Forget Platform Engineering - Think Developer Productivity

A customer-centric approach to developer platforms

Christian Denich

Sr. Customer Solution Manager
Amazon Web Services

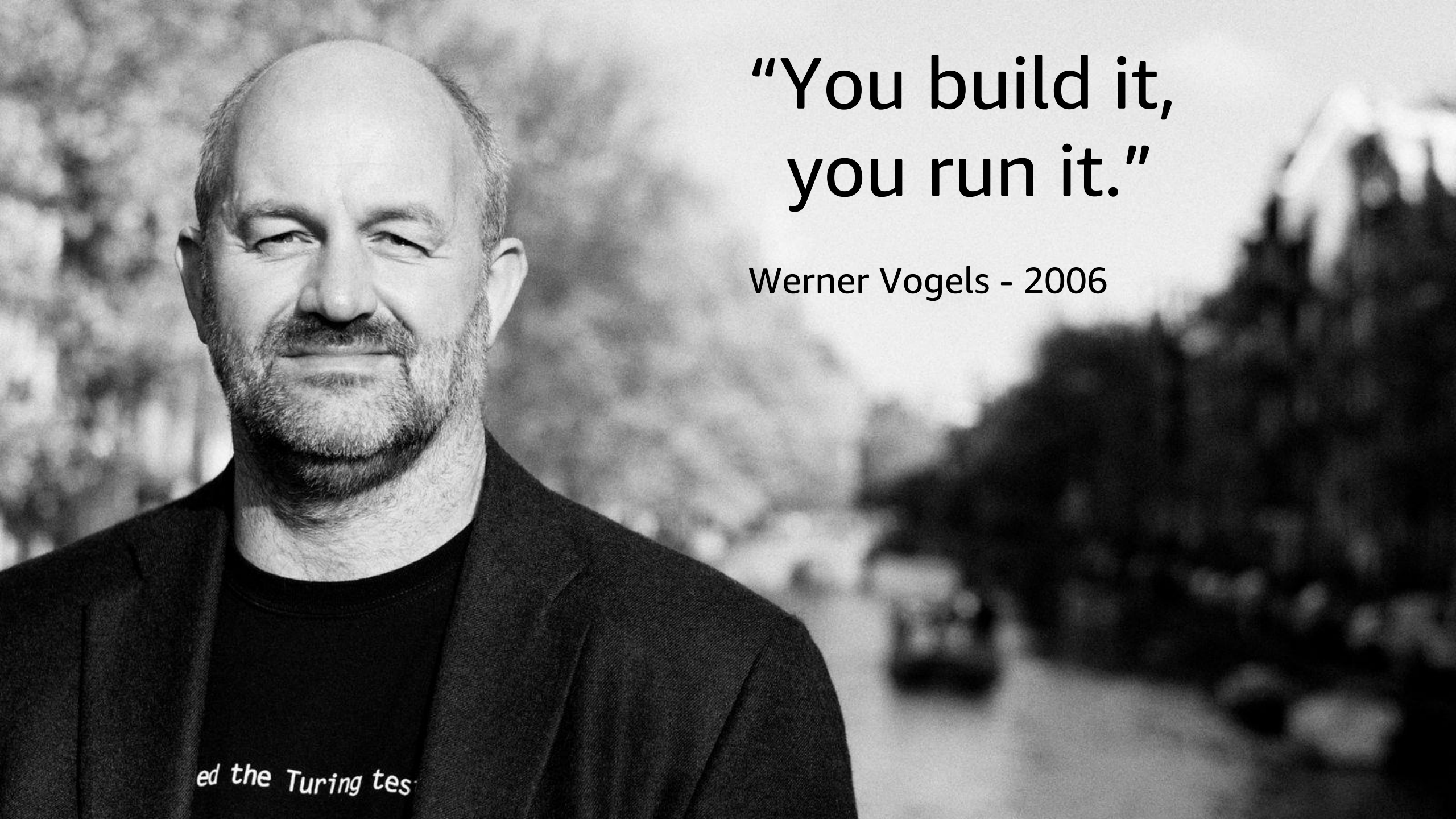
Robert Hoffmann

Sr. Solutions Architect
Amazon Web Services

Don’t loose yourself in (traditional) Platform Engineering by operating a lot of software in a centralized way.

Instead, compose and integrate cloud & SaaS, only build if you must.

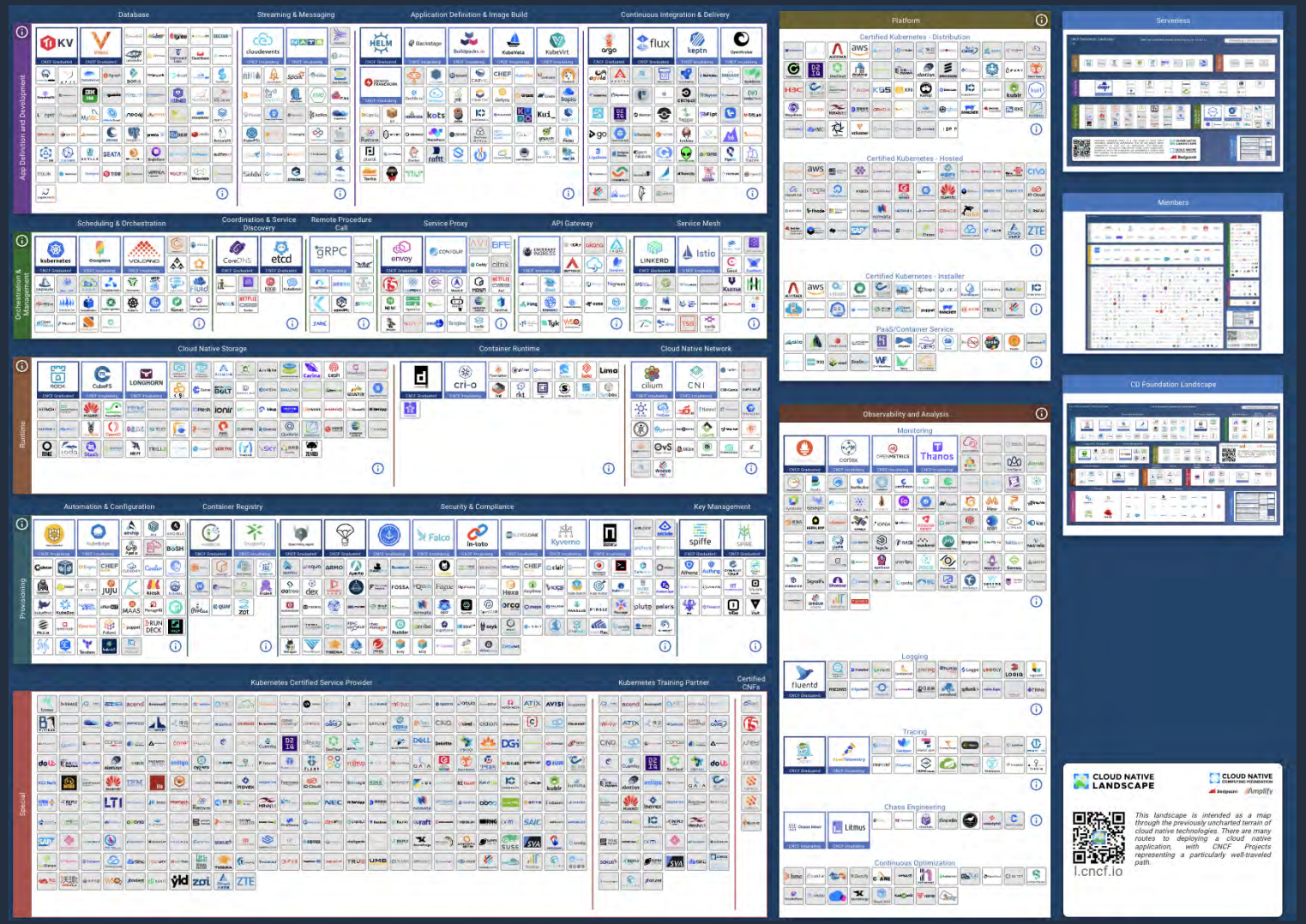
To spend your resources on the right things, you need think about Developer Productivity first, and work backwards from your customers – the developers.



“You build it, you run it.”

