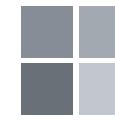




Jim Lamb

Full-stack Designer



Microsoft



Azure DevOps



Track-It!



Norton
by Symantec

I'm a user experience designer, product strategist, and information architect.

I help teams create products and experiences that solve real user problems.

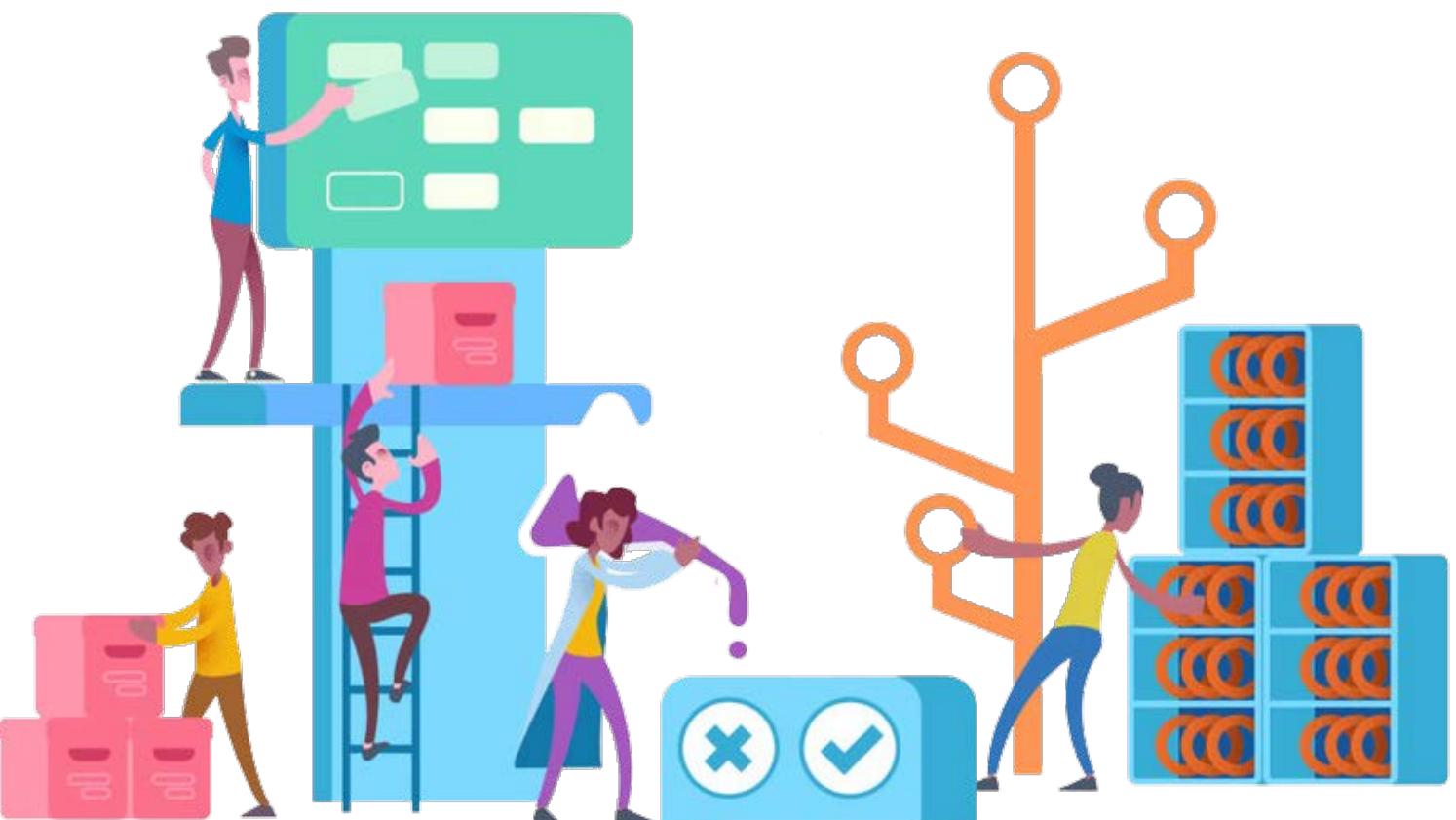
I use flow mapping, storyboarding, and prototyping to inform and guide the product design process.



About Azure DevOps

Microsoft's end-to-end DevOps toolchain for planning, developing and deploying software with over a million long-term engaged users.

As the primary designer for Azure Pipelines, I worked with a team of eleven designers and one researcher supporting over 300 technical staff.



The Challenge

- ✓ Simplify adoption of Azure Pipelines for GitHub users.
- ✓ Provide a streamlined authentication experience across the two services.
- ✓ Support configuration-as-code.

