



**Isabelle Mauny**  
Field Chief Technology Officer



**Steven Murawski**  
Cloud Developer Advocate



# REST API Security by Design with Azure Pipelines

# Security Matters in DevOps

The image is a conceptual illustration for a DevOps security topic. It features a dark purple background with a subtle hexagonal grid pattern. In the foreground, a large, light purple shield is positioned on the left side, symbolizing protection and security. To the right, there are several server racks or server cabinets, rendered in a similar purple hue, representing the infrastructure of a DevOps environment. The overall aesthetic is clean, modern, and tech-oriented.



# THE J-CURVE OF TRANSFORMATION

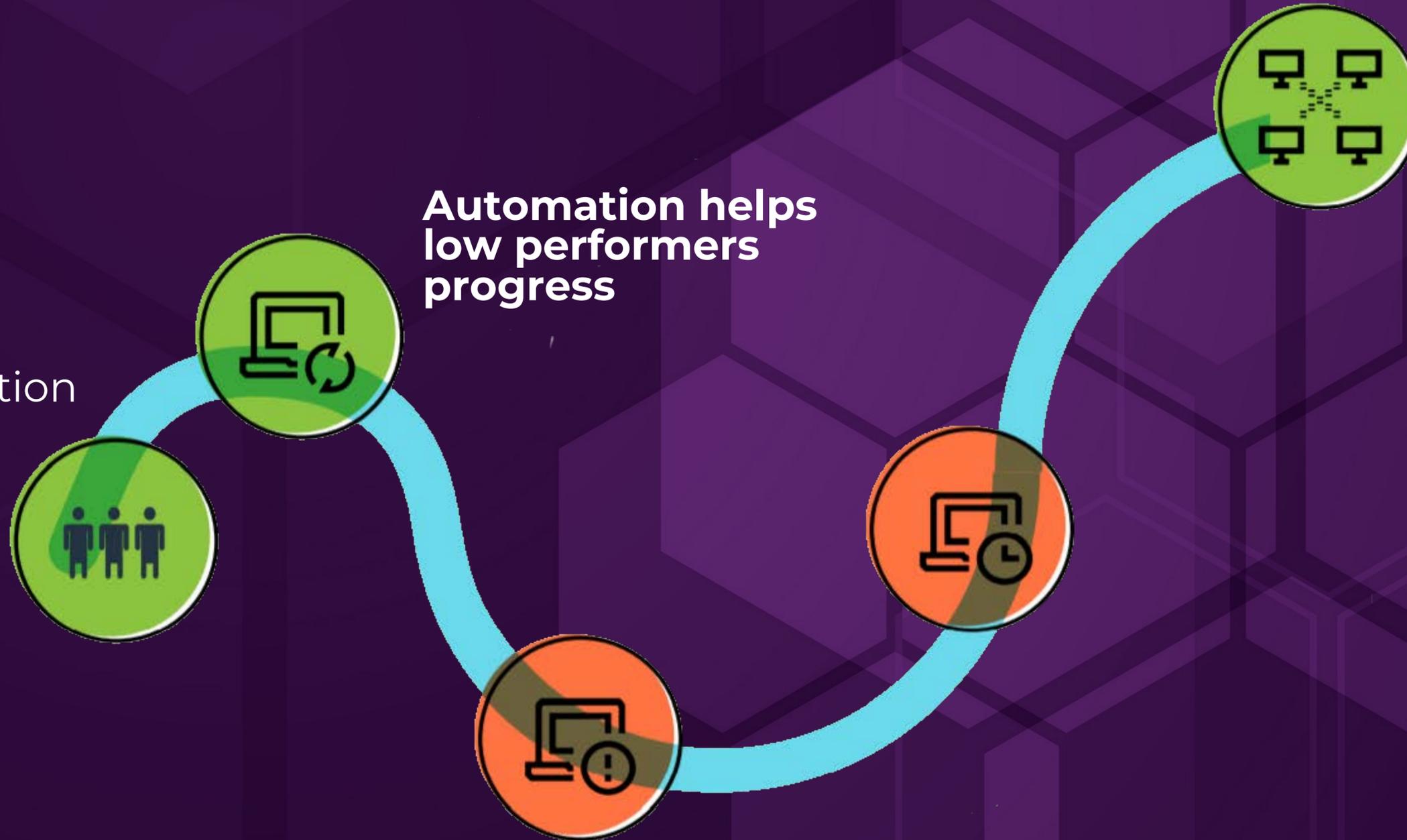
The transformation begins





# THE J-CURVE OF TRANSFORMATION

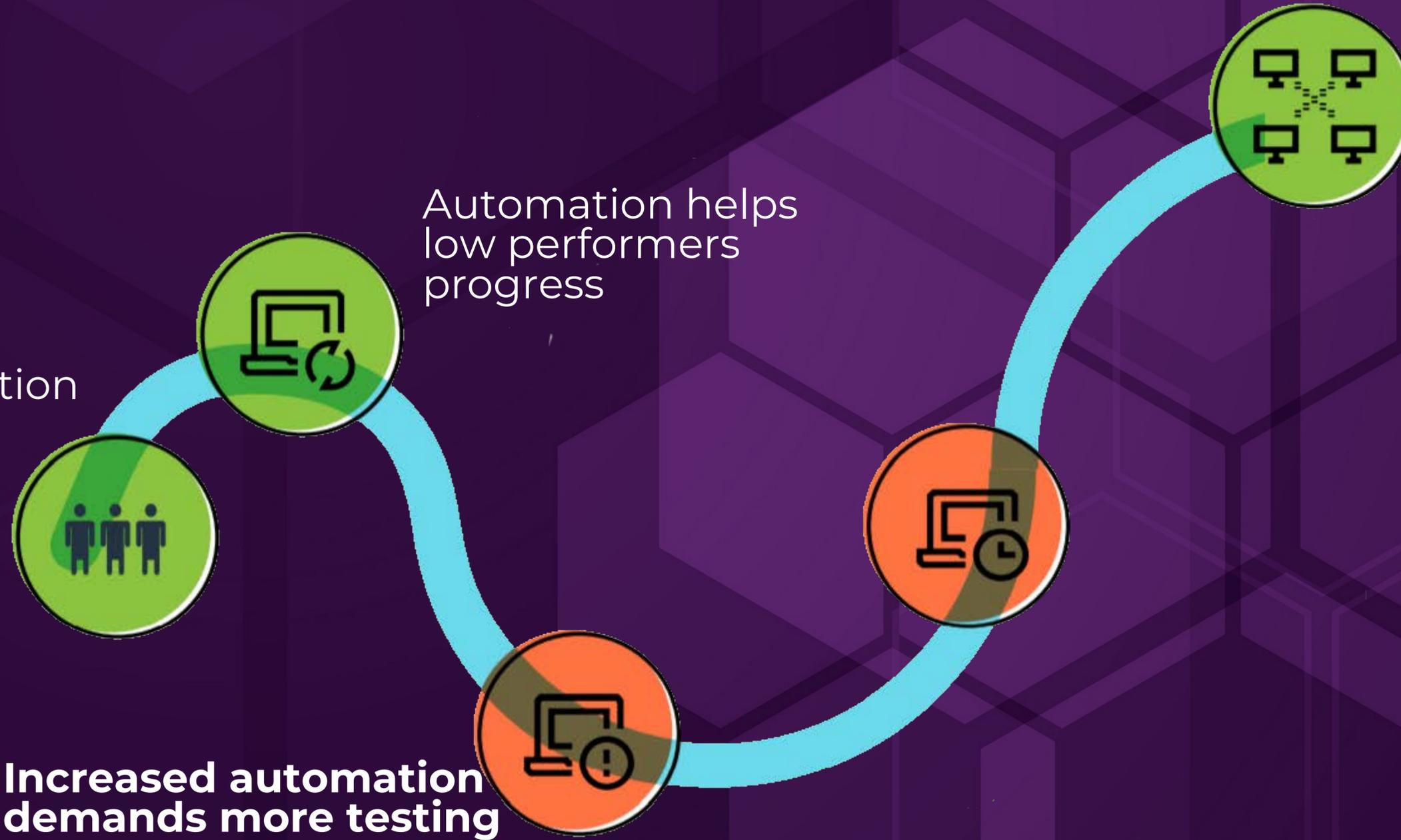
The transformation begins





# THE J-CURVE OF TRANSFORMATION

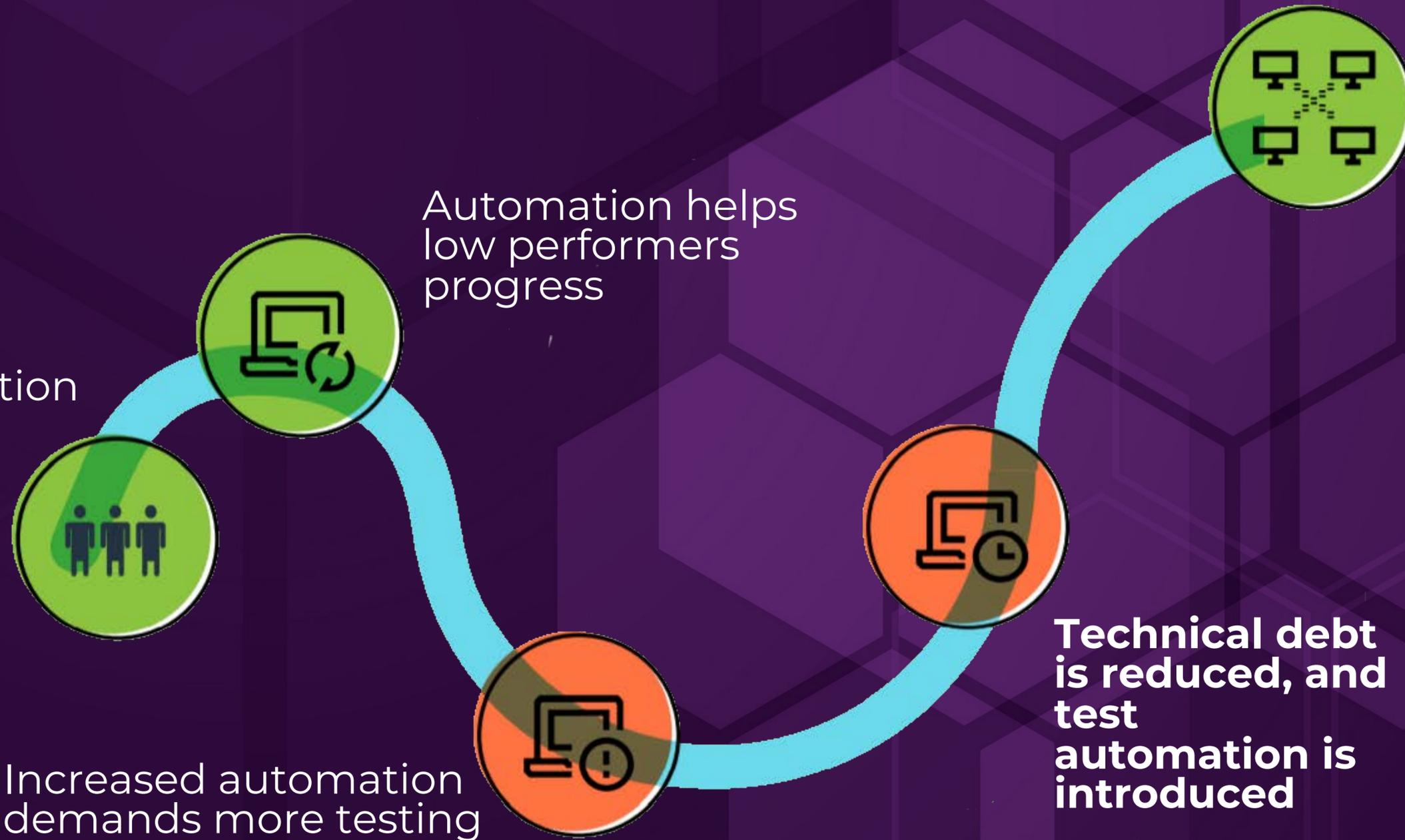
The transformation begins





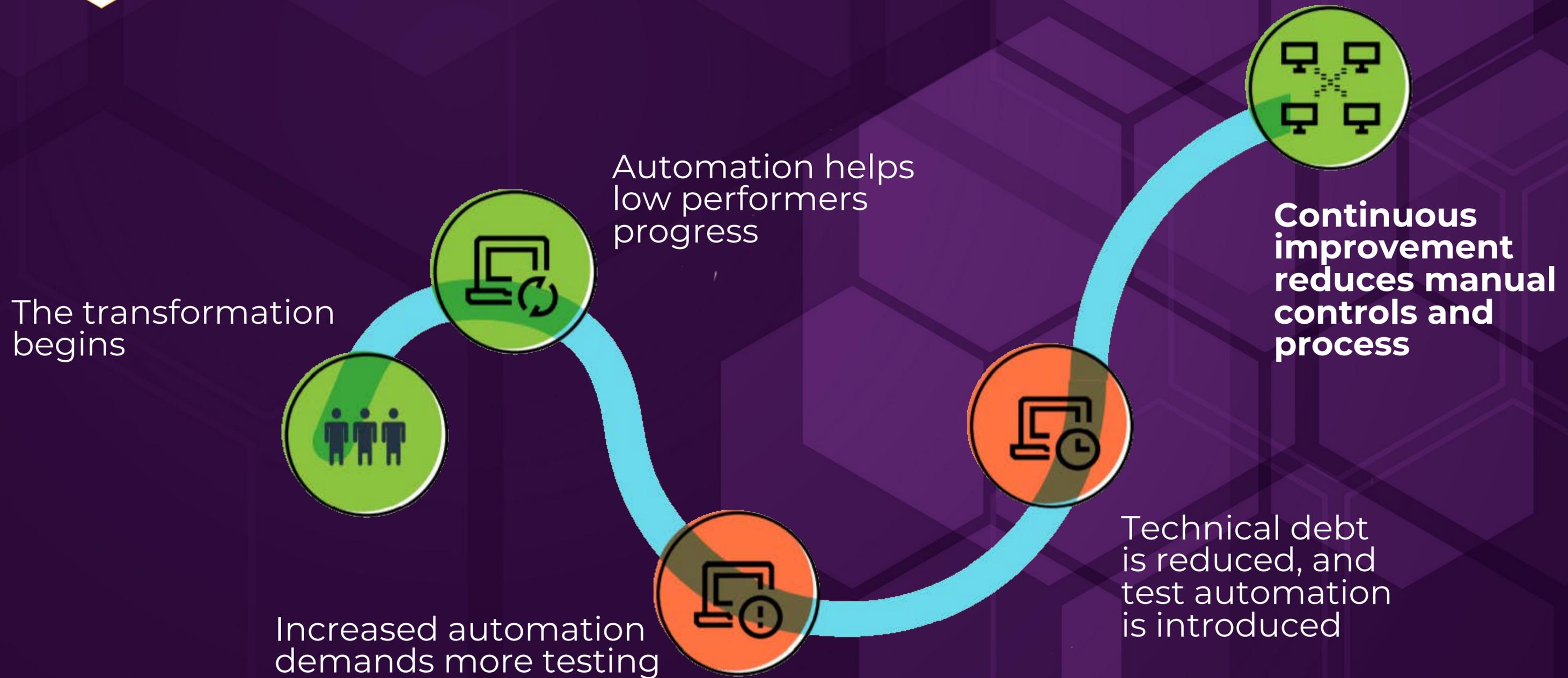
# THE J-CURVE OF TRANSFORMATION

The transformation begins





# THE J-CURVE OF TRANSFORMATION





# CATEGORIES OF PERFORMANCE

## ▶ Low

- ✓ Deploy once a month to once every six months
- ✓ Going from code commit to production can be one to six months
- ✓ Restoring service from an incident can be between one week to one month
- ✓ Approximate change failure rate of 46% to 60%