Werner Vogels - 2006

extraneous cognitive load



https://landscape.cncf.io/

“Platforms are a means of centralizing expertise while decentralizing innovation to the customer or user”

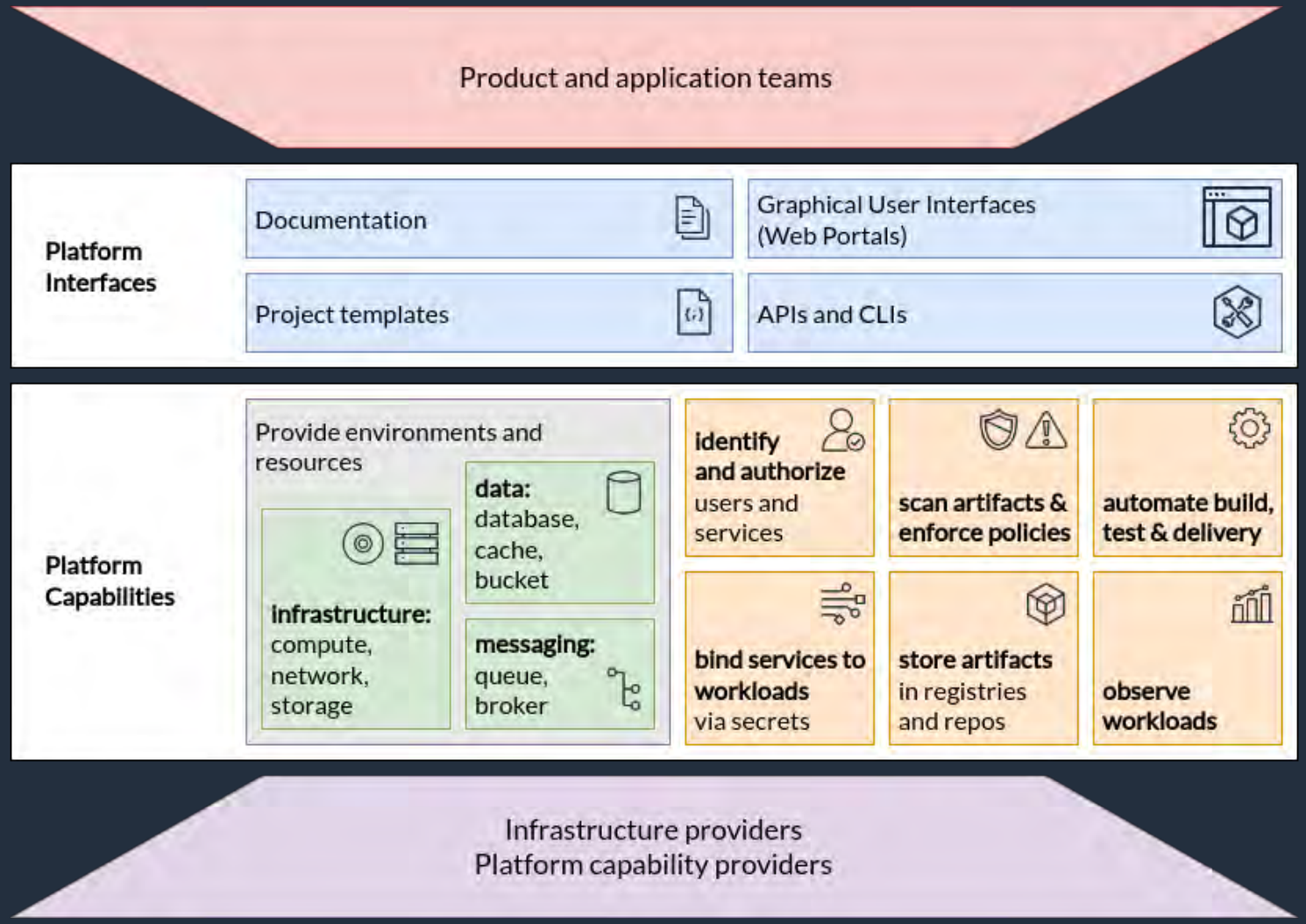
Peter Gillard-Moss

ThoughtWorks

There are many capabilities in a dev platform...

CNCF Platforms White Paper

https://tag-app-delivery.cncf.io/whitepapers/platforms/



The downward spiral of Platform Engineering: Running too much and the wrong things

1. prioritizing the wrong workstreams and features
2. reinventing the wheel by building undifferentiated services
3. spending most resources on operating these services
4. clinging to these services even if they should be deprecated
5. believing to be good at building abstractions
6. only measuring the existing services

"You have a platform engineering org, which wraps & packages your infrastructure needs...

by running as little infra as possible"

Charity Majors

Co-founder and CTO at Honeycomb

"Perils, Pitfalls and Pratfalls of Platform Engineering",
https://www.infoq.com/presentations/platform-engineering-teams/

To spend your resources on the right things, you need think about Developer Productivity first, and work backwards from your customers – the developers.

Nice buzzword, but how can we do this in practice?

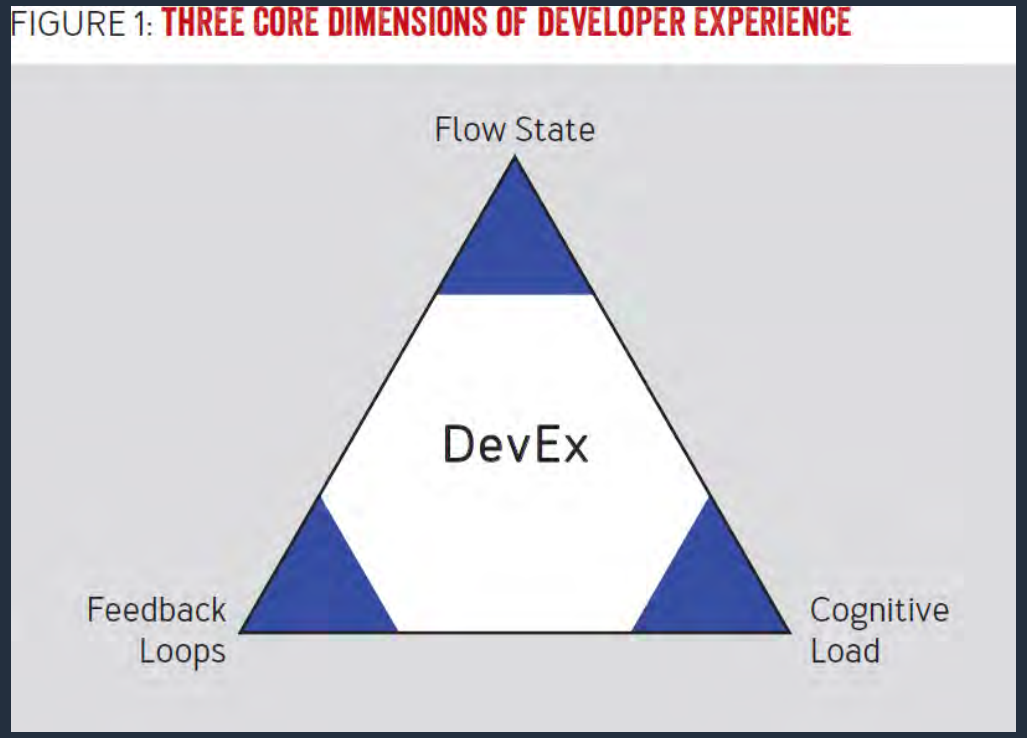
By putting DevEx at the center.

Developer experience (DevEx) is a developer-centric approach to improving developer productivity.

Feedback Loops: When a developer makes a change, can they get feedback about that change fast enough?

Cognitive Load: How much stuff do developers need to keep track of in order to complete a task?

Flow State: How often and how easy can a developer to get into a state of energized focus?

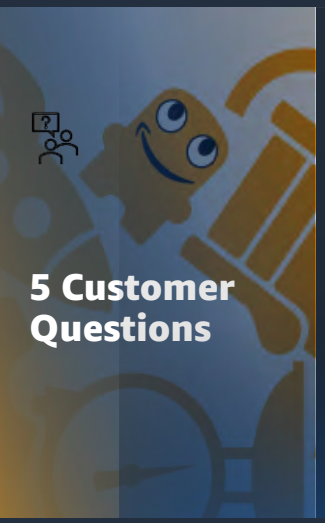


Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Michaela Greiler
"DevEx: What Actually Drives Productivity"
https://queue.acm.org/detail.cfm?id=3595878

"There are many advantages to a customer-centric approach, but here's the big one: **Customers are always beautifully, wonderfully dissatisfied,** even when they report being happy and business is great. Even when they don't yet know it, customers want something better, and your desire to delight customers will drive you to invent on their behalf."

- Jeff Bezos, Founder and Executive Chair, Amazon.com, Inc.
- 2016 letter to shareholders

Working backwards to PRFAQ & Visuals



5 Customer Questions

Who is the customer, and what insights do we have about them?

What is the prevailing customer problem or opportunity?

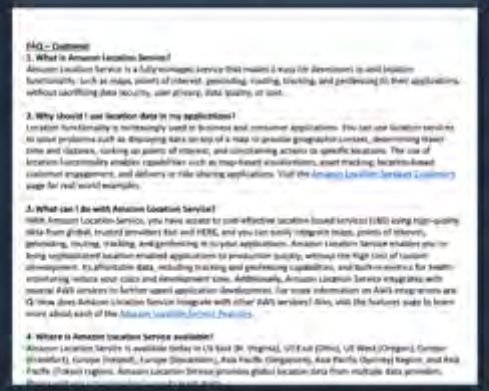
What is the solution and the most important customer benefit?

How do we describe the solution and experience to customers?

How do we test the solution with customers and measure success?

Press Release

FAQs



Visuals

Working backwards, refined by DevEx

You need to ask your customers, the developers!

