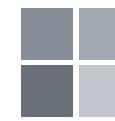




Jaime Lamb

Full-stack Designer



Microsoft



Azure DevOps



Track-It!



Norton
by Symantec

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

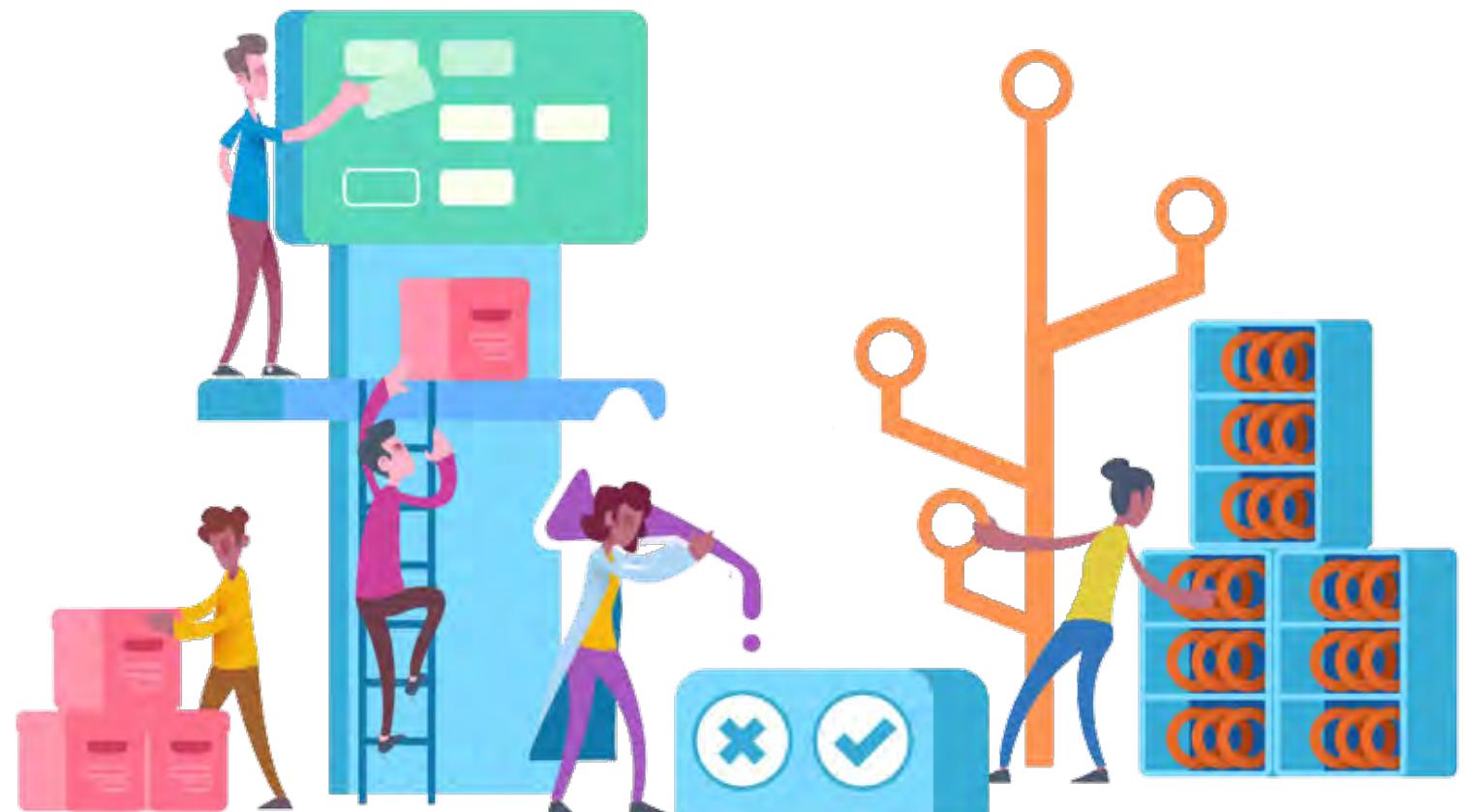
I've managed all aspects of design projects through initial strategy and ideation, sketches and wireframes, pixel-perfect comps, and working with engineers to make sure the implementation lives up to the design.



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.



Project #1

GitHub Marketplace App

Make it quick and painless for developers on GitHub to use Azure Pipelines to set up continuous integration and deployment.

The screenshot shows the Azure Pipelines app page on the GitHub Marketplace. At the top, there are navigation links: Marketplace, Apps, and Azure Pipelines. Below the header, the app icon (a blue gear-like shape) and the title "Azure Pipelines" are displayed. A message indicates that the user has already purchased the app. There are two main buttons: "Set up a new plan" (green) and "Edit your plan" (white). A "Configure access" link is also present. The app's description highlights its ability to "Continuously build, test, and deploy to any platform and cloud" using "cloud-hosted pipelines for Linux, macOS, and Windows". It mentions 10 free parallel jobs and unlimited minutes for open source projects. A "Read more..." link is available. On the left side, there are sections for "Categories" (Continuous integration, Deployment, GitHub Enterprise, Checks API, Free), "Supported languages" (Dockerfile, Go, Java, and 7 other languages supported), and "Developer" (AzurePipelines logo). On the right, a large blue box titled "Linux, macOS, and Windows agents" shows pipeline stages: Build Linux (0 succeeded), Unit tests (All succeeded), Distribute, Build macOS (1 succeeded), Functional tests (0 in progress), and Build Windows (1 succeeded).

Marketplace Apps Azure Pipelines

Application

Azure Pipelines

(i) You have already purchased this app on GitHub Marketplace.

Set up a new plan Edit your plan

Configure access

Verified by GitHub

GitHub confirms that this app meets the requirements for verification.

Categories

Continuous integration Deployment GitHub Enterprise Checks API Free

Supported languages

Dockerfile, Go, Java and 7 other languages supported

Developer

AzurePipelines

Linux, macOS, and Windows agents

Simplifying managing hardware and VMs by using Microsoft cloud-hosted agents. Get full CI/CD pipeline support for every major platform and tool.

Build Linux 0 succeeded

Unit tests All succeeded

Distribute

Build macOS 1 succeeded

Functional tests 0 in progress

Build Windows 1 succeeded

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.