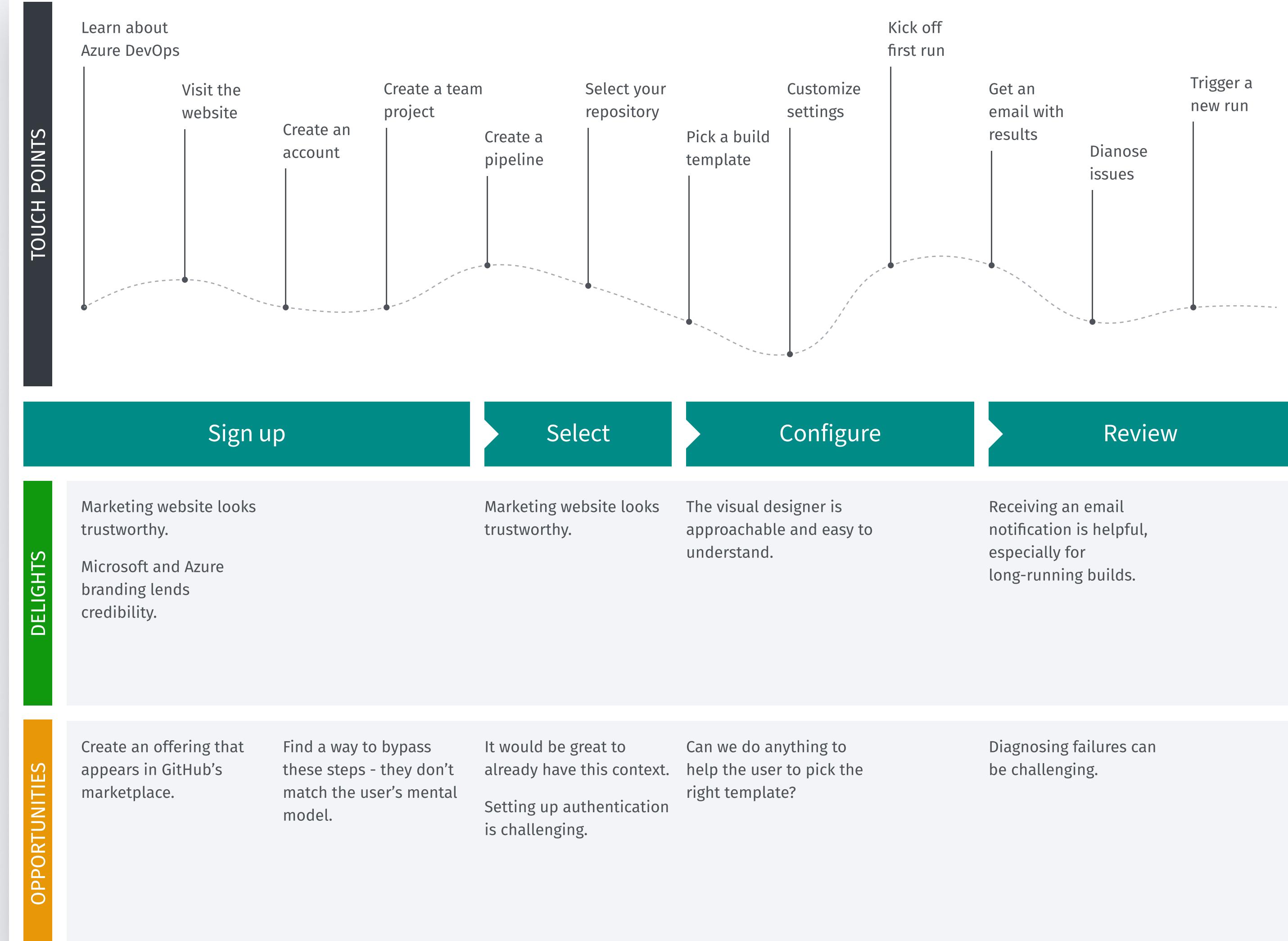
My Role

I was the lead designer for this project, supporting a team of about 12 engineers with one program manager over six months.

I also worked closely with the design system team as we defined new patterns and components to support this experience.

Journey map

I mapped out the existing experience for GitHub users getting started with Azure Pipelines. There were a number of areas for improvement.





Connect to host service

Users who install the GitHub marketplace app get to skip this view.

The list of supported version control services continued to grow over time.



Connect

Select

Configure

Review

New pipeline

Where is your code?



