categories
<input type="checkbox"/>	

Check if 255/45R20	Categories
<input type="checkbox"/>	

Check if P255/45RF20	Categories
<input type="checkbox"/>	

Check if 265/35ZR22	Categories
<input type="checkbox"/>	

Check if 265/40ZR21	Categories
<input type="checkbox"/>	

⋮

Live example

# Trigger Filters

From this live example, we can see how the feature works.

People can still simply start with the name of the trigger to find matches. However, they can also now enter the text of the description, or the details from the conditions or actions entered.

The feedback from this was huge.

(Categories were also included, although that was a related effort in which I had professional say, yet another designer worked on it.)

## Triggers

Set up event-based rules that run every time a ticket is created or updated. Popular triggers include notifying customers when a new comment is added to their ticket or an out-of-office reply. [Learn about triggers](#)

The screenshot shows the Zendesk Triggers page. At the top right are buttons for 'Edit order', 'Add trigger', and a dropdown menu. Below is a search interface with 'Filter' (set to 'Actions'), 'Field' (set to 'Location'), and a 'Value' input field containing a dropdown menu. The dropdown menu is open, showing a list of locations: Topeka, Denver, Fayetteville, Cheyenne, Austin, San Antonio, and St. Louis. A placeholder '(any)' is at the top of the list. Below the dropdown are tabs for 'Active' and 'Inactive', and a section for 'Name' with a dropdown arrow. Under 'Name' are seven entries, each with a checkbox and a 'Locations' button: Direct to Topeka, Direct to St. Louis, Direct to Fayetteville, Direct to Austin, Direct to San Antonio, Direct to Denver, and Direct to Cheyenne.

Live example

# Trigger Filters

From this live example, we can see how the feature works.

People can still simply start with the name of the trigger to find matches. However, they can also now enter the text of the description, or the details from the conditions or actions entered.

The feedback from this was huge.

(Categories were also included, although that was a related effort in which I had professional say, yet another designer worked on it.)

A screenshot of the Zendesk trigger filter interface. At the top, there is a search bar labeled "Filter" with "Name" selected. Below the search bar, it says "Showing 873 triggers".

A screenshot of the Zendesk trigger filter interface. The search bar shows "notify". Below the search bar, it says "Showing 8 triggers".

A screenshot of the Zendesk trigger filter interface. The filter section includes "Filter" (set to "Conditions"), "Field" (set to "Status"), "Operator" (set to "Is"), and "Value" (set to an empty input field). A dropdown menu is open, showing options: "(any)" (selected), New, Open, Pending, Solved, and Closed.

A screenshot of the Zendesk trigger filter interface. The filter section includes "Filter" (set to "Actions"), "Field" (set to "Priority"), and "Value" (set to an empty input field). A dropdown menu is open, showing options: "(any)" (selected), Low, Normal, High (selected), and Urgent.

Live example

# Custom Ticket Status

Another deep technical debt problem with a very tight set of constraints.

Customers had complained for over a decade that they needed more than the fixed list of ticket statuses.

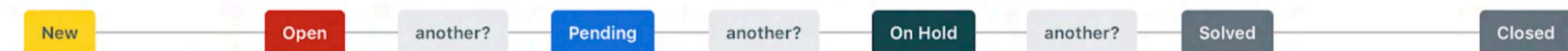
Led research and mentored a designer to work through the problem. Talked with 12 admins to diagram out how they would use custom ticket statuses.

Explored how adding custom statuses would affect several areas of the product, the agent experience, editing views, admin experience, and more.

Original fixed list



Did customers need additional statuses in the sequence?



Or did they need variations for describing the core statuses?