Existing UI

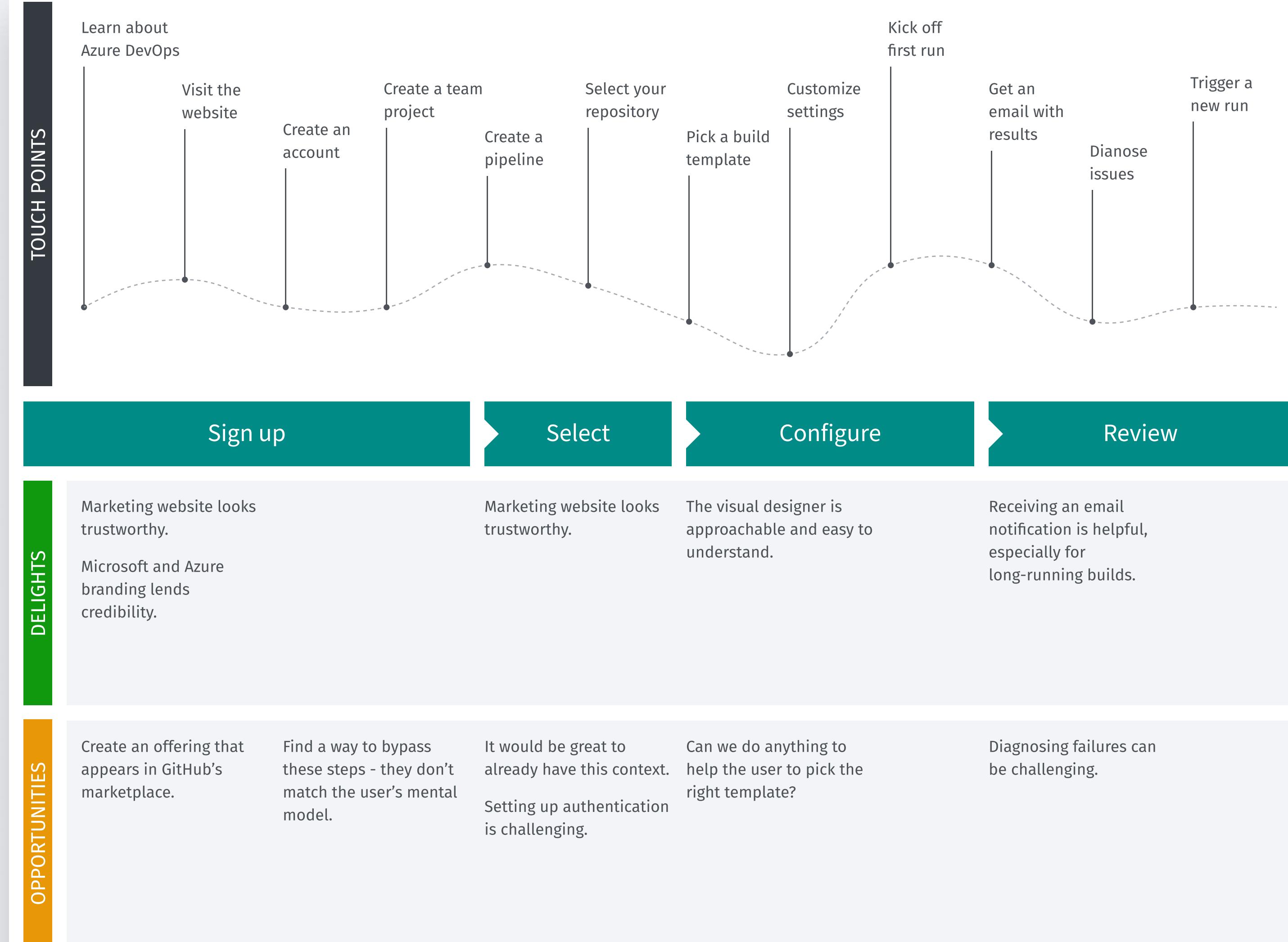
While the existing experience made it possible to build a GitHub hosted repository, it was a convoluted process requiring a deep understanding of the service.

The screenshot illustrates the existing user interface for integrating GitHub into Azure DevOps, highlighting several steps and configurations:

- Add new GitHub service connection:** A modal dialog titled "Add new GitHub service connection" shows the "Grant authorization" option selected. A "Authorize" button is visible. The connection name is "GitHub Endpoint".
- Endpoint: GitHub Endpoint:** The "Endpoints" page shows a "New Service Endpoint" dropdown and a search bar. A "github" entry is expanded, showing the "GitHub Endpoint" service connection.
- INFORMATION:** The "GitHub Endpoint" service connection details are shown, including "Type: Github", "Created by", and "Connecting to service using OAuth".
- ACTIONS:** Options to "Update service configuration" and "Disconnect" are available.
- Build Pipeline Editor:** The "PartsUnlimited_GitHub" pipeline is displayed. The "Tasks" tab is active, showing a "Get sources" task (step 1) configured for "VKarthik91/PartsUnlimitedE2E" and "master".
- Source Providers:** The "Build and Release" ribbon shows icons for "This project", "GitHub" (selected, step 2), "GitHub Enterprise", "Bitbucket® Cloud", and "External Git".
- Authorization:** A green status bar at the bottom indicates "Authorized using connection: GitHub Endpoint" (step 3).
- Repository:** The "Repository" field is set to "VKarthik91/PartsUnlimitedE2E" (step 4).
- Branch:** The "Branch" field is set to "master" (step 5).

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.

A single click selects and continues to the next step.