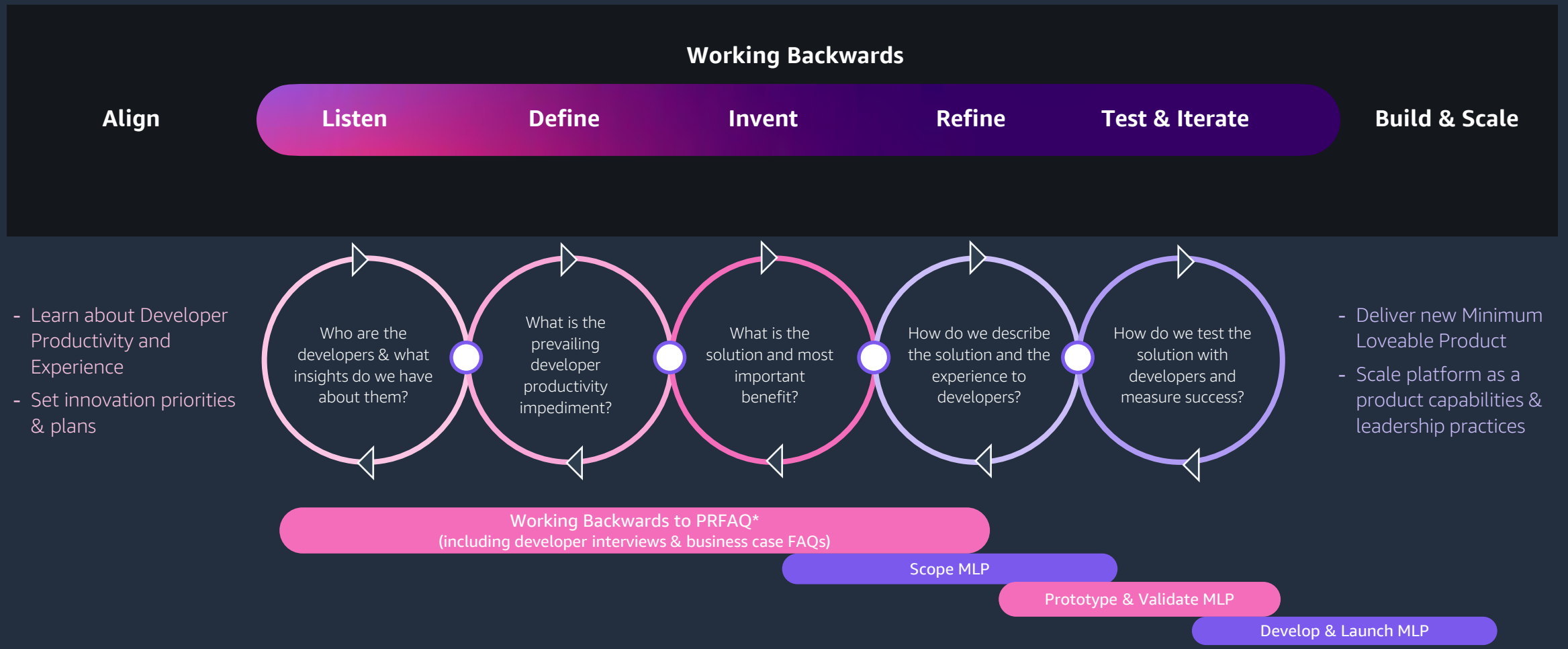
The perceptual measures outlined in the DevEx framework are best collected through a developer experience survey / interview.

TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS *Human attitudes and opinions*	• Satisfaction with automated test speed and output • Satisfaction with time it takes to validate a local change • Satisfaction with time it takes to deploy a change to production	• Perceived complexity of codebase • Ease of debugging production systems • Ease of understanding documentation	• Perceived ability to focus and avoid interruptions • Satisfaction with clarity of task or project goals • Perceived disruptiveness of being on-call
WORKFLOWS *System and process behaviors*	• Time it takes to generate CI results • Code review turnaround time • Deployment lead time (time it takes to get a change released to production)	• Time it takes to get answers to technical questions • Manual steps required to deploy a change • Frequency of documentation improvements	• Number of blocks of time without meetings or interruptions • Frequency of unplanned tasks or requests • Frequency of incidents requiring team attention
KPIS *North star metrics*	• Overall perceived ease of delivering software • Employee engagement or satisfaction • Perceived productivity		

Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Michaela Greiler
"DevEx: What Actually Drives Productivity"
https://queue.acm.org/detail.cfm?id=3595878

Working Backwards from your Developers



To spend your resources on the right things, you need think about Developer Productivity first, and work backwards from your customers – the developers.

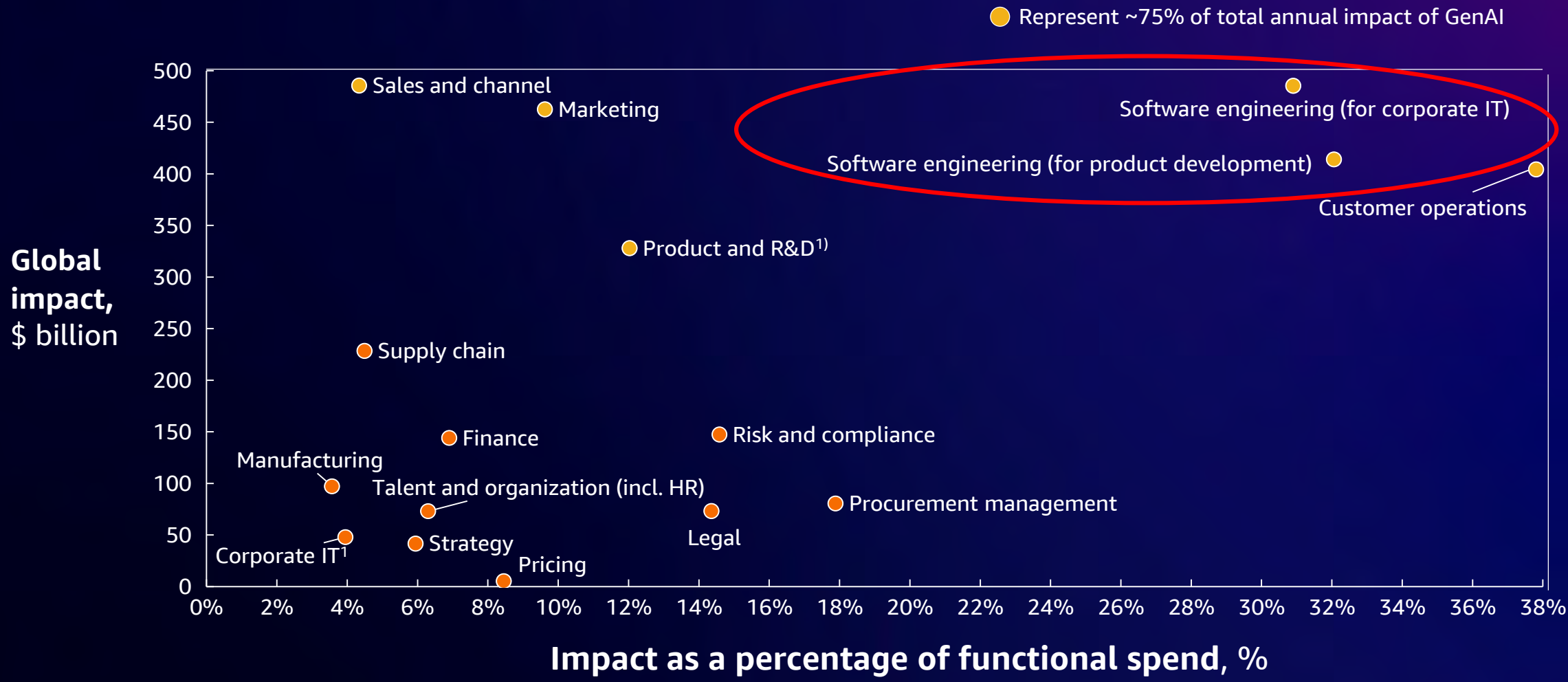
- Applying the DevEx framework helps you to understand your developers and the factors that impede their productivity.
- It helps platform teams to adopt the improvement of DevEx as their core “identity”, instead of a particular service or technology.
- Working backwards from DevEx opens up many different solutions to remove impediments – from a simple wiki page, to just a bunch of IaC templates, to the use of GenAI companions.

Unlock the next productivity frontier with Generative AI

Christian Denich

Customer Solution Manager @ AWS.

All views expressed are my own.



Note: Impact is averaged
1) R&D excluding software engineering
Source: McKinsey & Company (2023): The economic potential of generative AI: The next productivity frontier

How to measure developer productivity?