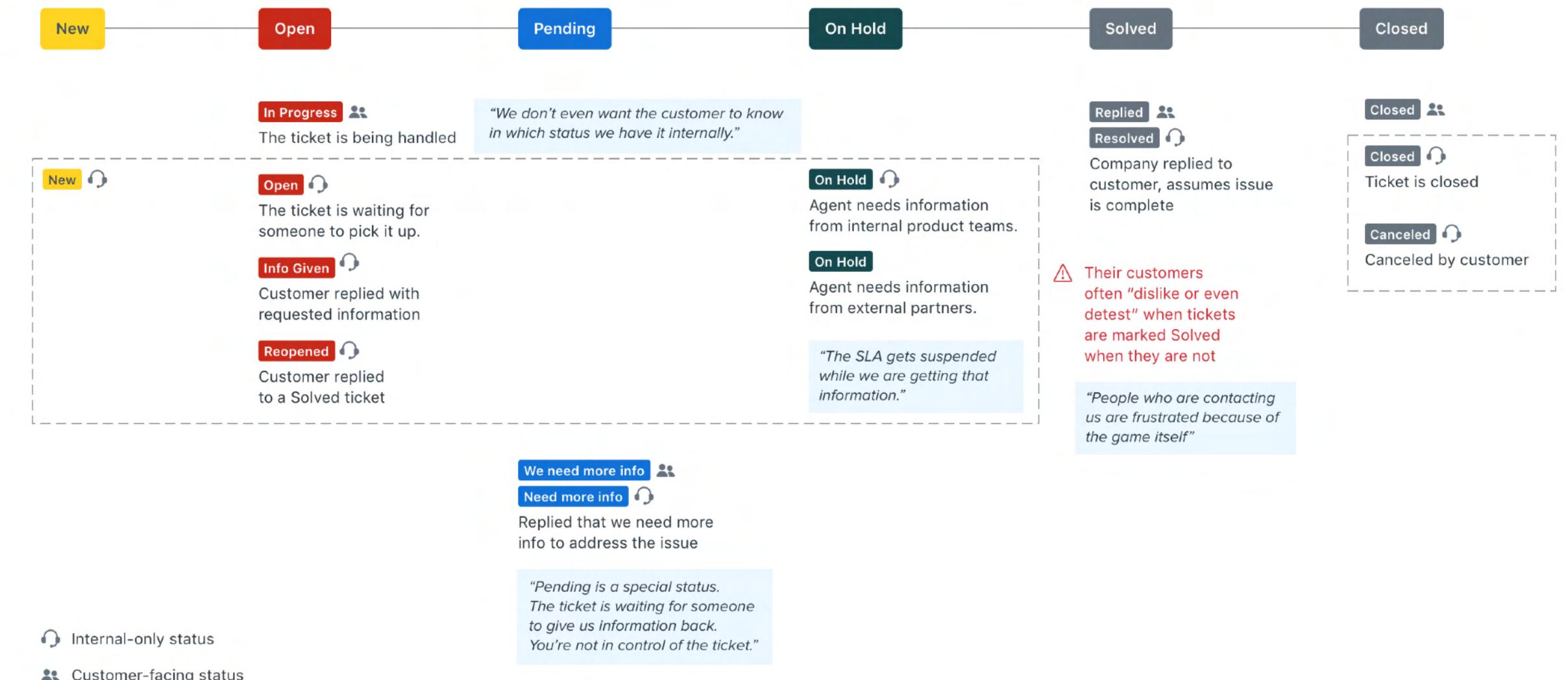
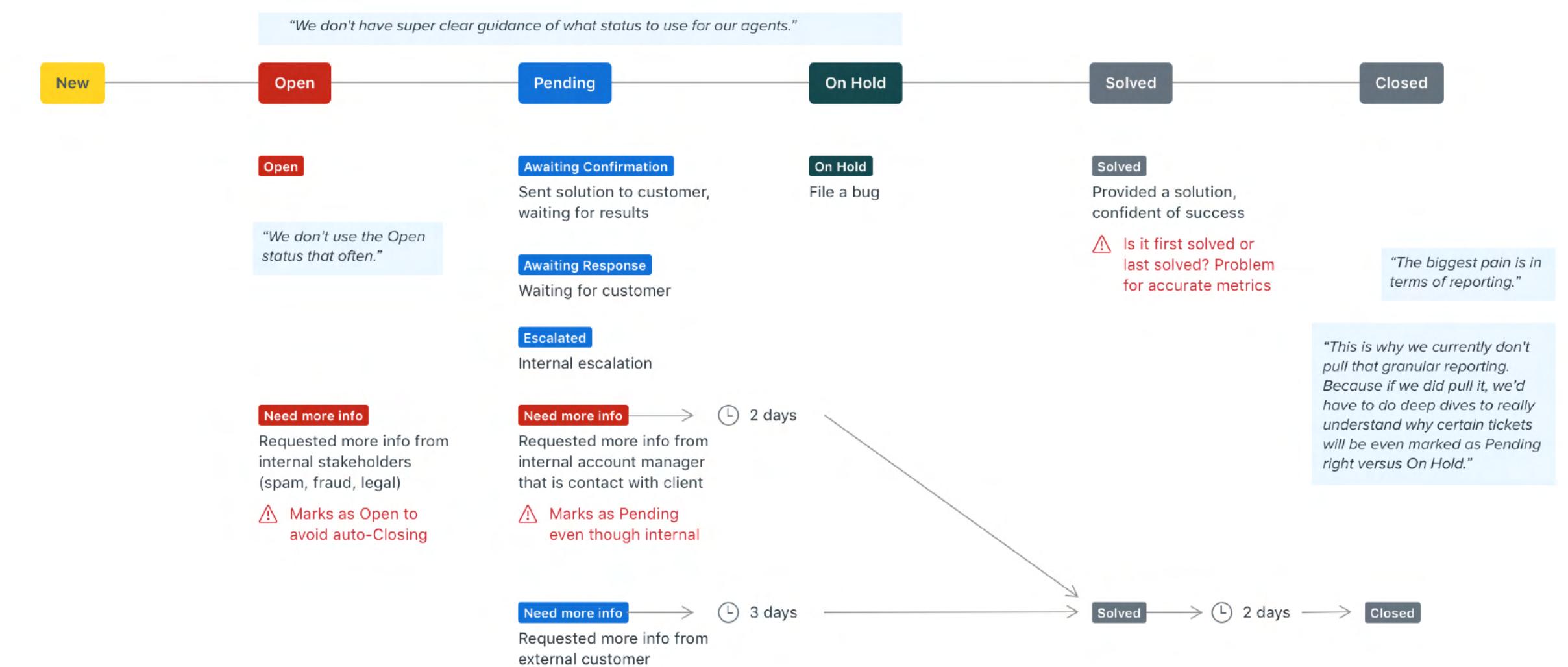
# Custom Ticket Status

Another deep technical debt problem with a very tight set of constraints.