MM



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

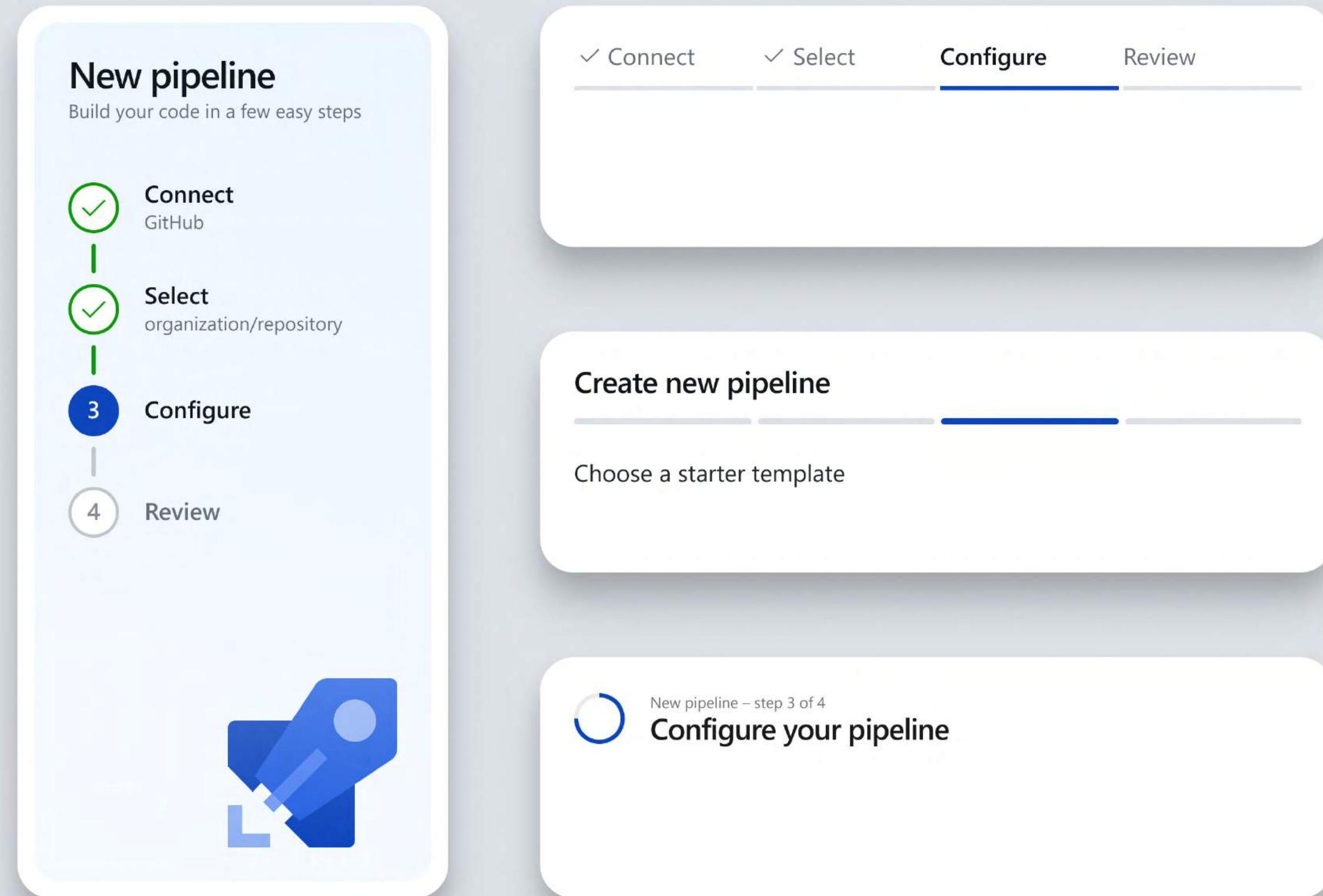
**Bitbucket**

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

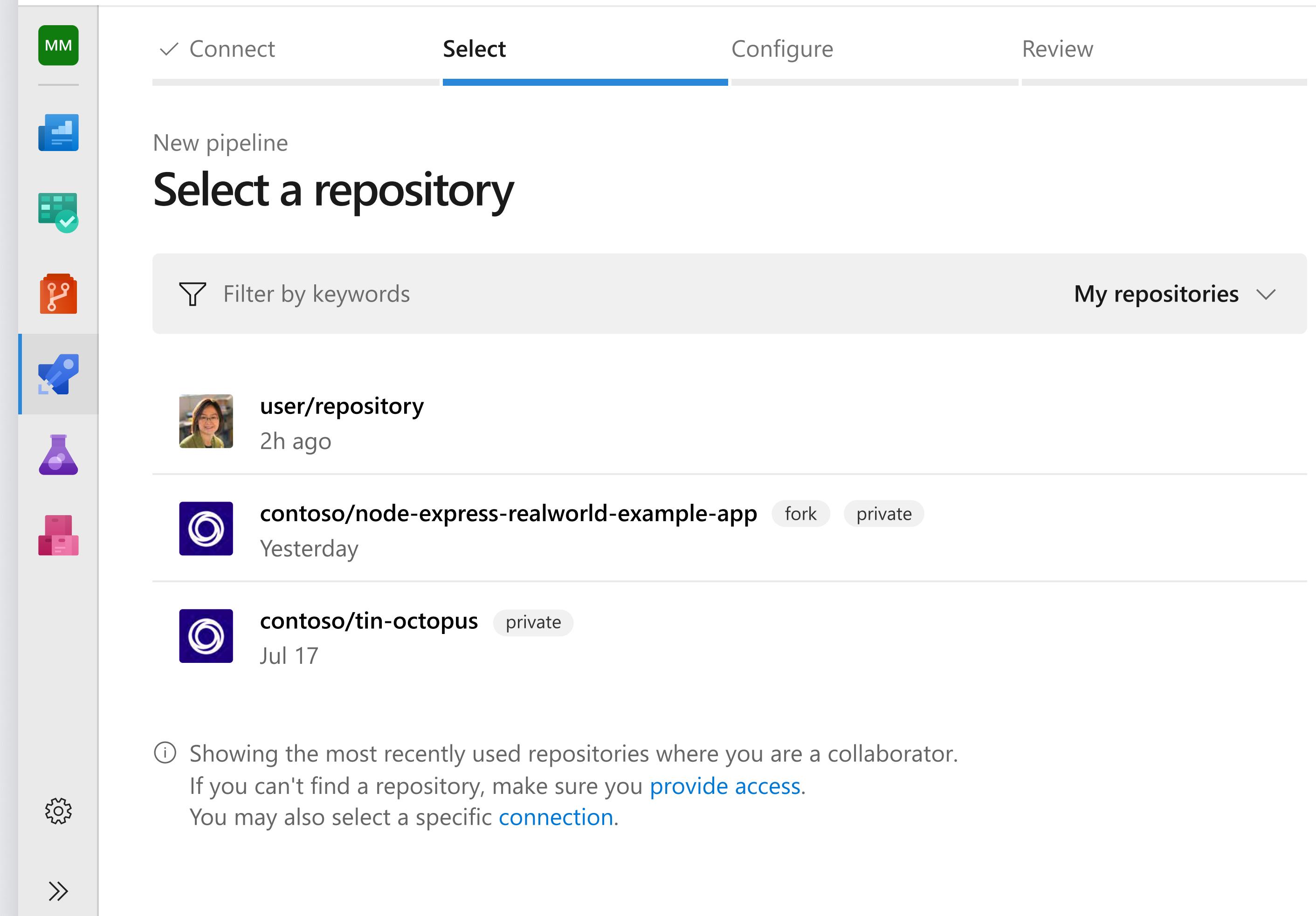
Authorize

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.



New pipeline

Select a repository

Filter by keywords My repositories ▾

Repository	Created
user/repository	2h ago
contoso/node-express-realworld-example-app	Yesterday
contoso/tin-octopus	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.