# CATEGORIES OF PERFORMANCE

## ▶ Medium

- ✓ Deploy once a week to once a month
- ✓ Going from code commit to production can be one week to one month
- ✓ Restoring service from an incident is usually less than a day
- ✓ Approximate change failure rate of 0% to 15%



# CATEGORIES OF PERFORMANCE

## ▸ High

- ✓ Deploy once a day to once a week
- ✓ Going from code commit to production can be one day to one week
- ✓ Restoring service from an incident is usually less than a day
- ✓ Approximate change failure rate of 0% to 15%



# CATEGORIES OF PERFORMANCE

## ▸ Elite

- ✓ Deploy on-demand (multiple deploys a day)
- ✓ Going from code commit to production is less than one day
- ✓ Restoring service from an incident is usually less than a day
- ✓ Approximate change failure rate of 0% to 15%



## AUTOMATION AND INTEGRATION - BUILD

Capability	Low	Medium	High	Elite
Automated build	64%	81%	91%	92%



## AUTOMATION AND INTEGRATION - TESTING

Capability	Low	Medium	High	Elite
Automated build	64%	81%	91%	92%
Automated unit tests	57%	66%	84%	87%
Automated acceptance tests	28%	38%	48%	58%



## AUTOMATION AND INTEGRATION - TESTING

Capability	Low	Medium	High	Elite
Automated build	64%	81%	91%	92%
Automated unit tests	57%	66%	84%	87%
Automated acceptance tests	28%	38%	48%	58%

More on how Microsoft shifts tests left at <https://aka.ms/shift-tests-left>



## AUTOMATION AND INTEGRATION – DEPLOYMENT

<b>Capability</b>	<b>Low</b>	<b>Medium</b>	<b>High</b>	<b>Elite</b>
Automated build	64%	81%	91%	92%
Automated unit tests	57%	66%	84%	87%
Automated acceptance tests	28%	38%	48%	58%
Automated provisioning and deployment to test environments	39%	54%	68%	72%
Automated deployment to production	17%	38%	60%	69%