DORA

DevOps Research and Assessment

DORA: Comparing medium and high ranked teams

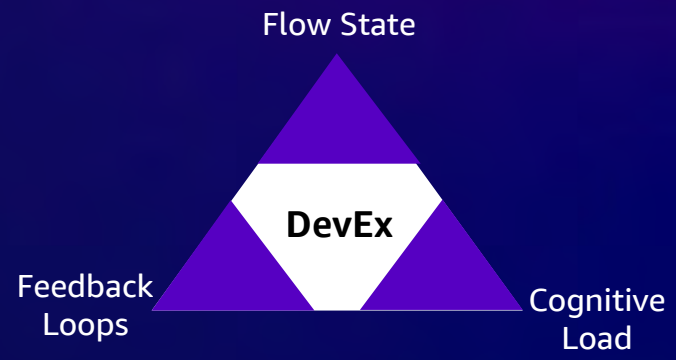
DEPLOYMENT FREQUENCY	Weekly - monthly	→	Daily - weekly
CHANGE LEAD TIME	1–4 weeks	→	1-7 days
CHANGE FAILURE RATE	15%	→	10%
TIME TO RESTORE SERVICE	1-7 days	→	< 1 day

SPACE

Satisfaction & Well-being
Performance
Activity
Communication & Collaboration
Efficiency & Flow

Developer Experience

Three core dimensions of DevEx



- There is no single dimension/metrics caputring productivity.
- Combine quantitative (instrumentation/telemetry) and qualitative (surveys!) data.
- **Developer Experience (DevEx)** may just be the best proxy for measuring **Developer Productivity.**

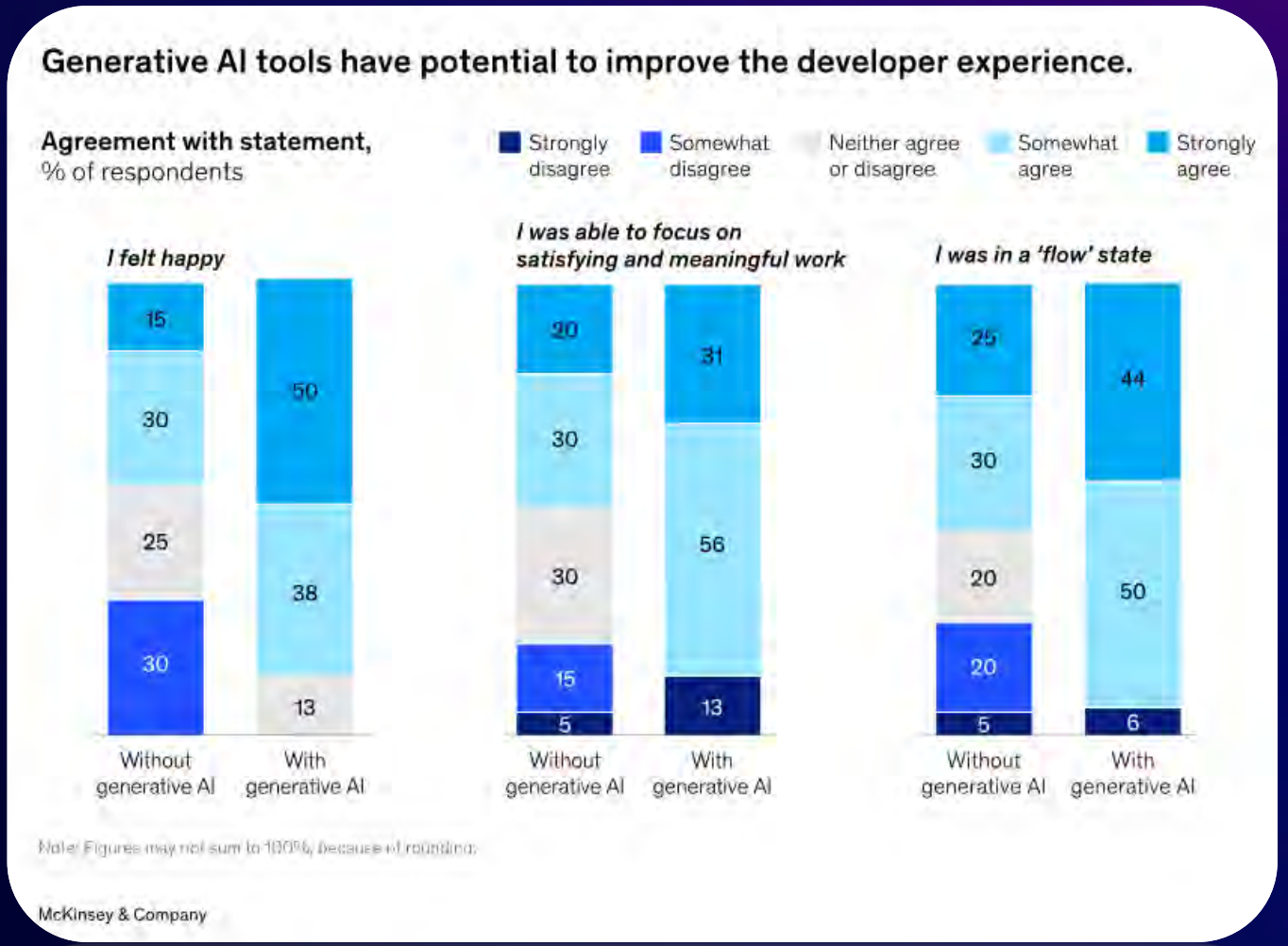
Source: DORA *State of DevOps report*, 2023,
„*DevEx in Action: A study of its tangible impacts*" accessed at https://queue.acm.org/detail.cfm?id=3639443,
„*DevEx: What Actually Drives Productivity*" accessed at https://queue.acm.org/detail.cfm?id=3595878.

GenAI tools can improve DevEx!

230% more engaged,
85% more likely to stay beyond 3 years

..if they have the **right technology supporting** their work.

Harvard Business Review Feb. 2022



Source: „*In a hybrid world, tech defines your employee experience*" Harvard Business Review Feb. 2022, McKinsey & Company (2023): "*Unleashing developer productivity with generative AI*".

Coding companion

- Natural language → multiple code suggestions
- Matches developer style and patterns
- Security scanning
- Open source reference tracking

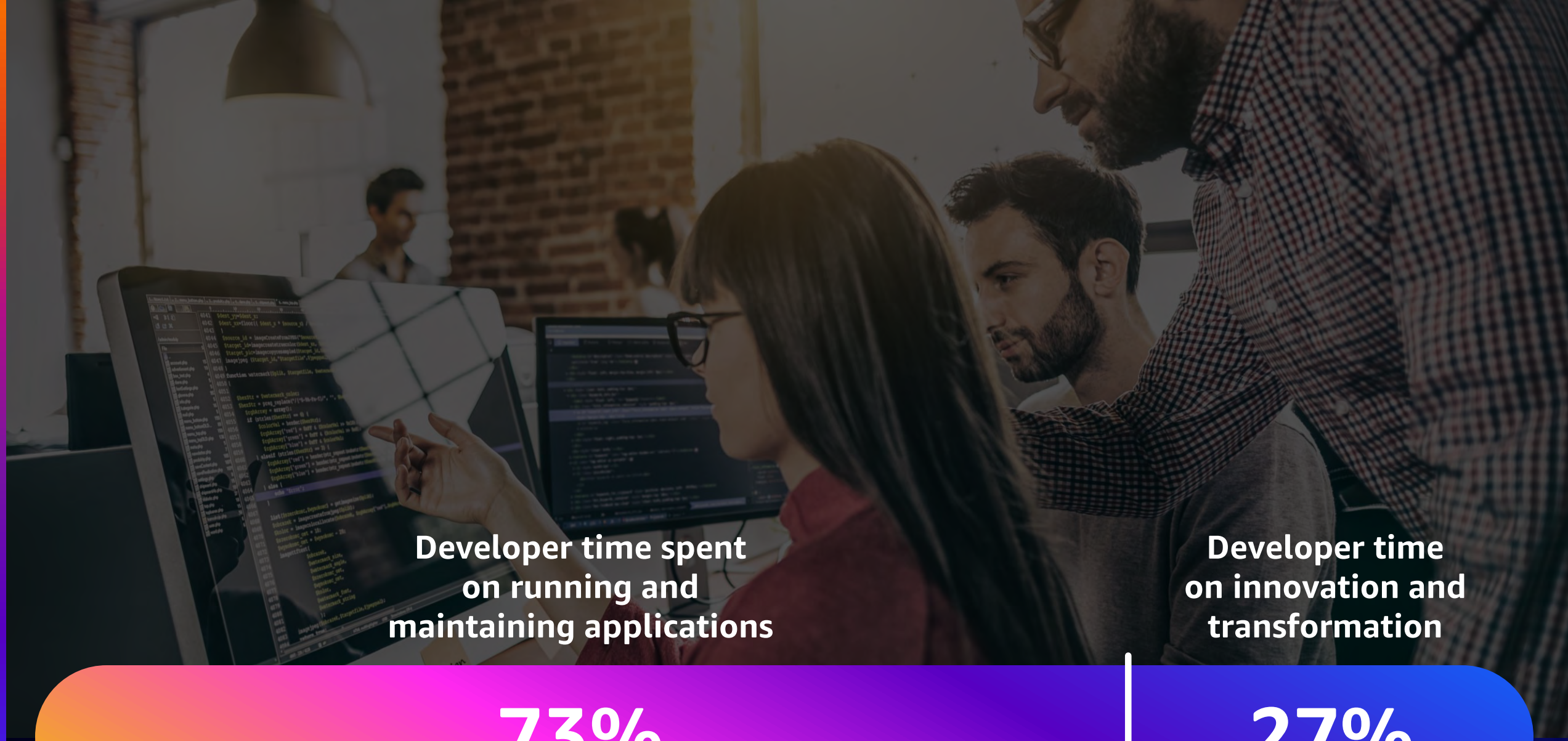
➢ 27% more likely to complete tasks successfully
➢ On average of 57% faster

➢ Felt more productive
➢ Less time searching
➢ Longer in „the flow“
➢ Higher job satisfaction

The typical developer spends 4h 21min coding

...per week. That is **52 min per day**.