✓ 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



Show assistant

repository-name / azure-pipelines.yml

```
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.

Organization / Project / New build

✓ Connect ✓ Select

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.

Project #2

Single Pipeline

Users want to build, package, test, and deploy their software with a single, automated pipeline.

The screenshot shows a web-based CI/CD pipeline interface. On the left, there's a sidebar with icons for code, database, and deployment. The main area is titled 'Pipelines' and shows a table of 'Recently run pipelines'. The table has columns for 'Name', 'Last run', and two small icons. The pipelines listed are:

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 2h ago (5m 2s)
mobile-ios-app	#32 • Update user service Scheduled develop 6h ago (33m 1s)
node-package	#385 • Add a request body validator PR validation test Yesterday (4m 17s)
parallel-stages	#792 • Clean up notifications styling Manually triggered develop Monday (2m 8s)
simple-web-app	#283 • Add extra padding on cells PR validation Features/feature-123 Aug 15 (49m 52s)
react-front-end	#607 • Update super-linter.yml PR validation dev Aug 12 (3m 31s)
dotnet-core-api	#291 • Correct an allOf related issue in swagger PR validation features/new-api-version Aug 11 (8m 4s)
k8s-hosted-microservices	#210 • Adding test recording with new param Release releases/2020.09.09.4 Aug 9 (12m 52s)
progressive-web-app	#822 • Bump grunt from 1.1.0 to 1.3.0 in /website PR validation develop Aug 8 (3m 14s)
flutter-mobile-app	#396 • Added new AssignedHostGroup field in the objects Manually triggered hotfixes/hotfix-3871 Aug 5 (5m 44s)
mobile-android-app	#988 • Adjusted: Dark mode PR validation master Jul 27 (6m 18s)
ecommerce-portal	#515 • Fix build break Release releases/M381 Jun 18 (11m 18s)
content-management-system	#164 • Change to dynamically loading of icon images for WebSearch Scheduled master May 1 (8m 21s)

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

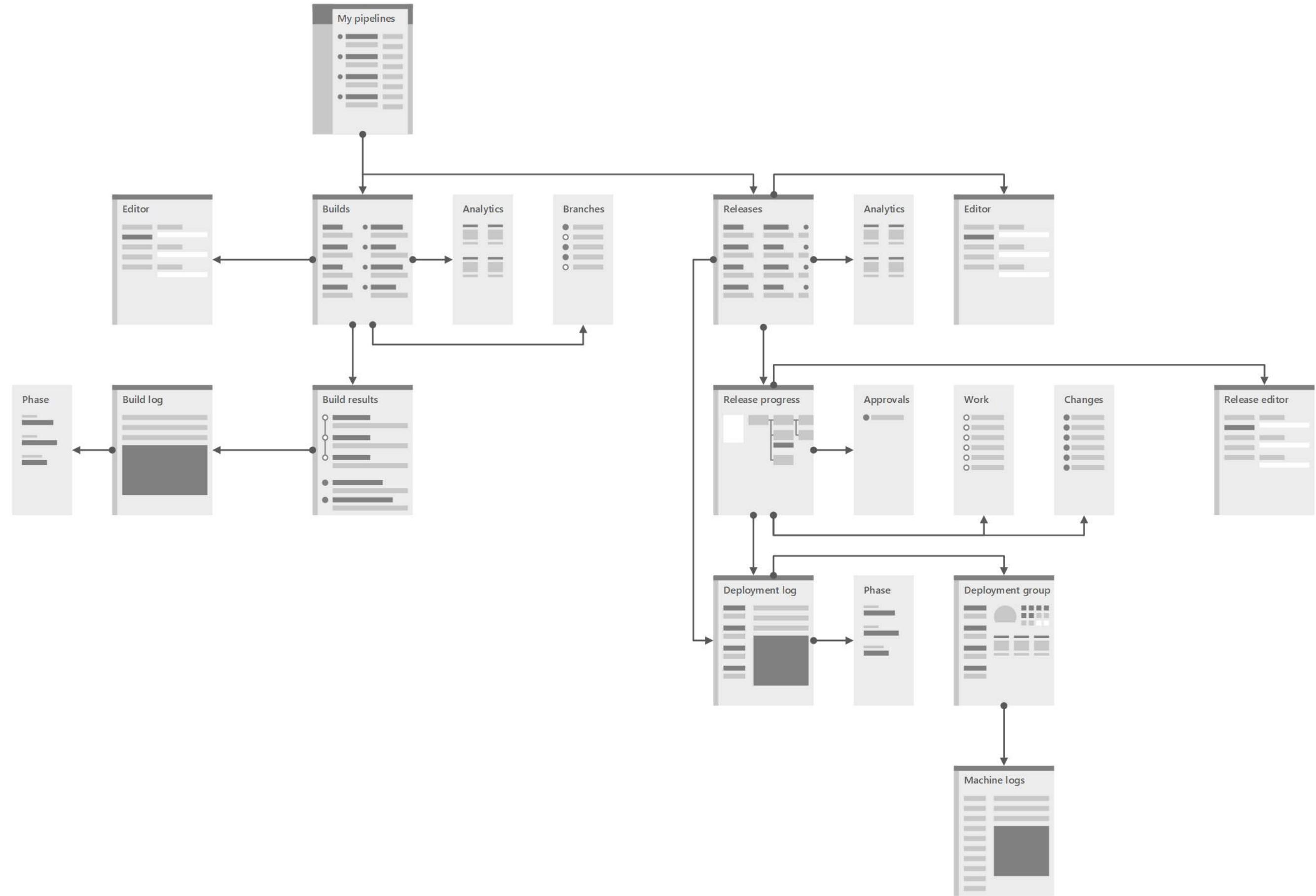
The views for automated builds were quite complex and could be challenging to navigate. This particular screenshot represents a “worst-case” scenario, but you can see how it can be overwhelming.

The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with various navigation options: One, Overview, Boards, Repos, Pipelines, Builds, Releases, Library, Task groups, Deployment groups, XAML, OneBranch Pipeline, Test Plans, Artifacts, Compliance, and OneBranch. The 'Builds' option is selected. The main area displays a list of build history items for a pipeline named 'Portal-ThresholdTests-Dev'. The list includes columns for build ID, status, commit hash, date, and duration. Most builds are labeled as 'Merged PR' or 'Manual build for CDP Buddy'. The list is very long, illustrating the complexity mentioned in the text.

Build ID	Status	Commit Hash	Date	Duration
20432431	Success	1446836	2019-02-26 17:45	0:00:00
20432376	Success	1455466	2019-02-26 17:47	0:00:00
20432278	Success	1455242	2019-02-26 17:49	0:00:00
20432129	Success	1455242	2019-02-26 17:53	0:00:00