## AUTOMATION AND INTEGRATION - SECURITY

Capability	Low	Medium	High	Elite
Automated security tests	15%	28%	25%	31%



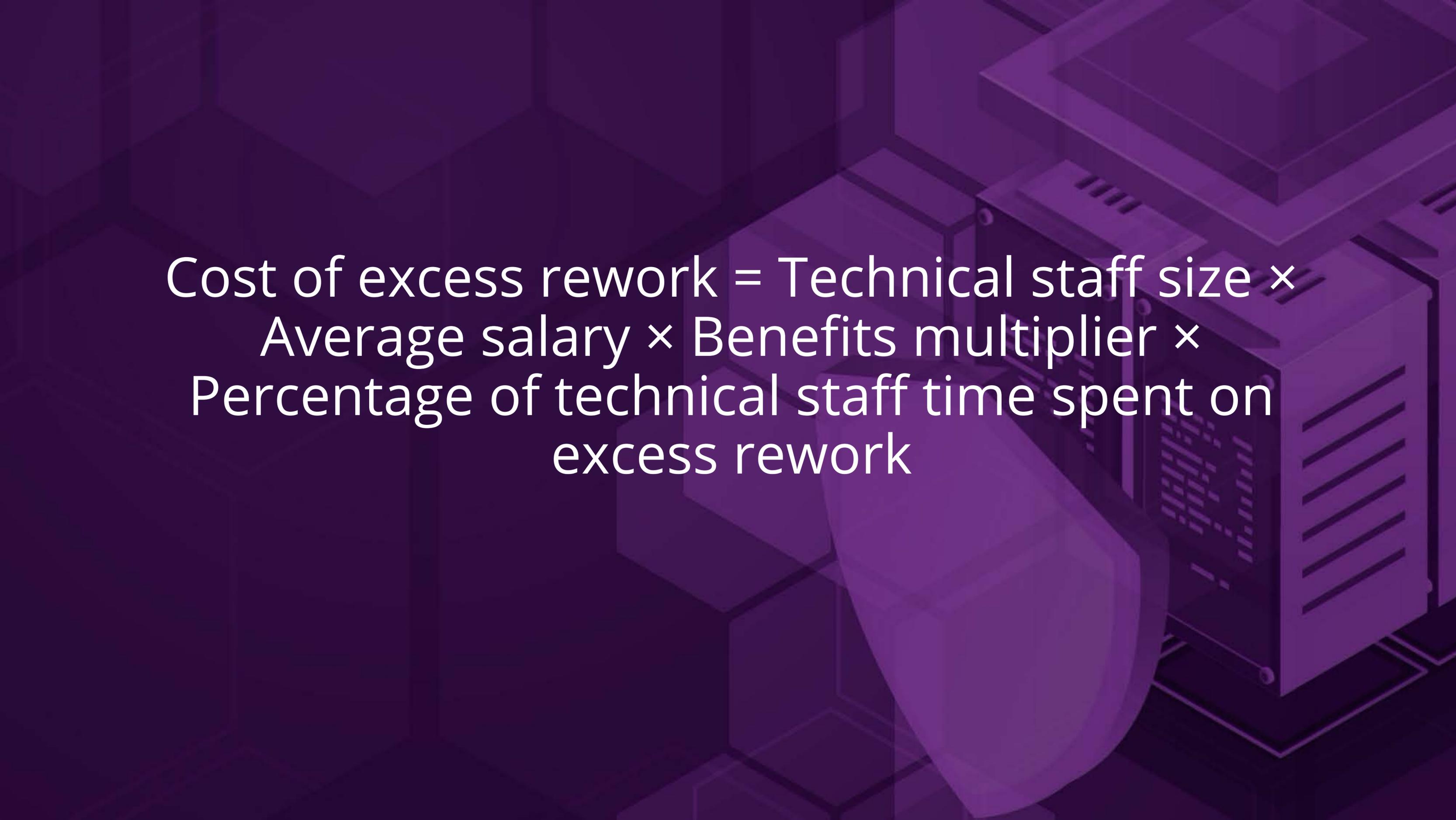
## WHERE IS WORK TIME SPENT

<b>Time Spent</b>	<b>Low</b>	<b>Medium</b>	<b>High</b>	<b>Elite</b>
New work	30%	40%	50%	50%
Unplanned work and rework	20%	20%	20%	19.5%
Remediating security issues	10%	5%	5%	5%
Working on end user reported issues	20%	10%	10%	10%
Customer support work	15%	10%	10%	5%



## WHERE IS WORK TIME SPENT

Time Spent	Low	Medium	High	Elite
New work	30%	40%	50%	50%
Unplanned work and rework	20%	20%	20%	19.5%
<b>Remediating security issues</b>	<b>10%</b>	<b>5%</b>	<b>5%</b>	<b>5%</b>
Working on end user reported issues	20%	10%	10%	10%
Customer support work	15%	10%	10%	5%