Azure Repos

Unlimited free private repos



GitHub

Home to the world's largest community of developers



GitHub Enterprise

Self-hosted version of GitHub



Bitbucket

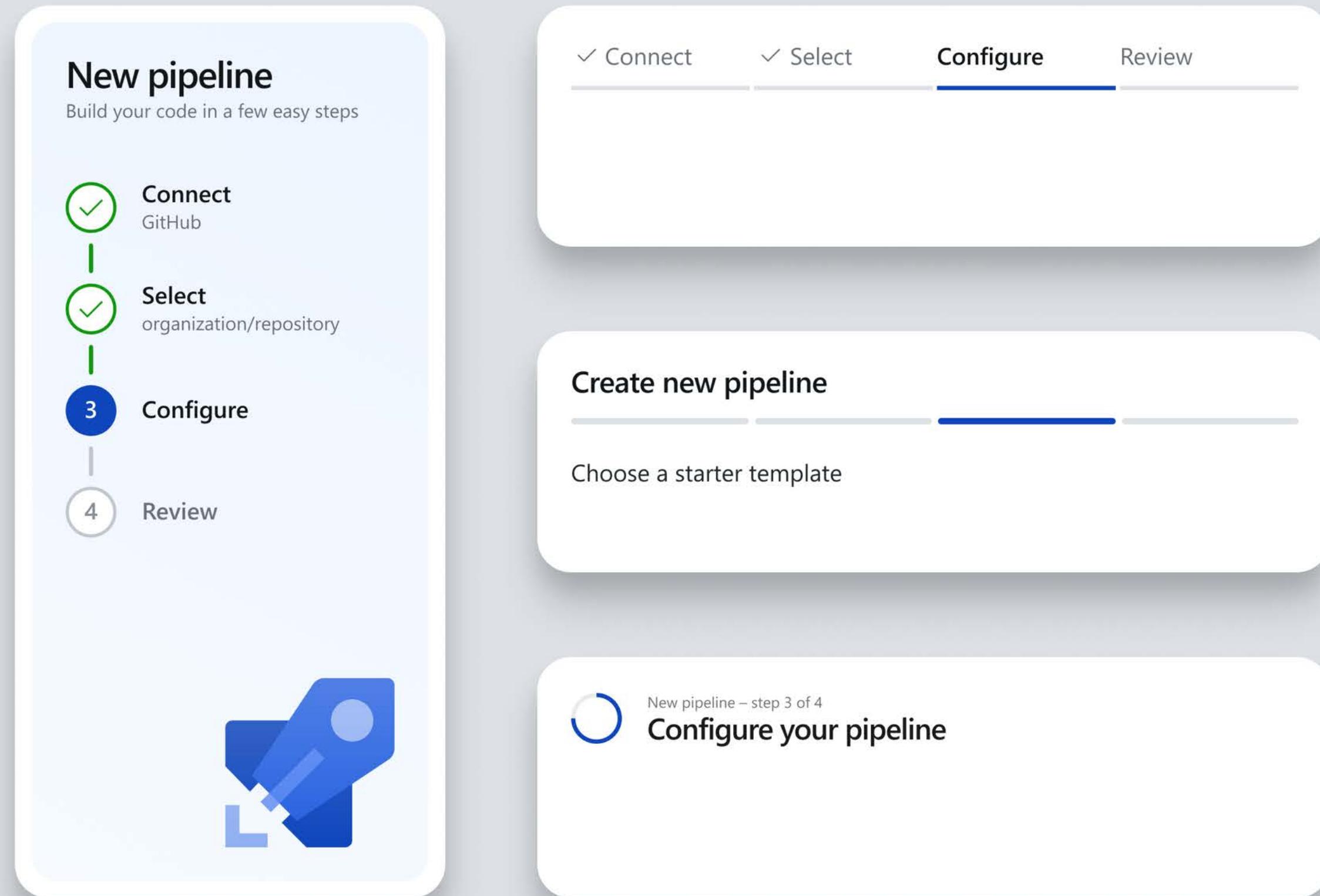
Hosted by Atlassian

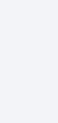
[Use the visual designer](#) to create a pipeline without YAML.



Conveying progress

I did several explorations for the “stepper” component. We decided to keep it pretty minimal to reduce the chances of having it clash with future design system elements.





Authorize access to your repository

Users who install the GitHub marketplace app get to skip this view.

For other users, support alternative authentication methods was still required.



✓ Connect

Select

Configure

Review

New pipeline

Select a repository

Use an existing security context to access your GitHub repositories



cburton

OAuth



Contoso

GitHub App



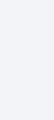
cballinger

Personal access token

Or, authorize a new security context

Authorize





Select your repository

By default, the list includes repositories that you own or collaborate on, sorted by how recently you accessed them.

Annotations call out forks and private repositories for easy differentiation.

For some users, this list can include thousands of repositories, so we decided to support filtering.



✓ Connect

Select

Configure

Review

New pipeline

Select a repository

Filter by keywords

My repositories



user/repository

2h ago



contoso/node-express-realworld-example-app

fork

private

Yesterday



contoso/tin-octopus

private

Jul 17

ⓘ Showing the most recently used repositories where you are a collaborator.

If you can't find a repository, make sure you [provide access](#).

You may also select a specific [connection](#).



Analyze source code & suggest a starter template

Templates are suggested based on the detection of key files/folders within the repository.

Engineering was skeptical that this analysis would work reliably, but some quick prototyping demonstrated that it would be relatively straightforward.



5



✓ Connect

✓ Select

Configure

Review

New pipeline

Configure your pipeline



Node.js with Gulp recommended

Build a Node.js application using the Gulp task runner.



Node.js with Grunt

Build a Node.js application using the Grunt task runner.



Starter pipeline

Start with a minimal pipeline you can flesh out to build your code.



Bring your own YAML

Select the branch with an existing Azure Pipelines YAML file



Bring your own Docker file

Select the branch with an existing Docker file

Show more





Generate YAML using the template

While the system can generate the YAML for the pipeline, the user will often want to customize it.

The text editor is full-featured. It uses the Monaco component (used by Visual Studio Code) and includes an assistive panel.