20431398	Success	1455242	2019-02-26 17:56	0:00:00
20431370	Success	1455242	2019-02-26 17:58	0:00:00
20431350	Success	1455242	2019-02-26 17:59	0:00:00
20431331	Success	1455242	2019-02-26 18:00	0:00:00
20431317	Success	1455242	2019-02-26 18:02	0:00:00
20430980	Success	1455242	2019-02-26 18:39	0:00:00
20429990	Success	1455242	2019-02-26 18:43	0:00:00
20428789	Success	1447618	2019-02-26 18:46	0:00:00
20428558	Success	1447618	2019-02-26 18:58	0:00:00
20428399	Success	1409394	2019-02-26 18:58	0:00:00
20428112	Success	azurite_one_compute_move_retail_cr...	2019-02-26 18:46	0:00:00
20428089	Success	ACS-RPv2-Pipeline	2019-02-26 18:53	0:00:00
20428066	Success	1454906	2019-02-26 19:30	0:00:00
20427913	Success	1455675	2019-02-26 19:40	0:00:00
20427656	Success	azurite_one_compute_move_retail_x86	2019-02-26 19:40	0:00:00
20427603	Success	1446672	2019-02-26 19:50	0:00:00

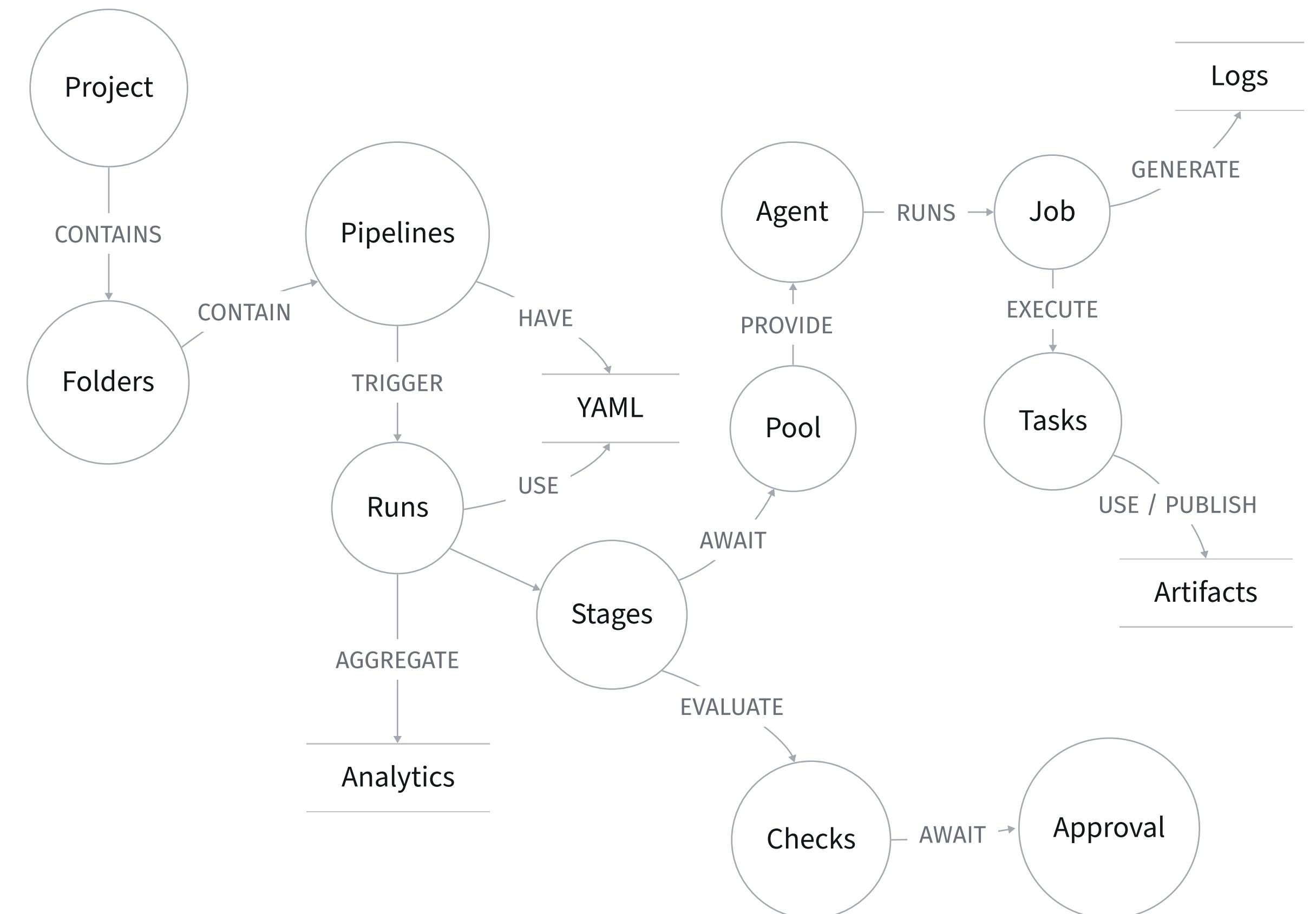
Wireflow

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.



Early exploration

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

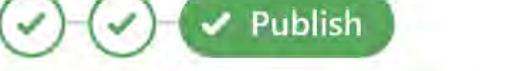
I put a heavy visual emphasis on the most recently completed stage and de-emphasized stages that finished successfully.

I used vertical delimiters to represent stages running in parallel.

Pipelines

Mine & active All pipelines

New ...

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

MM

Recent

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

MM

Recent

Pipelines

mobile-ios-app

Runs Branches Analytics

mobile-ios-app runs

Description Stages

Description	Stages	Time
#8 • Added testing for get_service_instance_... PR automated ↗ master 0031zf1	✓ - ✓ - ⚡ - ○	just now 23m 8s
#7 • Merge pull request 401701 from users/f... Manually triggered ↗ develop 0081mg4	⌚ - ○ - ○ - ○	2h ago 5m 2s
#6 • Update user service PR automated ↗ master 0032ff9	✓ - ✓ - ✓ - ✓ - ✓	6h ago 33m 1s
#5 • Add a request body validator Release new-features ↗ master 0064kp7	✓ - ✓ - ✗ - ⚡ - ○	7h ago 4m 17s
#4 • Cleanup notifications styling Scheduled ↗ master 0031zf1	✓ - ✓ - ✓ - !	12h ago 1h 14m 8s
#3 • Add extra padding on cells PR automated ↗ test 0077zg2	✓ - ✓ - ⚡ - ○ - ○	Yesterday 49m 2s
#2 • Make a property getter callable PR automated ↗ master 0077zg0	✓ - ✓ - ✓ - ⚡ - ✓	May 10 42m 17s
#1 • Update dashboard based on the new sc... PR automated ↗ master 0031zf1	✓ - ✓ - ✓ - ✓ - ✓	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

Runs Branches Analytics

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

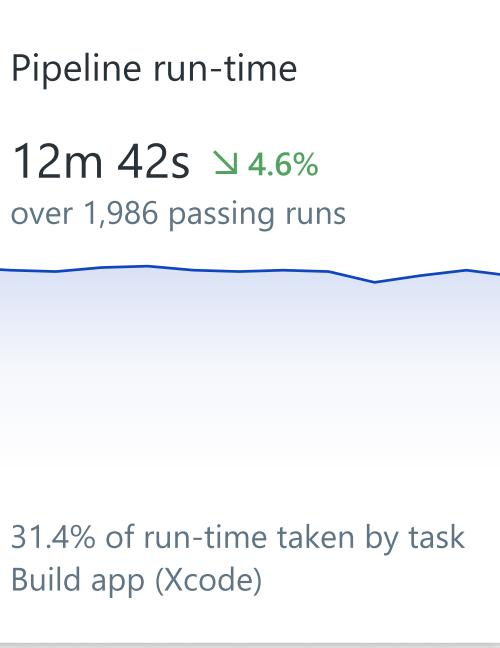
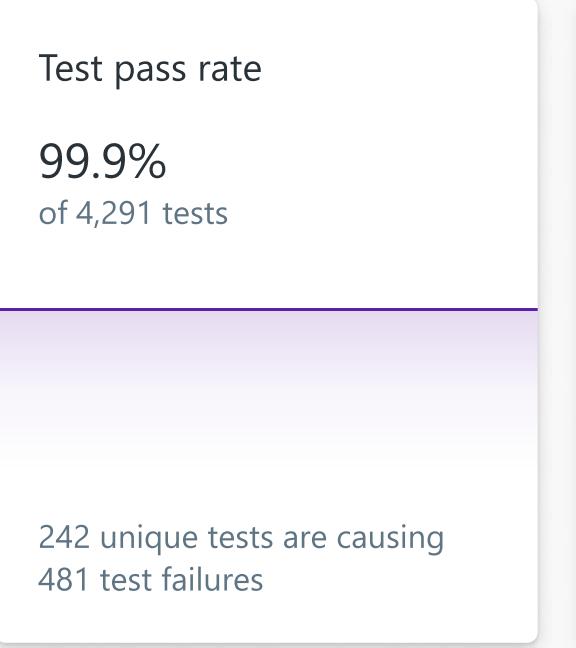
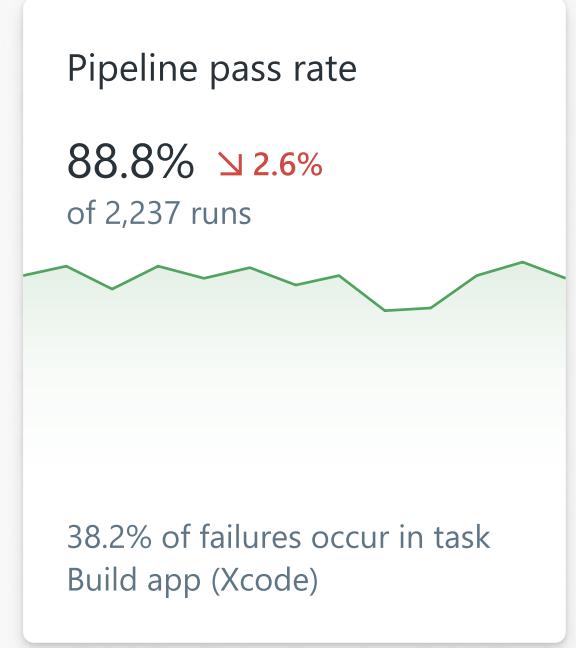
38.2% of failures occur in task Build app (Xcode)

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)



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.

← #689 • Update user service

multiple-stages

Build

- ✓ Build services 14s
- ✓ Initialize job <1 s
- ✓ Checkout 13s
- ✓ Build packages
- ✓ Publish artifacts
- ✓ Finalize job

Test

- > ✓ Integration tests 14s
- > ✓ Performance tests 14s

Deploy

- > ✓ Deploy to web 14s

✓ Checkout

Starting: Checkout

Task : Get sources