Customers had complained for over a decade that they needed more than the fixed list of ticket statuses.

Led research and mentored a designer to work through the problem. Talked with 12 admins to diagram out how they would use custom ticket statuses.

Explored how adding custom statuses would affect several areas of the product, the agent experience, editing views, admin experience, and more.



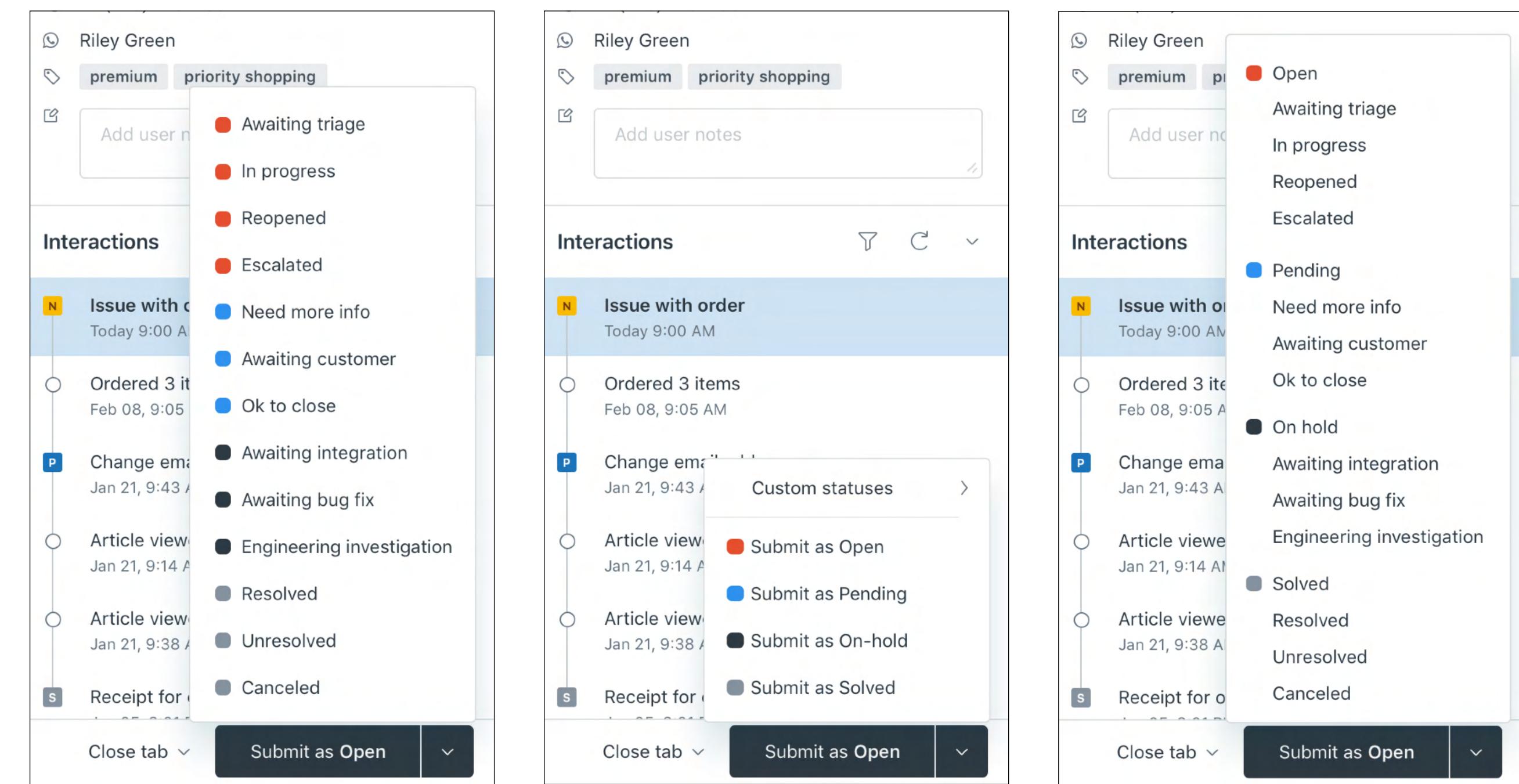
# Custom Ticket Status

Another deep technical debt problem with a very tight set of constraints.

Customers had complained for over a decade that they needed more than the fixed list of ticket statuses.

Led research and mentored a designer to work through the problem. Talked with 12 admins to diagram out how they would use custom ticket statuses.