✓ Connect

✓ Select

✓ Configure

Review

Review your YAML for pipeline

repository-name

Variables

Save and run

repository-name / azure-pipelines.yml

Show assistant

```
1 # Node.js with gulp
2 # Build a Node.js application using the gulp task runner.
3 # https://aka.ms/yaml
4
5 queue: 'Hosted VS2017'
6
7 variables:
8 # system.debug: 'true'
9
10 steps:
11 - task: Npm@1
12   displayName: 'npm install'
13
14 - task: Gulp@0
15   displayName: 'Run gulp'
16   inputs:
17     gulpFile: 'gulpfile.js'
18     targets: ''
```

Commit the changes to the repository

Completing the process requires committing the changes to the specified repository.

The panel provides reasonable defaults while providing some flexibility on the details of the commit.

✓ Connect ✓ Selected

Review your YAML for pipeline

repository-name

repository-name / azure-pipeline

```
1 # Node.js with gulp
2 # Build a Node.js application
3 # https://aka.ms/yaml
4
5 queue: 'Hosted VS2017'
6
7 variables:
8   system.debug: 'true'
9
10 steps:
11   - task: Npm@1
12     displayName: 'npm install'
13
14   - task: Gulp@0
15     displayName: 'Run gulp'
16     inputs:
17       gulpFile: 'gulpfile.'
18       targets: ''
```

Save and run pipeline

Pipeline name

repository-name

Commit message

Set up CI with Azure Pipelines

Created by Azure Pipelines

Changes (1)

.azure-pipelines.yml
Pipeline process

How to push

- Commit directly to the default branch
- Create a new branch for this commit

Cancel

Save and run

Outcomes

- 😊 Customer acquisition rates from GitHub increased dramatically.
- 😊 Added the stepper (progress) component to our nascent design system.
- 😊 Engineering was able to reuse the experience for all of the pipeline creation flows.

Lessons

Cross-service authentication is hard, but GitHub's new app model helped us simplify it dramatically.

Balance approachability for new users with flexibility for experienced ones.

When getting ahead of the design system, partner with avoid creating future design debt.

The Challenge

- ✓ Integrate the build and release areas of Azure DevOps into a single pipeline experience.
- ✓ Support configuration as code.
- ✓ Preserve support for complex scenarios while simplifying the experience for most users.

My Role

I worked with two other designers (who were new to the team) and two researchers for over nine months.

I was the sole designer for the second iteration of this design. The goal was to address the user feedback from the preview of the initial design.

Existing UI

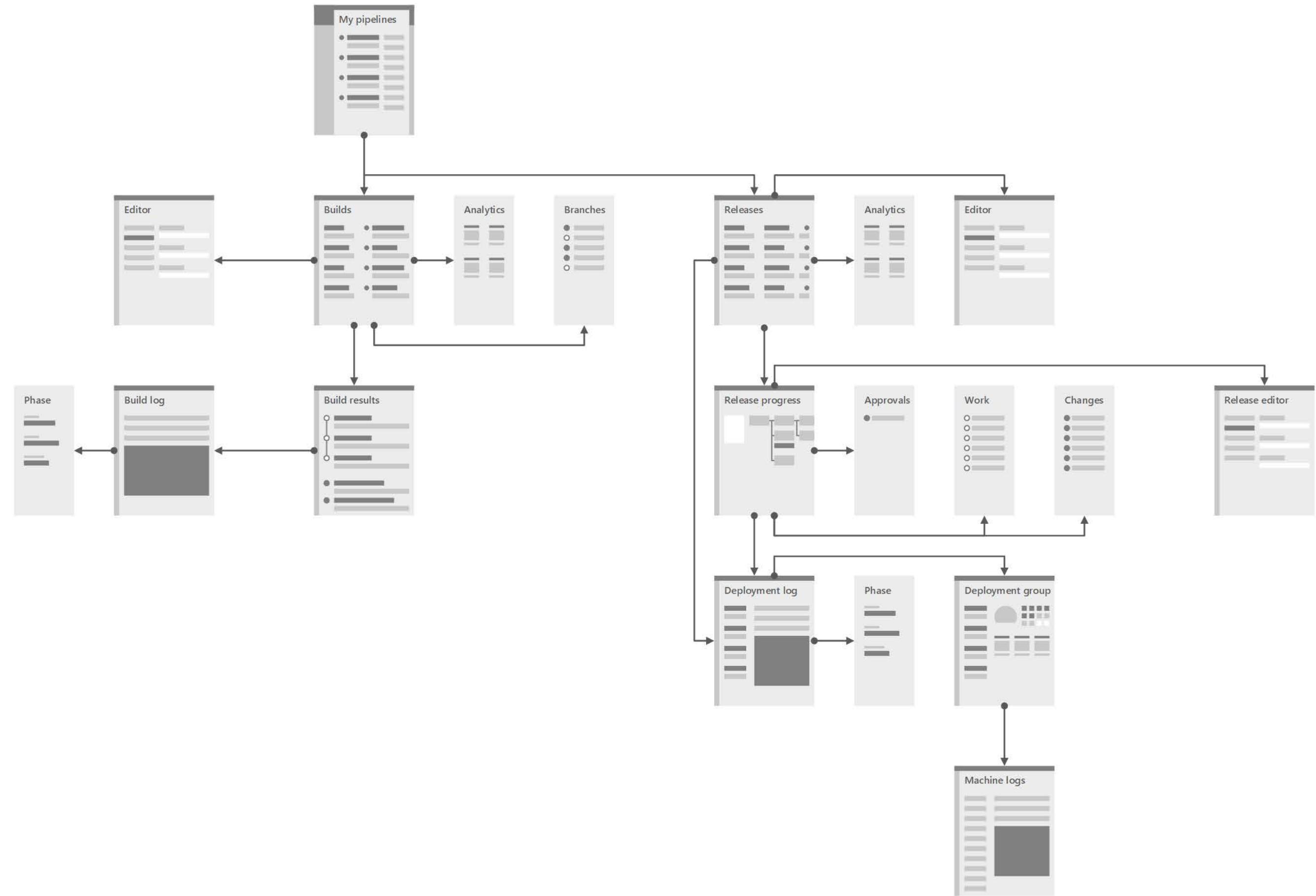
The views for automated builds were quite complex and could be challenging to navigate. This particular screenshot represents a “worst-case” scenario, but you can see how it can be overwhelming.