Description : Get sources from a repository. Supports Git, TfsVC, and SVN repositories.

Version : 1.0.0

Author : Microsoft

Help : [More Information](https://go.microsoft.com/fwlink/?LinkId=798199)

Syncing repository: AzureDevOps (Git)

Prepending Path environment variable with directory containing 'git.exe'.

git version

git version 2.21.0.windows.1

git config --get remote.origin.url

git config gc.auto 0

git config --get-all http.https://mseng.visualstudio.com/AzureDevOps/_git/_full/AzureDevOps.extraheader

git config --get-all http.proxy

git -c http.extraheader="AUTHORIZATION: bearer ***" fetch --force --tags --prune --progress --no-recurse-submodules origin

remote: Azure Repos

remote:

remote: Found 1301 objects to send. (1001 ms)

Receiving objects: 100% (1301/1301), 748.93 KiB | 7.06 MiB/s, done.

Resolving deltas: 100% (814/814), completed with 564 local objects.

From https://mseng.visualstudio.com/AzureDevOps/_git/_full/AzureDevOps

- [deleted] (none) -> origin/users/charry/getting-started-fixes

- [deleted] (none) -> origin/users/chsalgad/hashDetectionBuildType-on-teams-governance/releases/M157

- [deleted] (none) -> origin/users/davwolf/disableSignalRPingFeatureFlag

- [deleted] (none) -> origin/users/jorush/removing-vertical-command-bar

- [deleted] (none) -> origin/users/nickkirc/work0908

- [deleted] (none) -> origin/users/nimundra/org-audit-events

- [deleted] (none) -> origin/users/raja/update-token-npe-settings

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.

- Pipeline View:** Shows the "eShopOnWeb" pipeline with four recent runs listed:
 - #20200820.1 Refactor deployment stage (Manually triggered for master, duration 75945ms)
 - #20200318.17 Refactor deployment stage (Individual CI for master, duration 75945ms)
 - #20200318.16 Refactoring build stage to co... (Individual CI for master, duration 4c104ms)
 - #20200318.13 Update azure-pipelines.yml f... (Individual CI for master, duration 5f0f0ms)
- Run Details View:** Displays the summary for run #20200820.1, which was manually triggered by Jim Lamb. It shows the repository and version information (eShopOnWeb@master 75945c0). The "Jobs" tab is selected, showing two stages:

Name	Duration
Build image: Build	1m 57s
Development: Publish image	13s
- Job Log View:** Provides a detailed log for the "Build" job of run #20200820.1. The log includes:
 - Build (duration 1m 57s)
 - Initialize job
 - Checkout eShopOnWeb@master to s (duration 4s)
 - Build & push (duration 1m 50s)
 - Post-job: Checkout eShopOnWeb... (duration <1s)
 - Finalize Job (duration <1s)

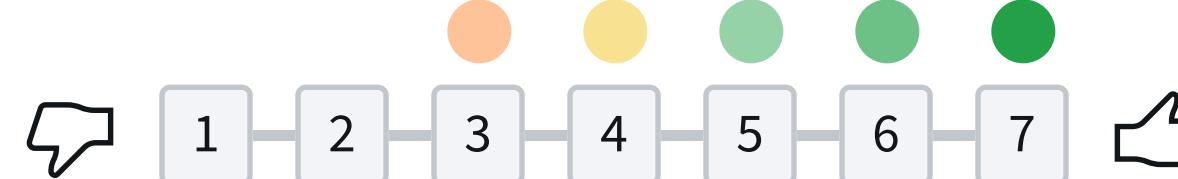
UX Research

Directed interviews,
roundtables, and card sorts
with “modern
developers”

Usability tests based on
static mock-ups and
click-through prototypes

I thought the system was easy to use

OLD

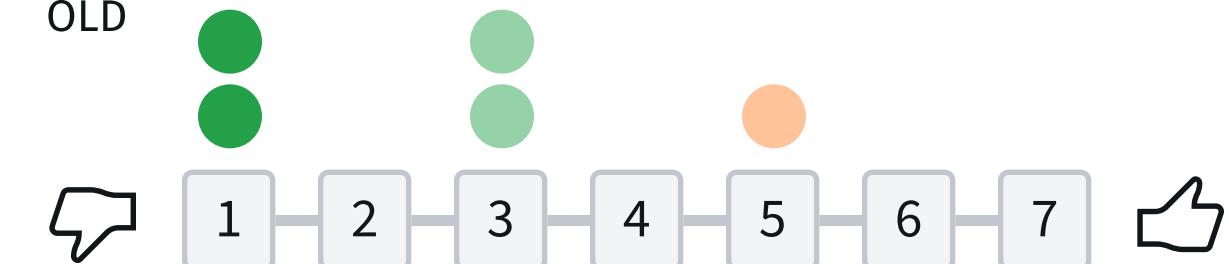


NEW

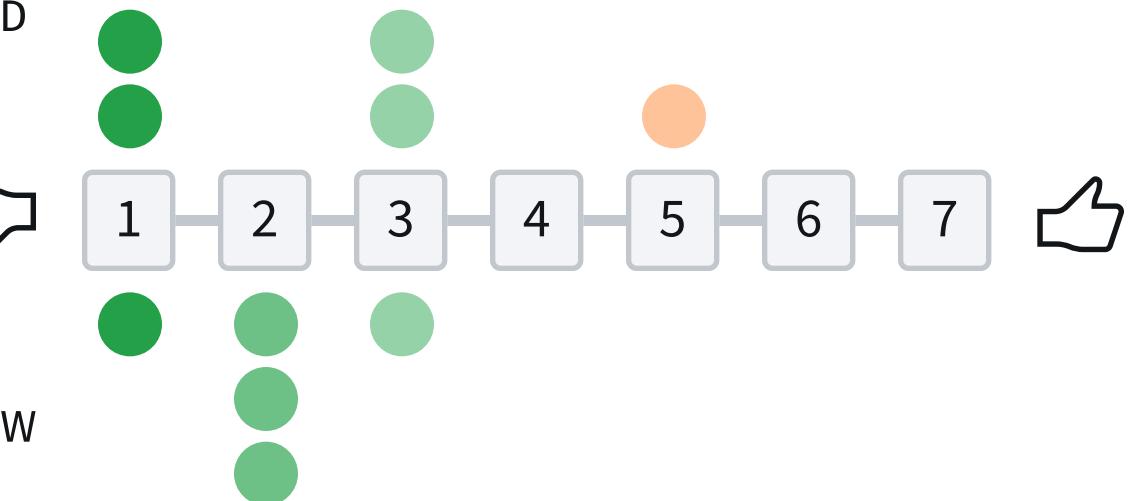


I found the tool unnecessarily complex

OLD

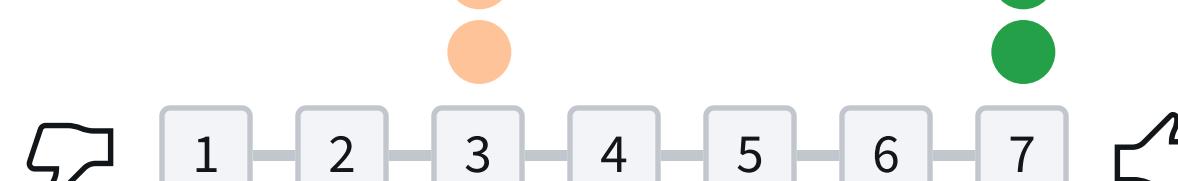


NEW

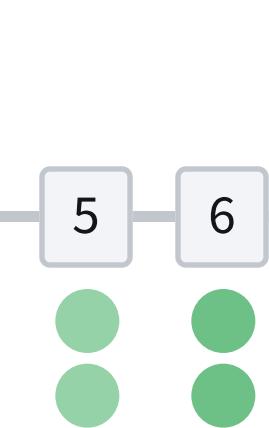


I felt very confident using the system

OLD

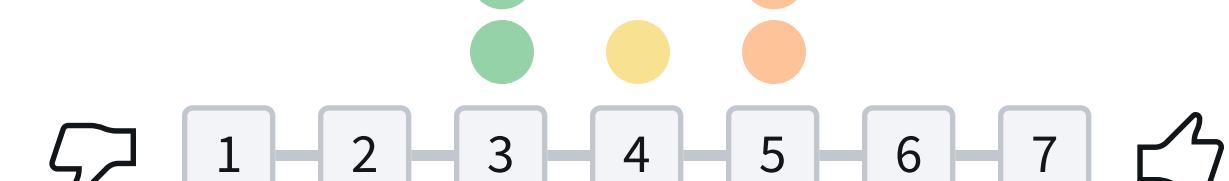


NEW

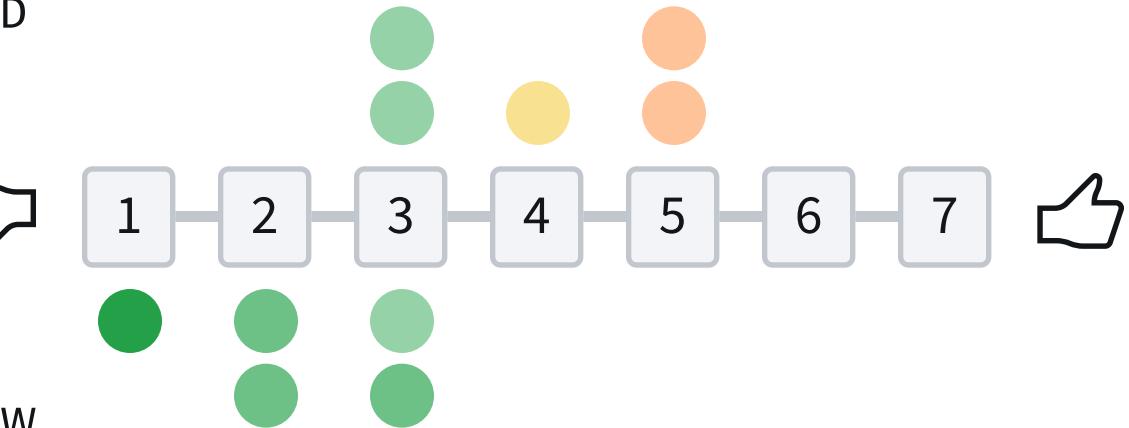