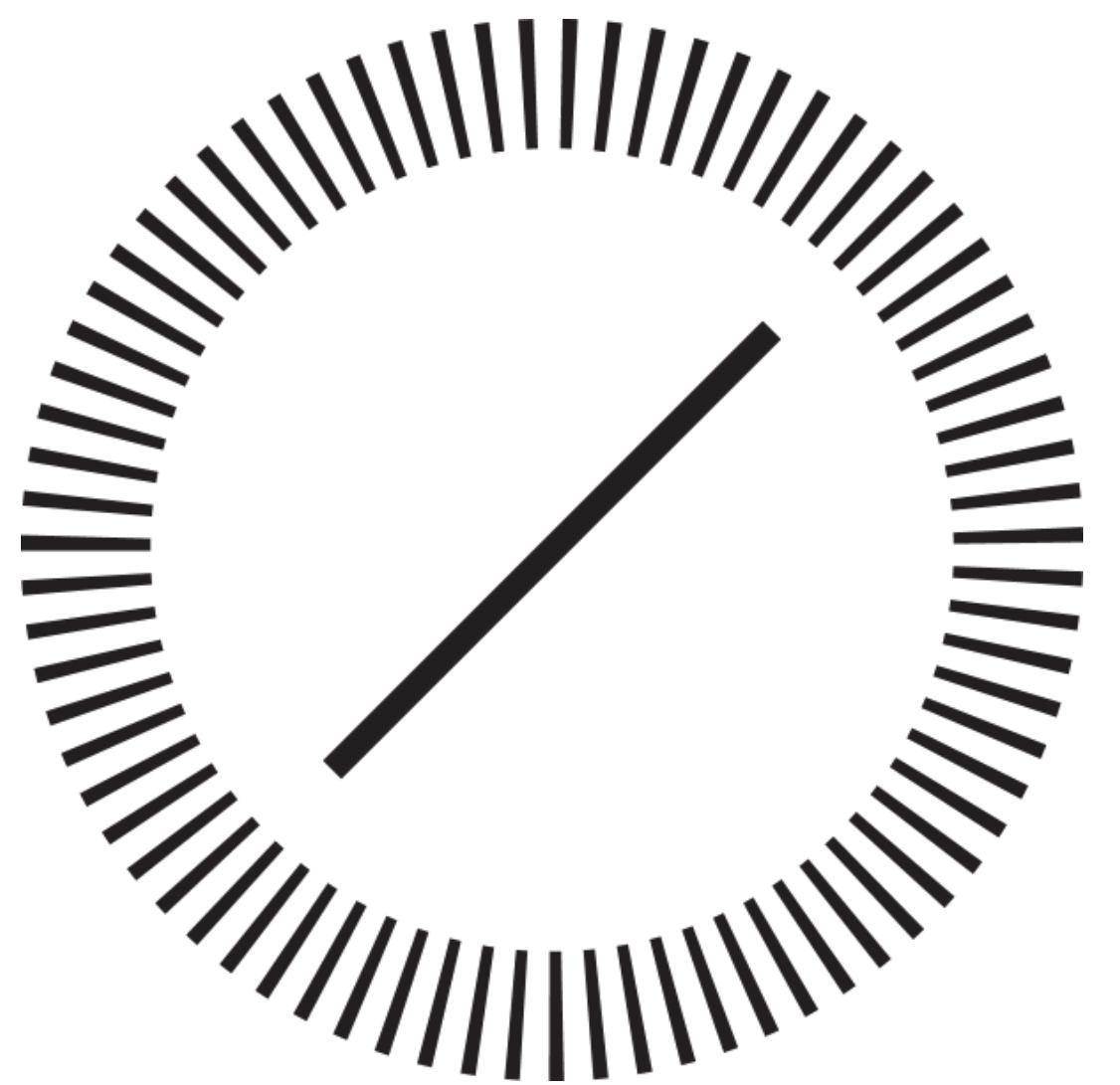
Explored how adding custom statuses would affect several areas of the product, the agent experience, editing views, admin experience, and more.



# Outcome: Product Momentum

Several product features that I kicked off have continued to be developed and hopefully will address admin pain.

Visual workflow builder, custom ticket status, custom conditions, and the general push to make workflow more holistic.