The screenshot shows the Azure DevOps Pipelines Builds page. The left sidebar has a 'One' project selected. Under 'Pipelines', 'Builds' is highlighted. The main area displays a list of build history items for the pipeline 'Portal-ThresholdTests-Dev'. The columns in the list are: Commit, Build #, Branch, Queued, and Duration. The list contains 20 entries, each representing a manual build for CDP Buddy. The builds were queued on 2019-02-26 and took between 1:12.863 and 2:30.539 minutes to complete.

Commit	Build #	Branch	Queued	Duration
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20432431	# dev	2019-02-26 17:45	
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20432376	# dev	2019-02-26 17:43	
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20432278	# dev	2019-02-26 17:39	2:30.539
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20432129	# dev	2019-02-26 17:33	2:12.863
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20431390	# dev	2019-02-26 17:06	1:52.796
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20430766	# dev	2019-02-26 16:44	2:12.512
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20430717	# dev	2019-02-26 16:42	1:40.538
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20430640	# dev	2019-02-26 16:39	2:12.099
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20429990	# dev	2019-02-26 16:13	1:57.505
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20429780	# dev	2019-02-26 16:06	2:12.092
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20429459	# dev	2019-02-26 15:59	2:12.355
Merged PR 1454980: #4043498 [AzBlade] Uri encode extension version...	20429398	# dev	2019-02-26 15:56	1:25.367
[Automated] Localization hand-back from build 5.0.0.2281	20429112	# dev	2019-02-26 15:46	2:16.012
Merged PR 1443340: #3861382 Focus order is incorrect on exiting from...	20428689	# dev	2019-02-26 15:31	1:52.436
Merged PR 1443340: #3861382 Focus order is incorrect on exiting from...	20428666	# dev	2019-02-26 15:30	2:32.045
Merged PR 1451920: #4027468 Flight simple batch at 50% in mpac	20427885	# dev	2019-02-26 15:00	2:24.452
Merged PR 1453993: #3845741 Expand "no 'ready' message" telemetry...	20427626	# dev	2019-02-26 14:50	1:50.911
Merged PR 1446672: #4004351 [Performance] Update BladeDefinitionP...	20427603	# dev	2019-02-26 14:50	1:49.094