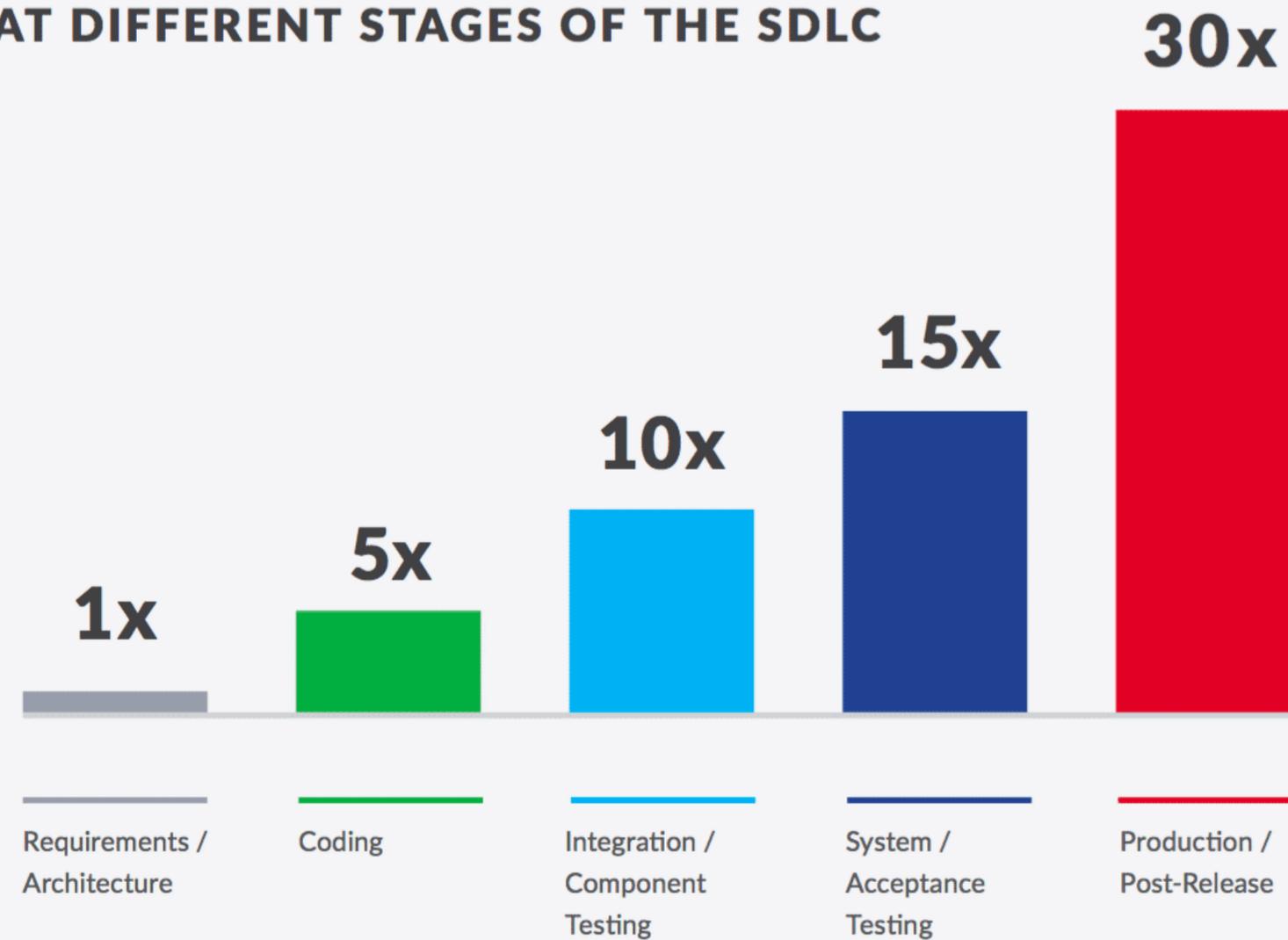


Cost of excess rework = Technical staff size ×  
Average salary × Benefits multiplier ×  
Percentage of technical staff time spent on  
excess rework