# **Compass Realty**

## Agent Jobs & Tasks

# About

High-end real estate company

Goal is to be the industry platform  
for real estate transactions

Struggling to incorporate two  
key acquisitions made in 2021



# Situation

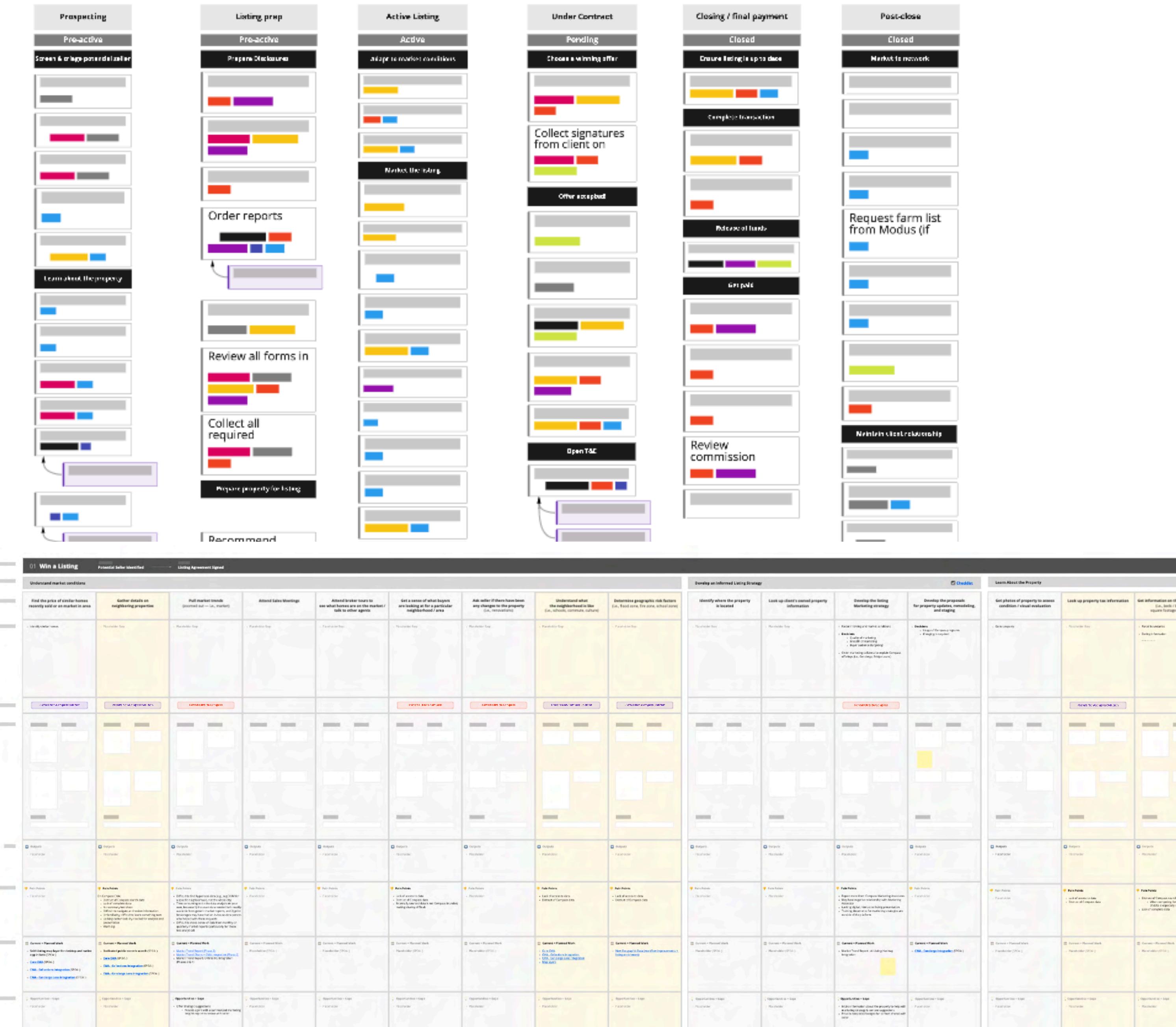
They asked for two things:

1. Map the customer journey of real estate agents  
(via the design leader)
2. Data for prioritizing the development roadmap  
(via the engineering leader)

# Prior Research

This was not the first attempt at mapping an agent's journey. Most of this research already existed, just not coherently.

Also, the results were too overwhelming for colleagues outside the research team to easily consume or act upon.



# Insights

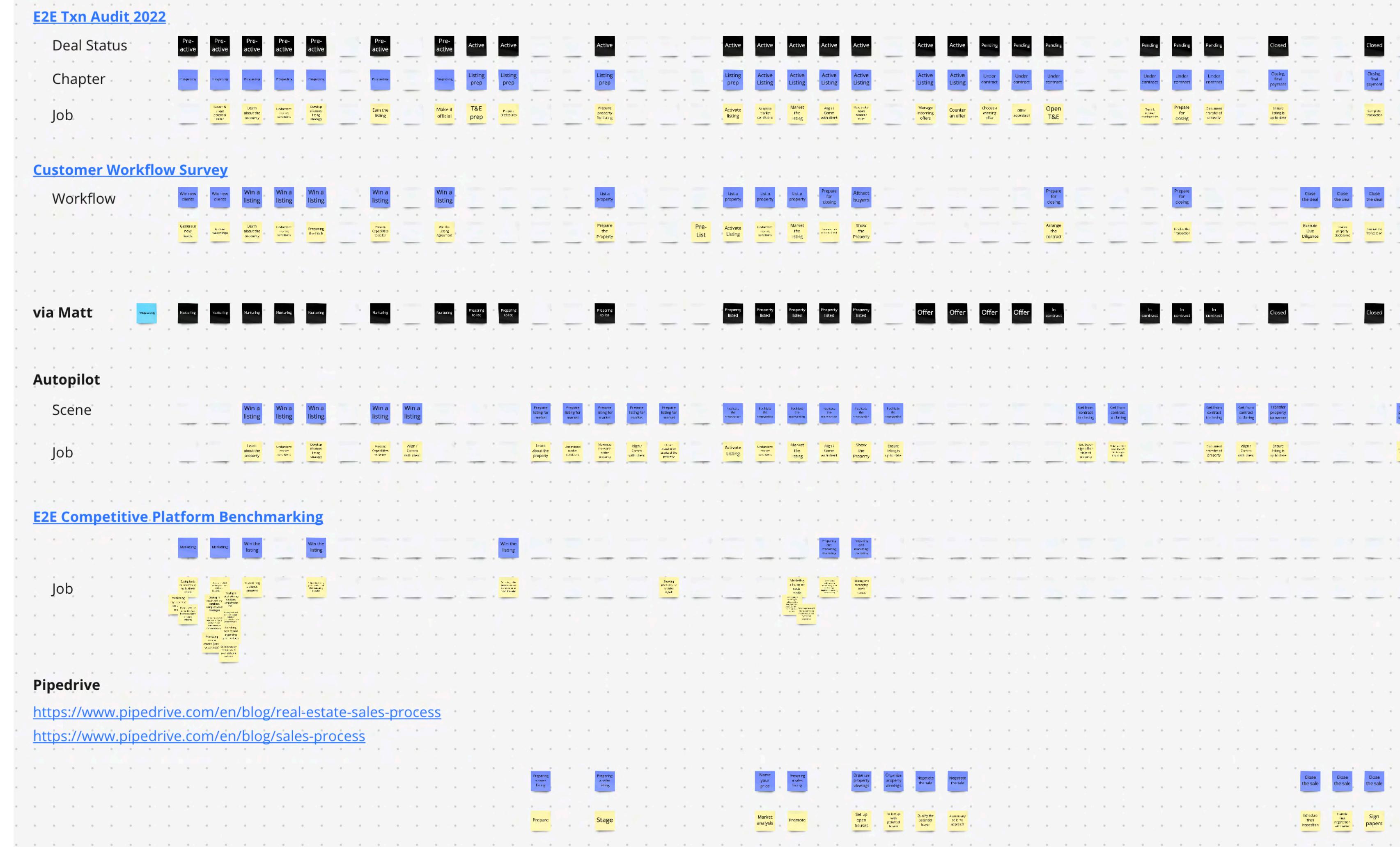
They suffered from a lack of single source of truth. Nobody was wrong but they had too many versions of “right”.