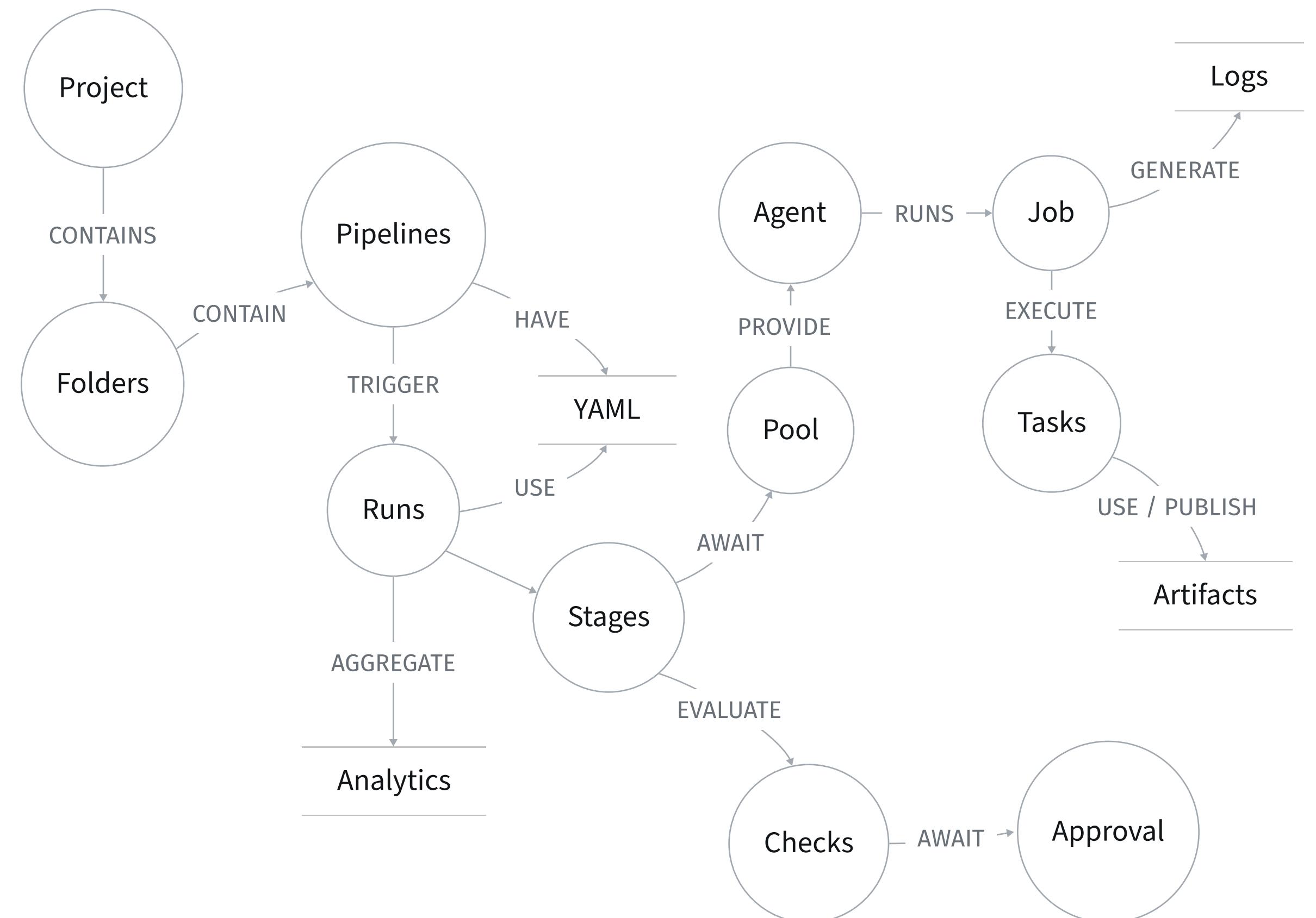
Wireflow

I created a wireflow
([https://www.nngroup.com
/articles/wireflows/](https://www.nngroup.com/articles/wireflows/)) to
map out how the various
views related to one
another.



Information architecture

I created this more conceptual diagram to map out the various objects within the system and their relationships.



New

...

Pipelines

Early exploration

In this exploration, I pushed the limits of how much information I could represent in the pipeline diagram.

I put a heavy visual emphasis on the most recently completed stage and de-emphasized stages that finished successfully.

I used vertical delimiters to represent stages running in parallel.

Mine & active All pipelines



Name	Progress	Updated
enterprise-distributed-service #482 · Added testing for get_service_instance_stats	! Test	19m ago
microservice-architecture-app #137 · Update user service	Teardown Canary	2h ago
mobile-ios-app #4000 · Bug 1470171 - Photon style tab button	Submit	4h ago
node-package #365 · Add a request body validator	Publish	5h ago
parallel-stages #137 · Update user service	! Load UAT	6h ago
simple-web-app #283 · Add extra max length and required constrain	Staging	Yesterday at 2:30pm
too-many-stages #94 · Fix bug with object address	! US-SE-	Monday at 10:38am
yet-another-node-package #94 · Fix bug with object address	Build	Monday at 8:12am



Listing

The initial preview only supported single stage pipelines, so we went with a simpler visualization of pipeline status.

Each row has a default click target as well as sub-targets for key elements and contextual commanding.

Pipelines

Recently run All pipelines All runs

Recently run pipelines Filter

Name	Last run	
multiple-stages	#482 • Added testing for get_service_instance_stats PR validation master	1h ago 23m 8s
simple-microservice	#137 • Update user service Release master	Yesterday 5m 2s
mobile-ios-app	#32 • Update user service Scheduled master	2h ago 33m 1s
node-package	#385 • Add a request body validator PR validation test	4h ago 4m 17s
parallel-stages	#792 • Clean up notifications styling Manually triggered develop	6h ago 2m 8s
simple-web-app	#283 • Add extra padding on cells PR validation feature-123	Monday 12 49m 52s

Detail, runs

Selecting a pipeline transitions to this view. A simplified version of the pipeline list appears on the left. The right-hand view shows the runs for the selected pipeline.

The transition helps keep the user oriented and makes it easy to check up on the status of multiple pipelines.

← Pipelines

- Recent
 - enterprise-distributed-service
 - microservice-architecture
 - mobile-ios-app**
 - node-package
 - parallel-stages
 - simple-web-app

mobile-ios-app

Runs Branches Analytics

mobile-ios-app runs		Filter
Description	Stages	
#8 • Added testing for get_service_instance_...	- - -	just now 23m 8s
#7 • Merge pull request 401701 from users/f...	- - - -	2h ago 5m 2s
#6 • Update user service	- - - -	6h ago 33m 1s
#5 • Add a request body validator	- - - -	7h ago 4m 17s
#4 • Cleanup notifications styling	- - -	12h ago 1h 14m 8s
#3 • Add extra padding on cells	- - - -	Yesterday 49m 2s
#2 • Make a property getter callable	- - - -	May 10 42m 17s
#1 • Update dashboard based on the new sc...	- - - -	May 4 28m 3s

Analytics

The analytics tab shows some key performance indicators for the selected pipeline.