Source: Software.com Global Code Time Report



Developer time spent on running and maintaining applications

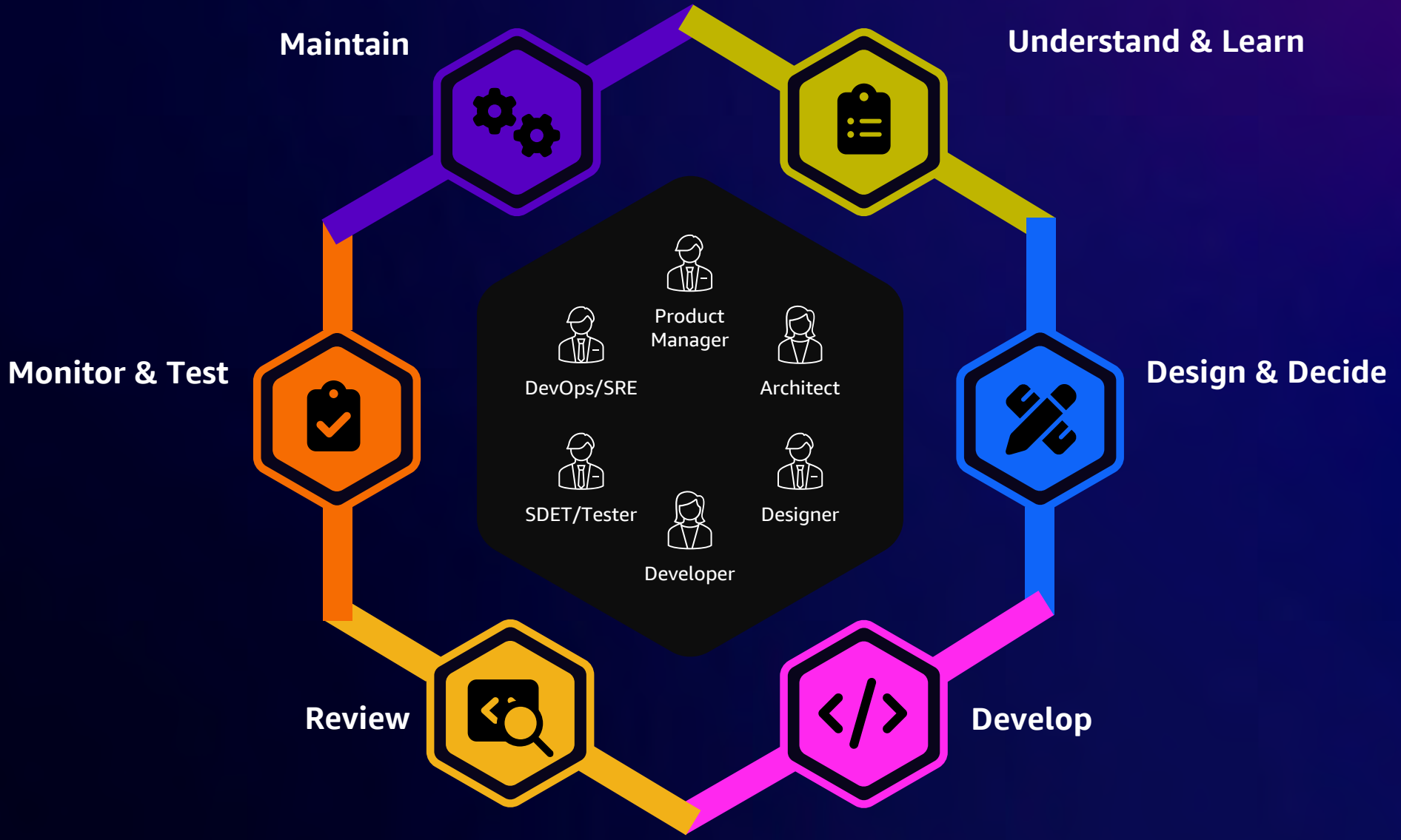
73%

Developer time on innovation and transformation

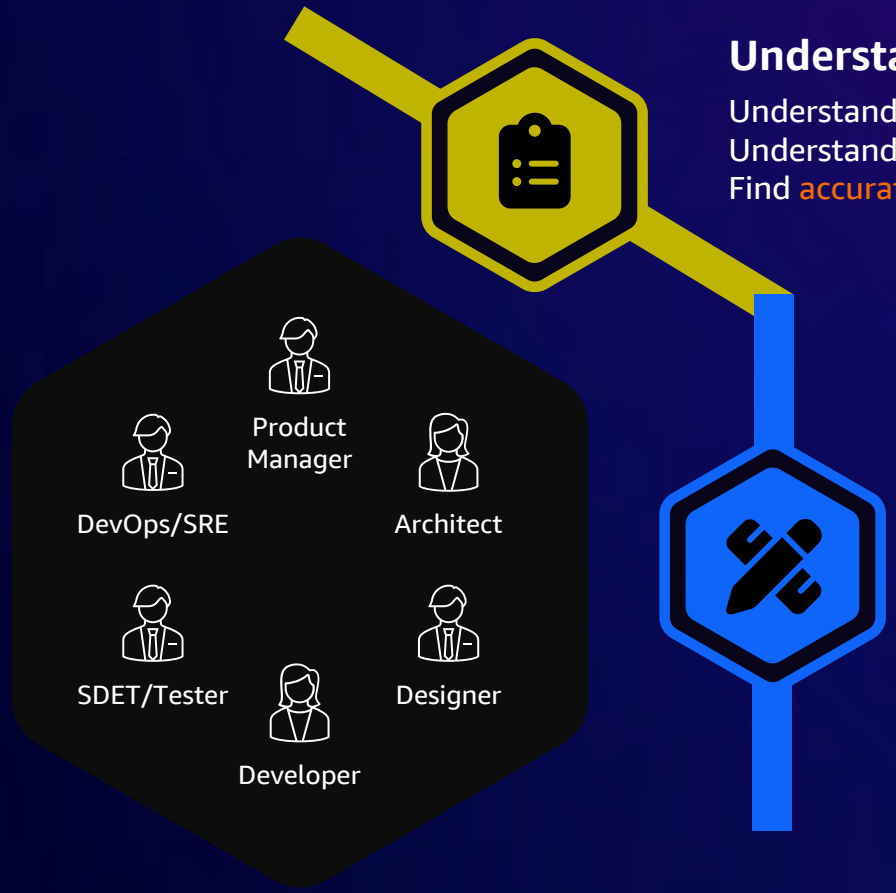
27%

Source: Gartner

The development companion: Reinventing across the development lifecycle



The development companion: Reinventing across the development lifecycle



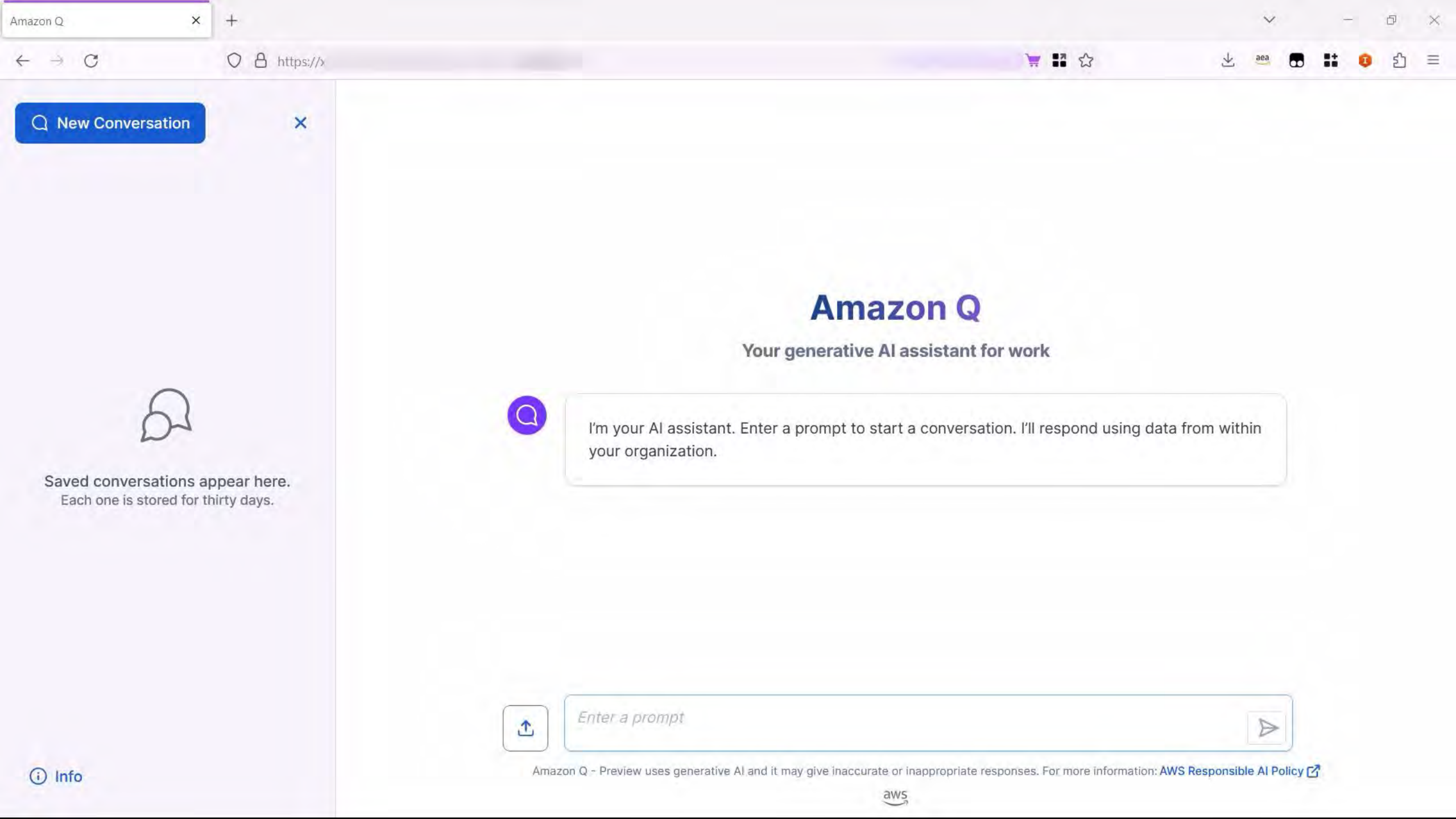
Understand & Learn

Understand requirements
Understand existing environment
Find accurate and specific technical guidance

Design & Decide

Design application based on business and technical requirements
Decide for an architecture

➢ **Knowledge Discovery Efficiency (KEDE)**


Amazon Q
https://
New Conversation
Saved conversations appear here.
Each one is stored for thirty days.
Info
Amazon Q
Your generative AI assistant for work
I'm your AI assistant. Enter a prompt to start a conversation. I'll respond using data from within your organization.
Enter a prompt
Amazon Q - Preview uses generative AI and it may give inaccurate or inappropriate responses. For more information: AWS Responsible AI Policy
aws

Cloud Excellence Center

Welcome to the Cloud Center of Excellence

The Cloud Center of Excellence (CCoE) provides governance, guidance, and resources to help teams adopt Amazon Web Services (AWS) across the organization. Our goal is to enable innovation and efficiency through the effective use of cloud while managing risk.

Enter a prompt

The development companion: Reinventing across the development lifecycle

Understand & Learn

Understand requirements
Understand existing environment
Find accurate and specific technical guidance

Design & Decide

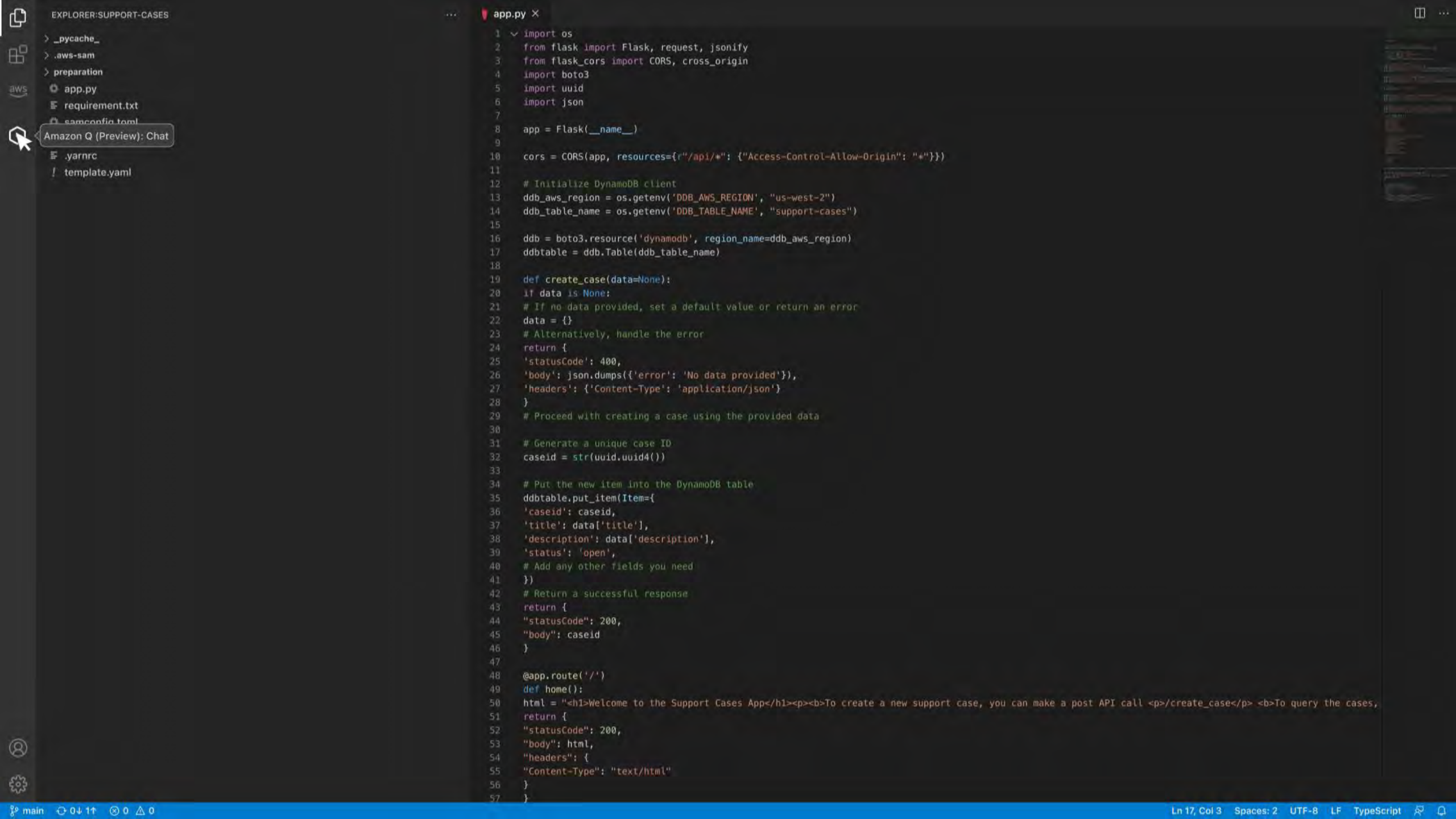
Design application based on business and technical requirements
Decide for an architecture

Develop

Understand existing code base
Generate code, open source reference tracking

Review

Summarize pull requests
Assist in identifying code issues
Compliant with coding standards
Security scanning


EXPLORER:SUPPORT-CASES
pycache
.aws-sam
preparation
app.py
requirement.txt
samconfig.toml
Amazon Q (Preview): Chat
.yarnrc
template.yaml
app.py