They weren’t speaking the same language.  
Even “transaction” had 3-4 different meanings.

Without a shared framework, they were perpetuating knowledge silos.

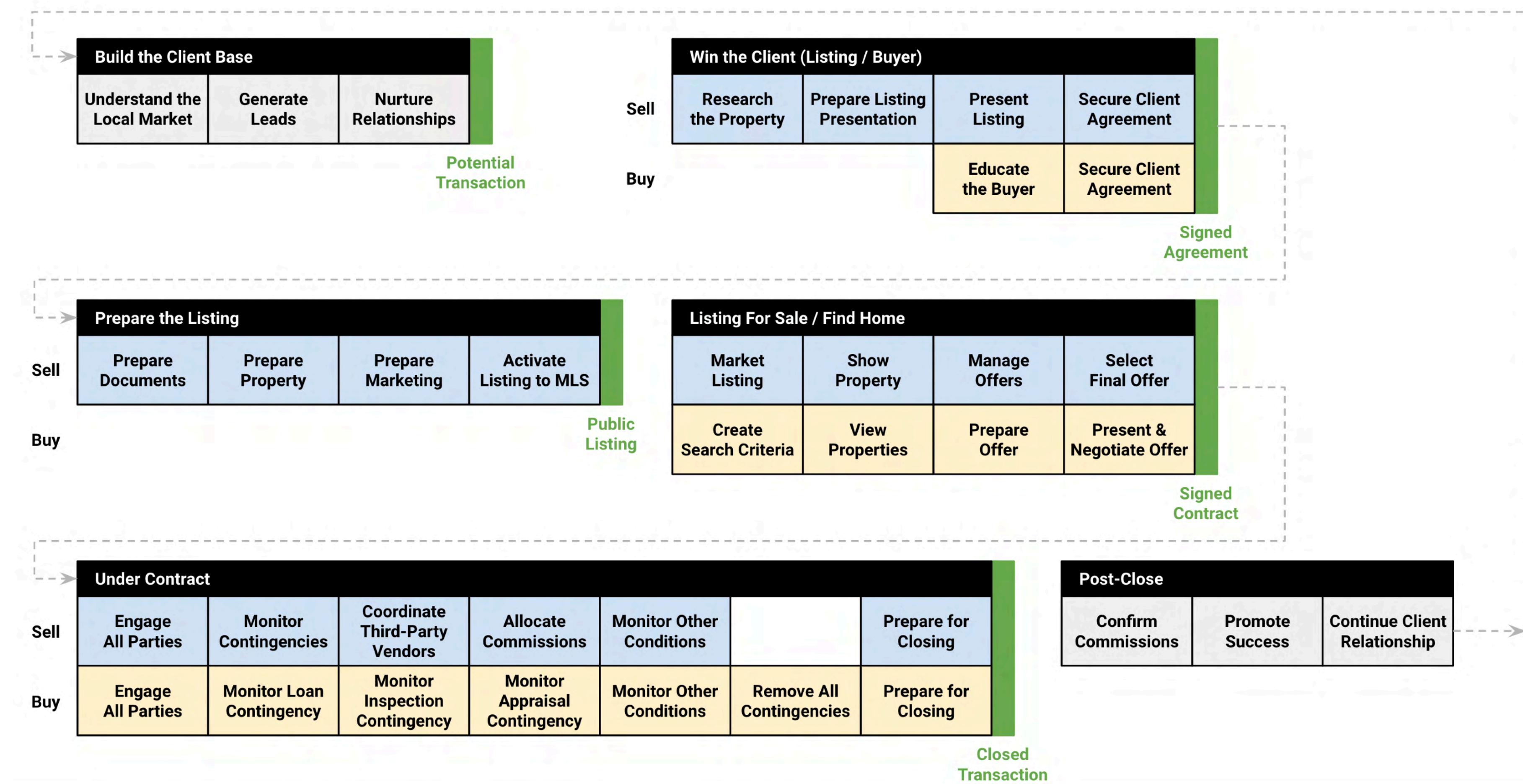
# Synthesis

Dissected, overlaid, and collated prior research efforts to discover gaps, overlaps, and how they phrased jobs and tasks.