The note at the bottom provides some analysis of the data beyond the graph.

← Pipelines

Recent

- enterprise-distributed-service
- microservice-architecture
- mobile-ios-app
- node-package
- parallel-stages
- simple-web-app

mobile-ios-app

Edit Run pipeline :

Runs Branches Analytics

Pipeline pass rate
88.8% ↓2.6%
of 2,237 runs

38.2% of failures occur in task Build app (Xcode)

Test pass rate
99.9%
of 4,291 tests

242 unique tests are causing 481 test failures

Pipeline run-time
12m 42s ↓4.6%
over 1,986 passing runs

31.4% of run-time taken by task Build app (Xcode)

This screenshot shows the 'Analytics' tab for the 'mobile-ios-app' pipeline. The top navigation bar includes 'Edit' and 'Run pipeline' buttons. The main area displays three key performance indicators: Pipeline pass rate (88.8% of 2,237 runs), Test pass rate (99.9% of 4,291 tests), and Pipeline run-time (12m 42s over 1,986 passing runs). Below each metric is a detailed note providing specific analysis. A sidebar on the left lists recent pipelines, and a bottom navigation bar includes 'Runs', 'Branches', and 'Analytics' tabs.



Error presentation

When a run fails, we want to make it as easy as possible to review and diagnose any errors or warnings.

#5 • Add a request body validator
multiple-stages

[Run new](#) [View artifacts](#) [⋮](#)

Summary Tests

Pull request 4583 by Colin Ballinger [View changes](#)

Repository and version	Time started and elapsed	Related	Tests and coverage
❖ AdventureWorks 513226	Today at 9:02 AM 6m 48s	1 user story 2 artifacts	100% passed 87.2% lines covered

Errors 3 Warnings 1

Stage • Job (agent/image) • First failed task

- ✖ Artifact drop not found for build 10600514. Please ensure you have published artifacts in any previous phases of the current build.
- ✖ Error CS0136: A local variable named 'requiredcount' cannot be declared in this scope because it would give a different meaning to 'requiredc...'
- ✖ Could not parse Jtokens from D:\v2.0\P1\work\7\s\Vssf\Web\extensions\vss-features\package-lock.json file.

Stages Jobs

```
graph LR; Build[Build] --- Test[Test]; Test --- Deploy[Deploy]
```

Stage	Jobs	Status
Build	1 job completed	6m 48s
Test	1 job skipped	
Deploy	1 job skipped	



Failure analysis

I explored the idea of having an assistive experience to help users diagnose problems. My goal was to reduce support costs and improve user satisfaction.

This feature didn't make it into the initial feature preview or the general availability release. Still, product management put it on the backlog, and engineering built it out a few months later.

The screenshot shows the Azure DevOps Pipeline details page for a failed merge PR. The pipeline name is "#AzureDevOps_merge_20191017.227" and it failed on AzureDevOps. The summary section shows the pull request (PR) 507297 by Colin Ballinger, which was started today at 2:07 PM and took 17m 49s. The error section details a specific error message: "Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output Stage (if more than one) • Job (Pool/Agent) • Task (line #)". The job section shows a single job named "PR".

Error analysis

Build / Build solution dirs.proj (MSBuild)

c1xx(0,0): Error C1250: Unable to load plug-in 'D:\v2.0\A2_work\26\tool\Mseng.Compliance.FxCop\content\Plugins\HRESULT.dll'.

C/C++ warnings

C1250: Unable to load plug-in.

The Code Analysis tool reports this warning when there is an internal error in the plugin, not in the code being analyzed.

Related commits

Add LiveSiteEventPage tests
e6424bef by Colin Ballinger today at 12:30 PM

Run environment

Variables

MSBuild task

Pipeline YAML history

About hosted agent image

View log



Logs

The logs view has to handle tasks in a range of states (pending, in progress, and completed). Each state has a unique set of attributes.

This view also has to scale to very, very large logs while being highly performant.

The screenshot shows the Azure DevOps Logs view for a pipeline named '#5 • Add a request body validator'. The left sidebar lists various log categories: MM, Metrics, Issues, CI/CD, Build, Test, Integration, Performance, Deploy, and More. The 'Build' category is selected, showing a list of steps:

- Errors (3):
 - Artifact drop not found for build 10600...
 - Error CS0136: A local variable named 'r...' (red)
 - Could not parse Jtokens from D:\v2.0\P... (red)
- Build:
 - Build services (14s):
 - Initialize job (<1 s) (green)
 - Checkout (13s) (green)
 - Build packages (red)
 - Publish artifacts (green)
 - Finalize job (green)
 - Test:
 - Integration tests (14s):
 - Performance tests (14s):
 - Deploy:
 - Deploy to web (14s):
 - More (14s)