import os
from flask import Flask, request, jsonify
from flask_cors import CORS, cross_origin
import boto3
import uuid
import json
app = Flask(__name__)
cors = CORS(app, resources={r"/api/*": {"Access-Control-Allow-Origin": "*"}})
# Initialize DynamoDB client
ddb_aws_region = os.getenv('DDB_AWS_REGION', "us-west-2")
ddb_table_name = os.getenv('DDB_TABLE_NAME', "support-cases")
ddb = boto3.resource('dynamodb', region_name=ddb_aws_region)
ddbtable = ddb.Table(ddb_table_name)
def create_case(data=None):
if data is None:
# If no data provided, set a default value or return an error
data = {}
# Alternatively, handle the error
return {
'statusCode': 400,
'body': json.dumps({'error': 'No data provided'}),
'headers': {'Content-Type': 'application/json'}
}
# Proceed with creating a case using the provided data
# Generate a unique case ID
caseid = str(uuid.uuid4())
# Put the new item into the DynamoDB table
ddbtable.put_item(Item={
'caseid': caseid,
'title': data['title'],
'description': data['description'],
'status': 'open',
Add any other fields you need
})
# Return a successful response
return {
"statusCode": 200,
"body": caseid
}
@app.route('/')
def home():
html = "<h1>Welcome to the Support Cases App</h1> <b>To create a new support case, you can make a post API call /create_case <b>To query the cases,
return {
"statusCode": 200,
"body": html,
"headers": {
"Content-Type": "text/html"
}
}
main
0↓ 1↑
0
0
Ln 17, Col 3
Spaces: 2
UTF-8
LF
TypeScript

decorator. This indicates it will handle requests to the root URL path.

- Within the handler, it first checks if the request contains data in the form of a JSON object with 'title' and 'description' fields. This data will be used to create a new support case record.
- It then proceeds to generate a unique ID for the case using UUID. This ID will be used as the primary key for the record in DynamoDB.
- The data, along with the generated ID and a default 'open' status, is then written as a new item to the DynamoDB table specified by the ddbtable variable. This will store the case details in the database.
- Finally, it returns a 200 OK response along with the generated case ID. This allows the client to identify the newly created record.