# Proposed List: Jobs to be Done

Distilled the prior research plus agent feedback into this simple list of agent jobs to be done.



# Proposed List: Tasks per Job

Documented the list of tasks per job, also iterated based on agent feedback.

SELL - Listing For Sale				BUY - Find Home			
Market Listing	Show Property	Manage Offers	Select Final Offer	Create Search Criteria	View Properties	Prepare Offer	Present & Negotiate Offer
Market to Compass Private Exclusive members	Coordinate availability and preparation with seller	Review and organize received offers	Discuss final decision with seller	Learn the buyer's preferences (location, schools, attributes, etc)	Learn the buyer's availability	Review market conditions	Alert listing agent of offer
Share listing with network	Prepare materials for open house	Communicate offers to seller	Collect seller signatures on offer (or counter offer) and send to buyer	Schedule a home tour	Coordinate with listing agents	Create competitive analysis	Submit offer, cover letter & buyer finance qualifications
Market listing to team members	List open house on the MLS	Strategize with seller whether to accept, reject, or counter each offer	Receive executed offer package from buyer agent (counter)	Obtain property access information	Create tour itinerary	Discuss price and terms with buyers	Sign seller disclosures
Run advertising campaign (social media, digital ads, regional ads)	Coordinate and host broker tours	Inform buyer agent that you're making a counter offer	Provide current market estimates for their preferences	Tour listings	Reach out to listing agent for any conditions and to establish connection	Review and respond to counter offer (if receive)	Review and respond to counter offer (if receive)
Monitor buyer interest levels	Host (or delegate) open house(s)	Complete counter offer forms	Update MLS status	Capture and review buyer's feedback	Review seller disclosures	Receive & review seller executed agreement	Attach completed offer documents to compliance checklist
Strategize with client to adjust pricing or marketing strategy, if needed		Review counter offer package with seller and collect signatures	Receive feedback from client and adjust search criteria as needed	Contact listing agent for additional information	Draft offer		
		Submit counter offer to buyer agent and confirm receipt	Curate initial listings to showcase	Learn if buyer wants to submit an offer	Draft cover letter		
			Set up automated searches	Share newly matching listings	Review offer with buyer and obtain signatures		

# Proposed Data Model

Developed a data model of information that they needed to capture about each job and task.

**Identity.** General metadata about this job.

- **job-ID.** Each job will need a unique identifier at a system level.
- **job-name.** Short 2-4 word phrase, starting with an action verb, written from an agent's perspective. For example: "Generate Leads"
- **job-desc.** One to two sentences that further describe this job.

**Ownership.** Who internally is responsible for this job?

- **flow-team.** The team responsible for improving this task for agents, clients and staff.
- **flow-contact.** Primary point of contact from Flow Owner to support data collection.

**Flow.** Where does this job sit in the flow?

- **job-side.** Sell side, buy side, or both?
- **job-stage.** A higher-level phase of the process where this job happens, such as Prospect, Nurture, Prepare listing, In Contract, Closed.
- **job-previous.** Job(s) required to be completed in order to start.
- **job-next.** Job(s) blocked until this job is complete.
- **job-inputs.** List of items that are required to start this job.  
(Need to work out how to handle regional differences here.)
- **job-outputs.** List of items that this job produces.  
(Need to work out how to handle regional differences here.)

**Current State.** How are we delivering right now?

- **coverage.** (high/medium/low/none) How much does Compass already cover this job?
- **mobile coverage.** (high/medium/low/none) How well do we support this job on mobile today?
- **web coverage.** (high/medium/low/none) What functionality do we have available today (if any)? How well do we feel that technology is meeting the needs of transaction participants? If functionality is forthcoming, what is the timeline for delivery?
- **service desk.** (high/medium/low/none) How extensive do we offer help with our paid services?
- **platform engagement.** (high/medium/low/none) How often are agents using the Compass platform as part of their workflow for this item

**Research.** What have we learned about this job?

- **roles.** (multiple) List of roles typically involved in achieving this job. Buyer agent, seller, agent, transaction coordinator, buyer, seller, mortgage broker. (Another deep complicated section, likely needing its own data model. For now, a list of sample roles is enough.)
- **insights.** (multiple) Things we've learned about this job.
- **opportunities.** (multiple) Product concepts we have to improve this job.
- **documents.** (multiple) Links to documents about this job.
- **frequency.** (high/medium/low) How often do agents do this job? (likely more task than job)
- **complexity.** (high/medium/low) How difficult is this job to perform?
- **impact.** (high/medium/low) How much would improving this impact an agent's workflow?
- **entitlement.** (high/medium/low) How much do agents expect Compass to provide this (vs readily-available solutions like calendaring)
- **mood.** (high/medium/low) Can we capture an agent's mood about this job?
- **primary competing process.** What are the most common ways agents complete this task today and what are the primary general purpose and real-estate specific tools used to assist in the completion of this task?