The 'Build packages' step is highlighted with a red error icon. The log output for this step is as follows:

```
* Analysis was not performed; at least one valid rules assembly and one valid target file must be specified.  
* 1 total analysis engine exceptions.  
410>GenWebConfig:  
    Skipping target "GenWebConfig" because it has no outputs.  
47 ✘ CA0055 : Could not identify platform for 'D:\v2.0\A2\_work\22\bin\Debug.AnyCPU\Vssf.Client\netstandard2.0\Microsoft.VisualStudio.Services.Common.dll'.   
121>MSBUILD : error : CA0055 : Could not identify platform for 'D:\v2.0\A2\_work\22\bin\Debug.AnyCPU\Vssf.Client\netstandard2.0\Services.Common.dll'. [D:\v2.0\A2\_work\22\s\Vssf\Client\Common\MS.VS.Services.Common.NetStandard.csproj]  
49 ✘ CA0052 : No targets were selected.  
50     121>MSBUILD : error : CA0052 : No targets were selected.  
[D:\v2.0\A2\_work\22\s\Vssf\Client\Common\MS.VS.Services.Common.NetStandard.csproj]  
Code Analysis Complete -- 2 error(s), 0 warning(s)  
51     1>Project "D:\v2.0\A2\_work\22\s\dirs.proj" (1) is building "D:\v2.0\A2\_work\22\s\CodeReview\dirs.proj" (492) on node 1 (BuildFlatAndAggregate target(s)).  
52     492>Project "D:\v2.0\A2\_work\22\s\CodeReview\dirs.proj" (492) is building "D:\v2.0\A2\_work\22\s\CodeReview\Sdk\dirs.proj" (496) on node 1 (BuildFlatAndAggregate target(s)).  
53     496>Project "D:\v2.0\A2\_work\22\s\CodeReview\Sdk\dirs.proj" (496) is building "D:\v2.0\A2\_work\22\s\CodeReview\Sdk\Discussion\dirs.proj" (497) on node 1 (BuildFlatAndAggregate target(s)).  
54     497>Project "D:\v2.0\A2\_work\22\s\CodeReview\Sdk\Discussion\dirs.proj" (497) is building "D:\v2.0\A2\_work\22\s\CodeReview\Sdk\Discussion\Plugins\MS.VS.Services.CodeReview.Discussion.Server.Plugins.csproj" (543 ) on node 11 (BuildFlatAndAggregate target(s)).  
55     543>PrepareForBuild:  
        Creating directory "D:\v2.0\A2\_work\22\bin\Debug.AnyCPU\CodeReview.Sdk\".  
        Creating directory "D:\v2.0\A2\_work\22\obj\Debug.AnyCPU\CodeReview.Sdk\MS.VS.Services.CodeReview.Discussion.Server.Plugins.csproj_9cf6d89\".  
56     410>CopyFilesToOutputDirectory:  
        Creating hard link to copy
```

Mobile views

We also designed mobile (responsive) versions of commonly used views.

The image displays three screenshots of the Azure DevOps mobile application interface, illustrating its responsive design for mobile devices.

- Pipelines View:** Shows the Pipelines page for the "eShopOnWeb" project. It lists four recent pipeline runs:
 - #20200820.1 Refactor deployment stage (Manually triggered for master)
 - #20200318.17 Refactor deployment stage (Individual CI for master)
 - #20200318.16 Refactoring build stage to co... (Individual CI for master)
 - #20200318.13 Update azure-pipelines.yml f... (Individual CI for master)A "Run" button is visible in the top right corner.
- Job Details View:** Shows the details for a specific pipeline run (#20200820.1). It includes:
 - Summary tab (selected), showing the run was manually triggered by Jim Lamb.
 - Environment and Code Coverage tabs.
 - Stages and Jobs sections, listing the steps of the pipeline.
- Job History View:** Shows the detailed history of the pipeline run #20200820.1. It lists the individual jobs and their durations:
 - Build image: Build (1m 57s)
 - Development: Publish image (13s)
 - Post-job: Checkout eShopOnWeb... (less than 1 second)
 - Finalize Job (less than 1 second)
 - Publish image (13s)
 - Report build status (less than 1 second)

Outcomes

- :(Initially high opt-out rates due to navigational changes.
- : Smiley Dramatically reduced opt-out rate during feature preview by addressing user feedback.
- : Smiley Improved adoption of single pipeline solution over legacy features.

Lessons

Simpler isn't always better.
Design for scale *and* performance.
Don't over-index on features users haven't adopted yet.

Other recent projects

- ✓ Improvements to the logs view for pipeline runs
- ✓ Onboarding to Azure's Kubernetes Service
- ✓ Notifications for Teams & Slack
- ✓ Pipeline YAML editor with integrated assistance
- ✓ Accessibility Insights integration with GitHub actions

Design system contributions

- Navigating multiple pages within a panel/fly-out
- Empty states
- Master/detail views
- Accordion, collapsible cards
- Segmented control (choice chips)
- Multi-pane workspaces

Thank you!

Any questions?

You can find me at:

-  <https://www.linkedin.com/in/jlamb>
-  jimlamb@hey.com