In summary, this application provides a RESTful API endpoint to create new support cases, generating a unique ID and storing the details in a DynamoDB database table. The ID is returned to the client so they can reference the newly created record. This creates a basic backend for managing support cases in a serverless architecture.

```
29  # Proceed with creating a case using
30
31  # Generate a unique case ID
32  caseid = str(uuid.uuid4())
33
34  # Put the new item into the DynamoDB
35  ddbtable.put_item(Item={
36  'caseid': caseid,
37  'title': data['title'],
38  'description': data['description'],
39  'status': 'open',
40  # Add any other fields you need
41  })
42  # Return a successful response
43  return {
44  "statusCode": 200,
45  "body": caseid
46  }
47
48  @app.route('/')
49  def home():
50  html = "<h1>Welcome to the Support Ca
51  return {
52  "statusCode": 200,
53  "body": html,
54  "headers": {
55  "Content-Type": "text/html"
56  }
```

main 0↓ 1↑ 0 0

File Edit Selection View Go Run ···

demo

AMAZON Q (PREVIEW): CHAT

Chat

Hi, I'm Amazon Q. I can answer your software development questions. Ask me to explain, debug, or optimize your code. You can enter / to see a list of quick actions.

Try Examples:

Explain selected code

How can Amazon Q help me?

Ask a question or enter "/" for quick actions

4000/4000

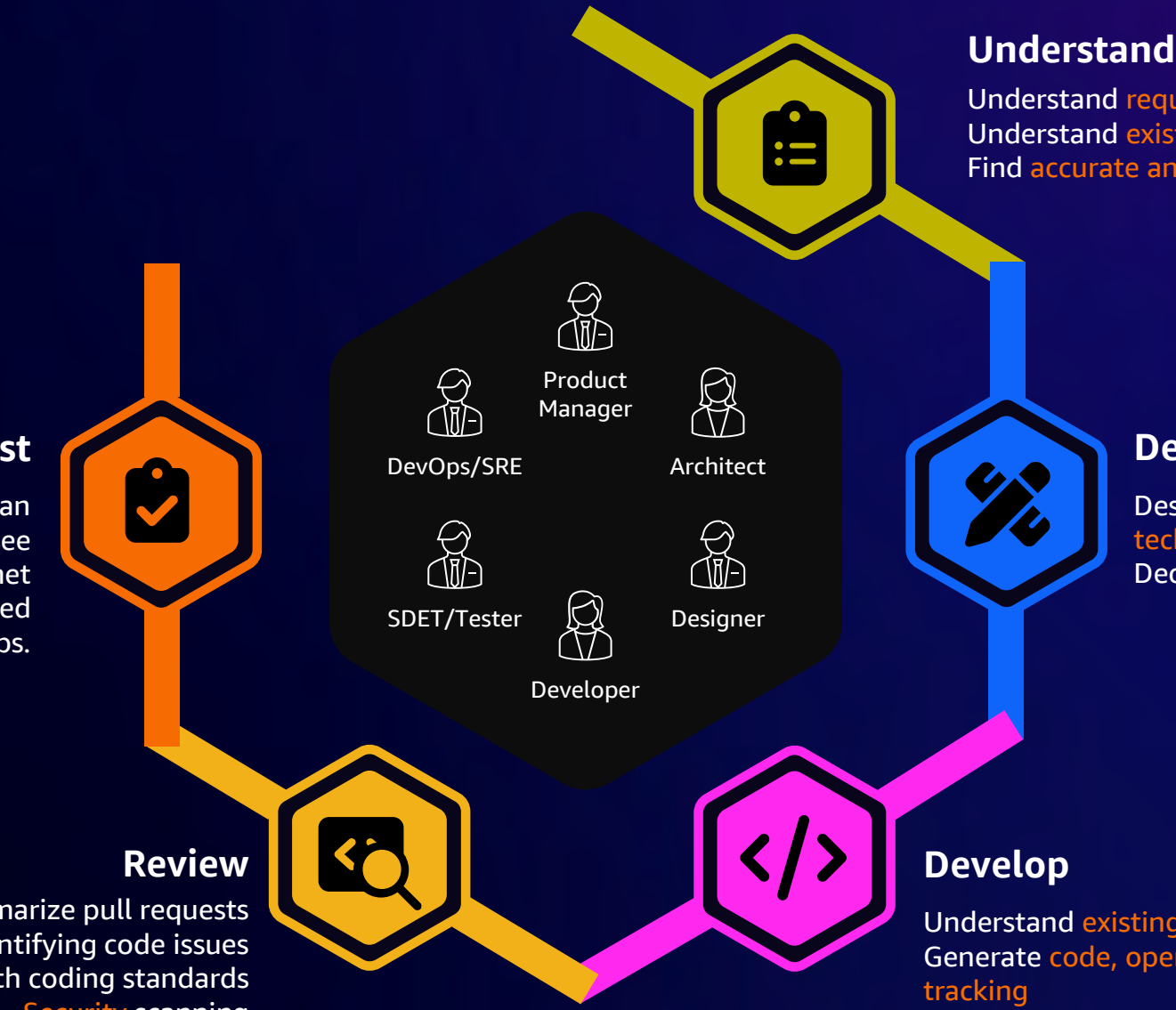
Use of Amazon Q is subject to the AWS Responsible AI Policy.

ProductController.java

src > main > java > com > example > spring > demo > demo > product > ProductController.java > Language Support for Java(TM) by Red Hat > Pro

```
30  public class ProductController {
31
32      private final ProductRepository repo;
33
34      public ProductController(ProductRepository repo) {
35          this.repo = repo;
36      }
37
38      @GetMapping
39      public ResponseEntity<ArrayList<Product>> getProducts() {
40          ArrayList<Product> products = repo.getProducts();
41          return new ResponseEntity<>(products, HttpStatus.OK);
42      }
43  }
44
```

The development companion: Reinventing across the development lifecycle



Understand & Learn

Understand requirements
Understand existing environment
Find accurate and specific technical guidance

Design & Decide

Design application based on business and technical requirements
Decide for an architecture

Develop

Understand existing code base
Generate code, open source reference tracking

Review

Summarize pull requests
Assist in identifying code issues
Compliant with coding standards
Security scanning

Monitor & Test

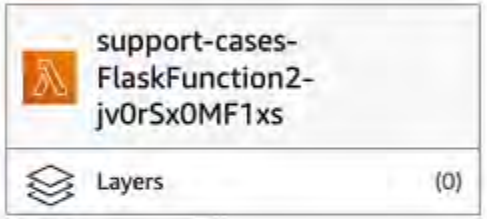
Security and performance of an application or cloud infrastructure to see that business needs are met
Track, diagnose and fix errors combined with feedback loops.

support-cases-FlaskFunction2-jv0rSx0MF1xs

Throttle | Copy ARN | Actions

This function belongs to an application. Click here to manage it.

Function overview Info



+ Add trigger

+ Add destination

Description
-

Last modified
3 days ago

Function ARN
arn:aws:lambda:us-west-2:012345678910:function:support-cases-FlaskFunction2-jv0rSx0MF1xs

Application
support-cases

Function URL Info
-

Code | Test | Monitor | Configuration | Aliases | Versions

Test event Info

Delete | Save | Test

Resolve with Ask AWS Beta

Session started with id: sn-0d194533-9227-4386-ae13-a6965416e935

Ask AWS is helping you to resolve the following error

```
[ERROR] ClientError: An error occurred (AccessDeniedException) when calling the PutItem operation: User:
arn:aws:sts::012345678910:assumed-role/support-cases-FlaskFunction2Role-cboFT4oJAS7F/support-cases-
FlaskFunction2-jv0rSx0MF1xs is not authorized to perform: dynamodb:PutItem on resource:
arn:aws:dynamodb:us-west-2:012345678910:table/support-cases because no identity-based policy allows the
dynamodb:PutItem action Traceback (most recent call last):   File "/var/task/app.py", line 93, in
lambda_handler     response=create_case(body)   File "/var/task/app.py", line 41, in create_case
    ddbtable.put_item(Item={   File "/var/task/boto3/resources/factory.py", line 520, in do_action
    response = action(self, *args, **kwargs)   File "/var/task/boto3/resources/action.py", line 83, in
__call__     response = getattr(parent.meta.client, operation_name)(*args, **params)   File
"/var/task/botocore/client.py", line 357, in _api_call     return self._make_api_call(operation_name,
kwargs)   File "/var/task/botocore/client.py", line 676, in _make_api_call     raise
error_class(parsed_response, operation_name)END RequestId: a0af4742-300e-4a2e-88b4-c6ba55110352 REPORT
RequestId: a0af4742-300e-4a2e-88b4-c6ba55110352 Duration: 166.04 ms Billed Duration: 167 ms Memory Size:
128 MB Max Memory Used: 73 MB```
```

Analysis

The error indicates that the Lambda function does not have the necessary permissions to perform a PutItem operation on the DynamoDB table. Specifically, the IAM role attached to the function lacks a policy allowing the dynamodb:PutItem action.

Ask AWS is generating resolution...

Is this analysis helpful?