I found the system very cumbersome to use

OLD



NEW



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.

Opportunities for improvement

- 🔮 Full text search experience for logs
- 🔮 Full featured error and warning analysis experience
- 🔮 Reduce log latency and simulate live logging experience
- 🔮 Notification emails for weekly analytics, key indicators

Project #3

Failure analysis

Diagnosing pipeline run failures is difficult due to the number of possible failure points. Support spends too much time investigating run failures that users should be able to resolve themselves.

The screenshot shows a pipeline run summary for a pull request. The pipeline is named "#5 • Add a request body validator" and is a "multiple-stages" type. The summary card includes details like the repository (AdventureWorks), version (513226), start time (Today at 9:02 AM), duration (6m 48s), related user stories (1), artifacts (2), test coverage (100% passed, 87.2% lines covered), and changes (View changes). Below the summary is a detailed view of the "Errors" section, which lists three specific failure points. The stages section shows the flow from Build to Test to Deploy, with the Build stage failing and both Test and Deploy stages being skipped.

#5 • Add a request body validator
multiple-stages

Pull request 4583 by Colin Ballinger

Repository and version

AdventureWorks

513226

Time started and elapsed

Today at 9:02 AM

6m 48s

Related

1 user story

2 artifacts

Tests and coverage

100% passed

87.2% lines covered

View changes

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

Build 1 job completed 6m 48s

Test 1 job skipped

Deploy 1 job skipped

The Challenge

- ✓ Provide decision support for diagnosing failed runs.
- ✓ Improve customer satisfaction.
- ✓ Reduce support costs by making it easier for users to self-diagnose failures.

My Role

I proposed this experience when my team was looking for ways to reduce support costs without negatively affecting customers.

Error presentation

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

#AzureDevOps_merge_20191017.227
on AzureDevOps

Summary

Pull request 507297 by Colin Ballinger [View changes](#)

Repository and branch	Time started and elapsed	Tests	Related
AzureDevOps	Today at 2:07 PM	Get started	1 user story
507297	17m 49s		

Error

Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output conflict detected: the file c:\v2.0\A1_work\1\obj\Release.AnyC... [Troubleshooting failed runs](#)

Job

Name	Status	Duration
PR	Failed	17m 44s

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.

#AzureDevOps_merge_20191017.227
on AzureDevOps

Summary

Pull request 507297 by Colin Ballinger

Repository and branch AzureDevOps

Time started and elapsed Today at 2:07 PM
17m 49s

Error

- Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output Stage (if more than one) • Job (Pool/Agent) • Task (line #)

Job

Name 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](#)

Variables

When the value of a variable has changed since the last successful run, it's a possible cause of the failure.

#AzureDevOps_merge_20191017.227 • Status L0 Warning
on AzureDevOps

Summary