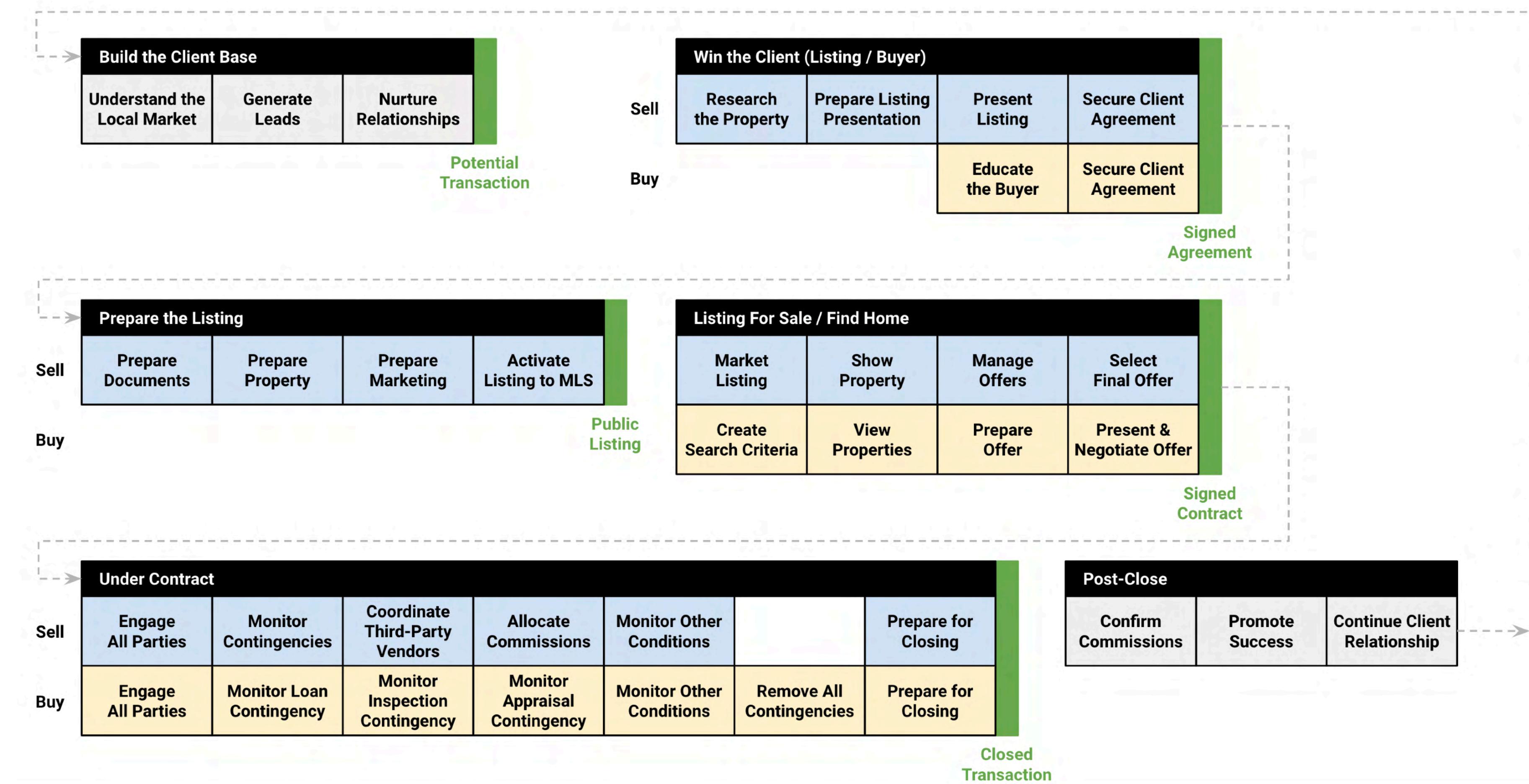
**Opportunity**

- **potential value-add.** (high/medium/low) How strong is our conviction that technology can positively impact this task? Do we have specific ideas on how this might be done?
- **effort.** (high/medium/low) What is the relative size of the development effort? Is there legacy tech debt to overcome?
- **platform integration.** (high/medium/low) Would improving this job also contribute to a platform effort, improving other jobs as a side benefit?
- **third-party integration ease.** (high/medium/low) How effectively could we provide coverage of this task more via a third-party integration? Are there any third-party tools we should prioritize integrating due to their popularity? (e.g. DocuSign)
- **mobile.** (high/medium/low) Where should this task fall in our priority list

# Outcome: A Shared Language

They never had had a simple list like this.

Several leaders said this framework should be used in annual planning, performance monitoring, and new agent training.



# Final Thoughts

# Overall Theme

**If you want to create coherence,  
you have to act coherently.**

The organization is the system.  
A disjointed organization will  
output a disjointed experience.

**For me, information architecture is  
much more than just the product.**

Focusing on a common internal  
language and helping people visualizing  
the larger system helps to unlock the  
collaboration necessary to deliver a  
coherent experience to customers.

**Thank you!**