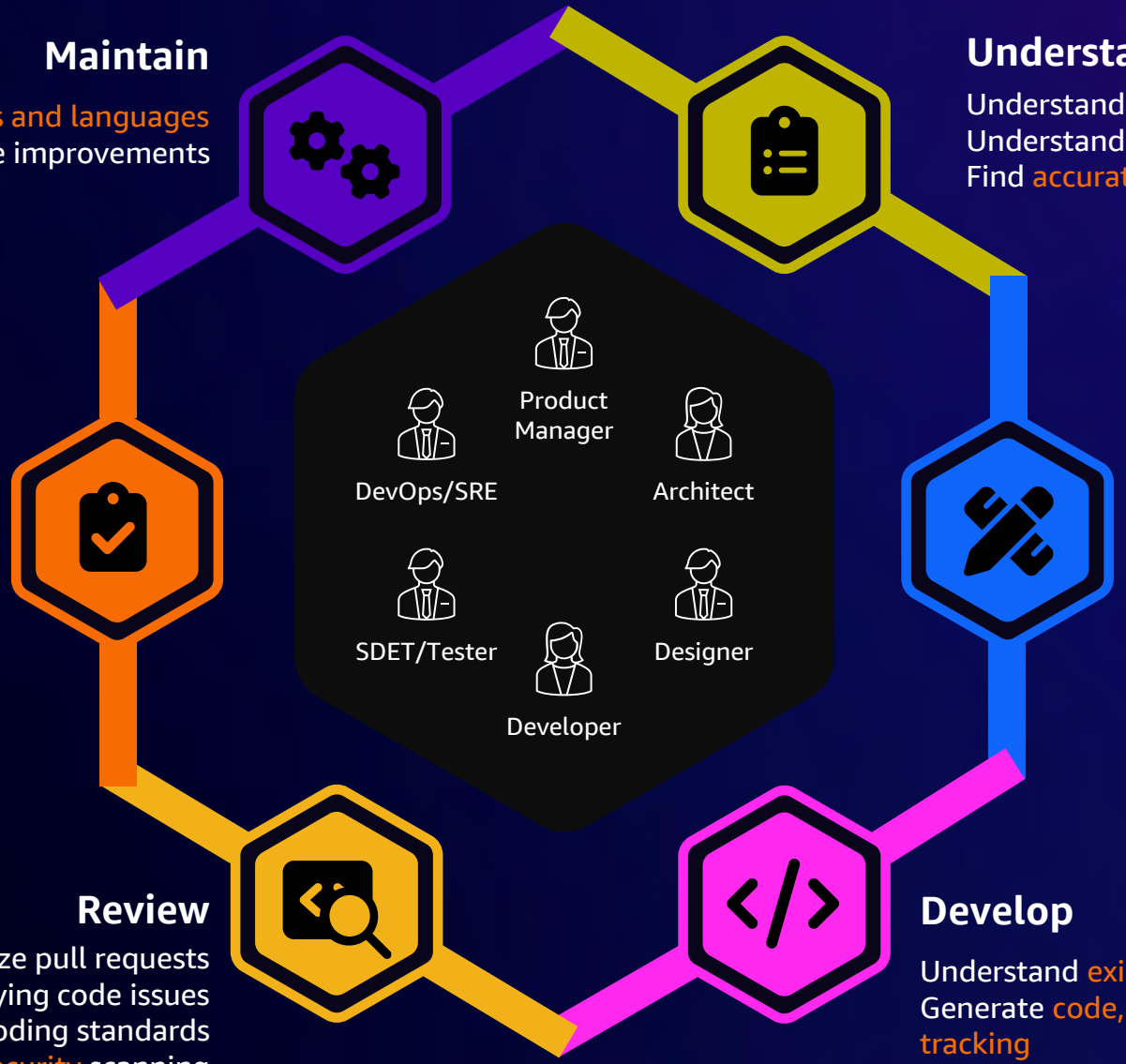
The development companion: Reinventing across the development lifecycle

Maintain

Upgrading dependencies and languages
Analyze business operations to facilitate improvements

Understand & Learn

Understand requirements
Understand existing environment
Find accurate and specific technical guidance



Monitor & Test

Security and performance of an application or cloud infrastructure to see that business needs are met
Track, diagnose and fix errors combined with feedback loops.

Design & Decide

Design application based on business and technical requirements
Decide for an architecture

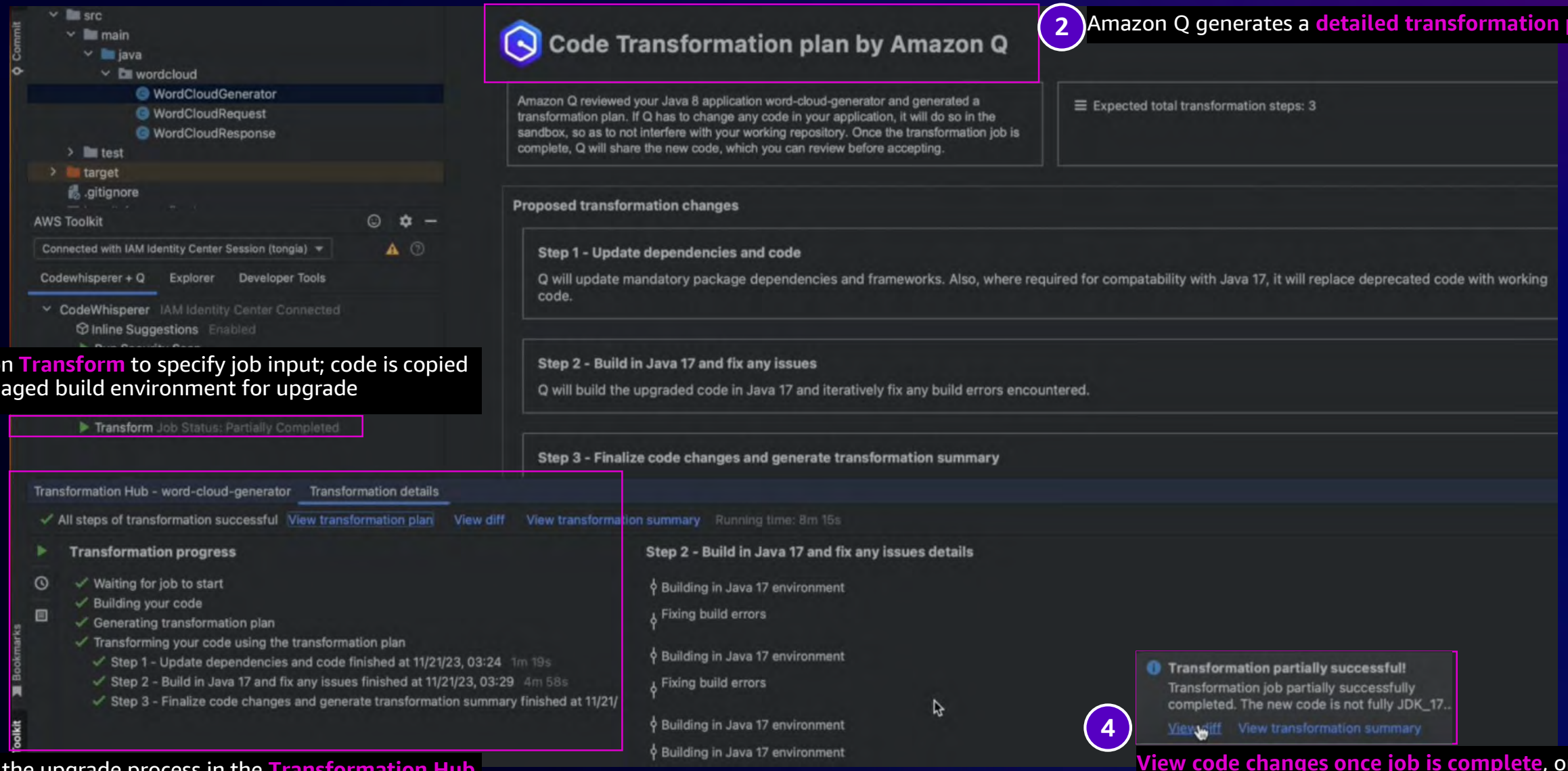
Review

Summarize pull requests
Assist in identifying code issues
Compliant with coding standards
Security scanning

Develop

Understand existing code base
Generate code, open source reference tracking

Code Transformation in the IDE



1 Click on **Transform** to specify job input; code is copied to managed build environment for upgrade

2 Amazon Q generates a **detailed transformation plan**

3 Follow the upgrade process in the **Transformation Hub**

4 **View code changes once job is complete**, or in case of failure view reasons and retry information

- Dev. Experience is likely the best proxy metric for Dev. Productivity
- Tools can help to increase DevEx, but ..Work backwards from your customers/developers: Feedback loops, cognitive load, flow time.
- Platform provide abstraction, but they can be illusions, which you will notice when an issue breaks the abstraction.
- Going from a coding companion to development companion
- Support you though your SDLC
  - From understanding your *internal* documentation and large code bases
  - To develop code or even whole features
  - To testing and troubleshooting
  - And upgrading from old versions or languages to new ones.
  - By having a mentor who is always available to answer any questions you may have.
- Spend less time building the perfect abstraction, instead have a development companion explain it to you.