Pull request 507297 by Colin Ballinger

Repository and branch
AzureDevOps
507297

Time started and ended
Today at 2:07 PM
17m 49s

Error

Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output conflict
Stage (if more than one) * Job (Pool/Agent) * Task (line #)

Job

Name	Status	Duration
PR	Failed	17m 44s

← Variables

Modified 3 Stable

Changed since last successful run

first-variable-name
[old-value] → [new-value]

second-variable-name
[old-value] → [new-value]

secret-variable-name
Secret value was modified

Task

It's possible for a task to be updated between pipeline runs. While these updates shouldn't generally have unexpected changes, third-party tasks may not rigorously conform to this requirement.

The screenshot shows a pipeline run titled "#AzureDevOps_merge_20191017.227" with a status of "L0 Warning". The summary indicates it's on AzureDevOps and triggered by a Pull request (PR) 507297 by Colin Ballinger. The repository and branch are AzureDevOps, PR 507297. The task started today at 2:07 PM and took 17m 49s. A warning message states: "Version changed since last successful run v1.160.0 (previously v1.159.0)". The error section shows a detailed error message: "Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output conflict detected. Stage (if more than one) * Job (Pool/Agent) * Task (line #)". The job section lists a single job named "PR" which failed after 17m 44s.

← MSBuild task

Build with MSBuild

! Version changed since last successful run
v1.160.0 (previously v1.159.0)

This task has failed 41 times across 102 pipelines in the last 14 days.

[View analytics](#)

Property	Value
Display name	Build solution dirs.proj
Project	**/*.sln
MSBuild	Version
MSBuild Version	Latest
MSBuild Architecture	MSBuild x86
Configuration	Release
MSBuild Arguments	
Clean	False
Enabled	True

Helpful links

[About this task](#)

YAML History

Changes to the pipeline's YAML are more of a jumping off point for further investigation.

#AzureDevOps_merge_20191017.227 • Status L0 Warning
on AzureDevOps

Summary

Pull request 507297 by Colin Ballinger

Repository and branch
AzureDevOps
507297

Time started and ended
Today at 2:07 PM
17m 49s

Error

Toolsets\ConflictDetection\ConflictDetection.targets(57,5): Error : Output conflict
Stage (if more than one) • Job (Pool/Agent) • Task (line #)

Job

Name	Status	Duration
PR	Failed	17m 44s

← Pipeline YAML history

Changes since last successful run

Update pool demand
f0fde15c by Kristin Patterson May 6 at 4:41 PM

Merged PR 421518: Valid job names
ea763646 by Erik Nason Jan 7 at 8:11 PM

Earlier changes

Update pool demand
f0fde15c by Lydia Bauer May 6 at 4:41 PM

Merged PR 421518: Valid job names
ea763646 by Erik Nason Jan 7 at 8:11 PM

Outcomes

- 😊 Reduced support call volume due to pipeline failures.
- 😊 Improved customer satisfaction.

Lessons

When resources are limited, plan for staged delivery.

Opportunities for improvement

- 🔮 Analysis of issues across runs.
- 🔮 Analysis of issues across pipelines and repositories.
- 🔮 Ruling out factors based on evidence from other runs.

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>
-  jaime@jflamb.com