# COST OF DEFECTS ALONG THE LIFECYCLE

**THE RELATIVE COST OF FIXING A FLAW  
AT DIFFERENT STAGES OF THE SDLC**



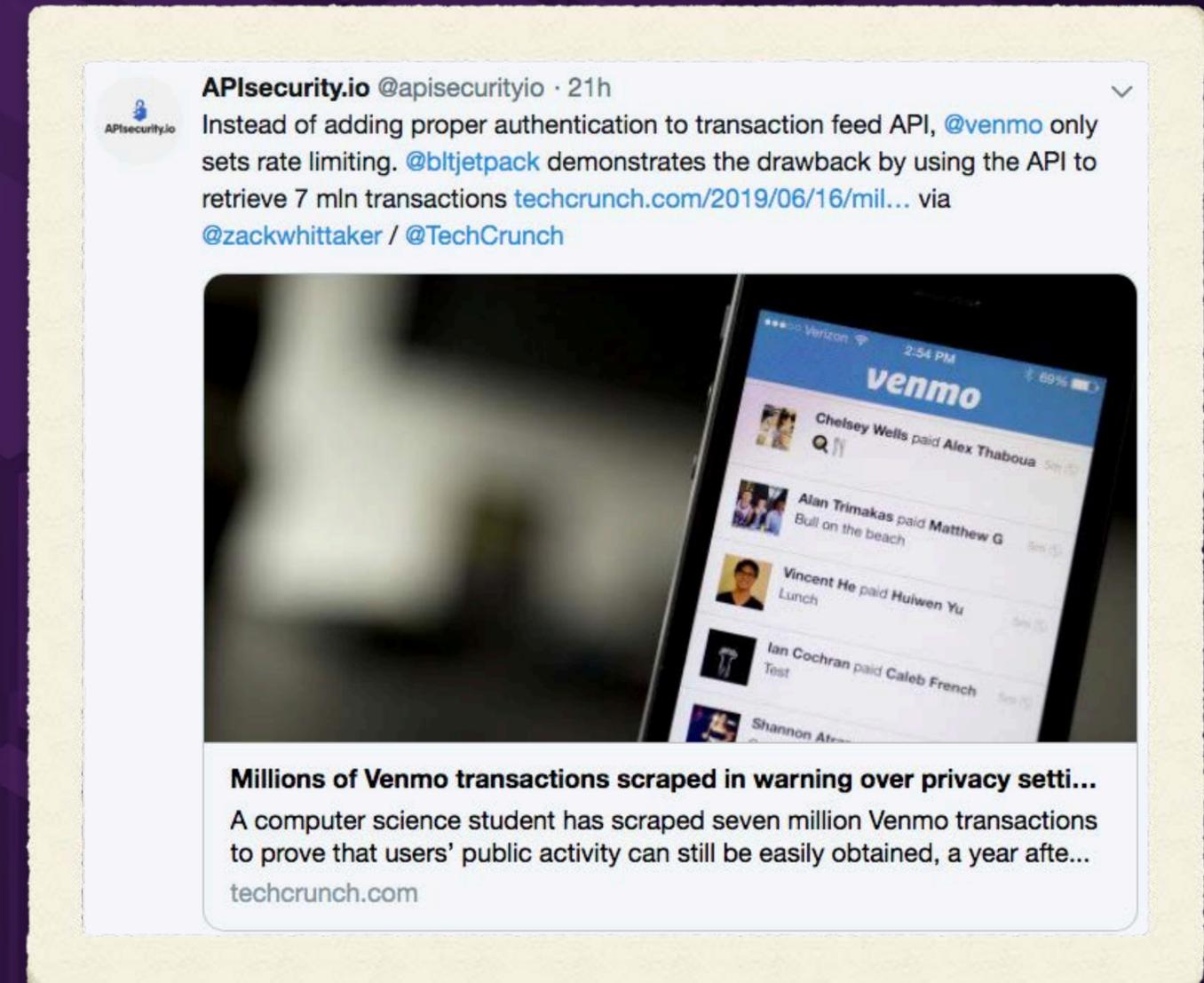


# AUTOMATING API THREAT PROTECTION



# APIS ARE THE NEW ATTACKS VECTOR...

- ▶ Data breaches via APIs are on the rise
  - ✓ 200+ breaches reported on [apisecurity.io](https://apisecurity.io) since Oct. 2018
  - ✓ And those are just the public ones!
- ▶ Most recurrent causes:
  - ✓ Lack of Input validation
  - ✓ Data/Exception leakage
  - ✓ Broken authentication





# 42C MANY APIS, DEPLOYED OFTEN



APPLICATION  
DEVELOPMENT



APPLICATION  
SECURITY



**SECURING APIS  
REQUIRES A NEW  
APPROACH**

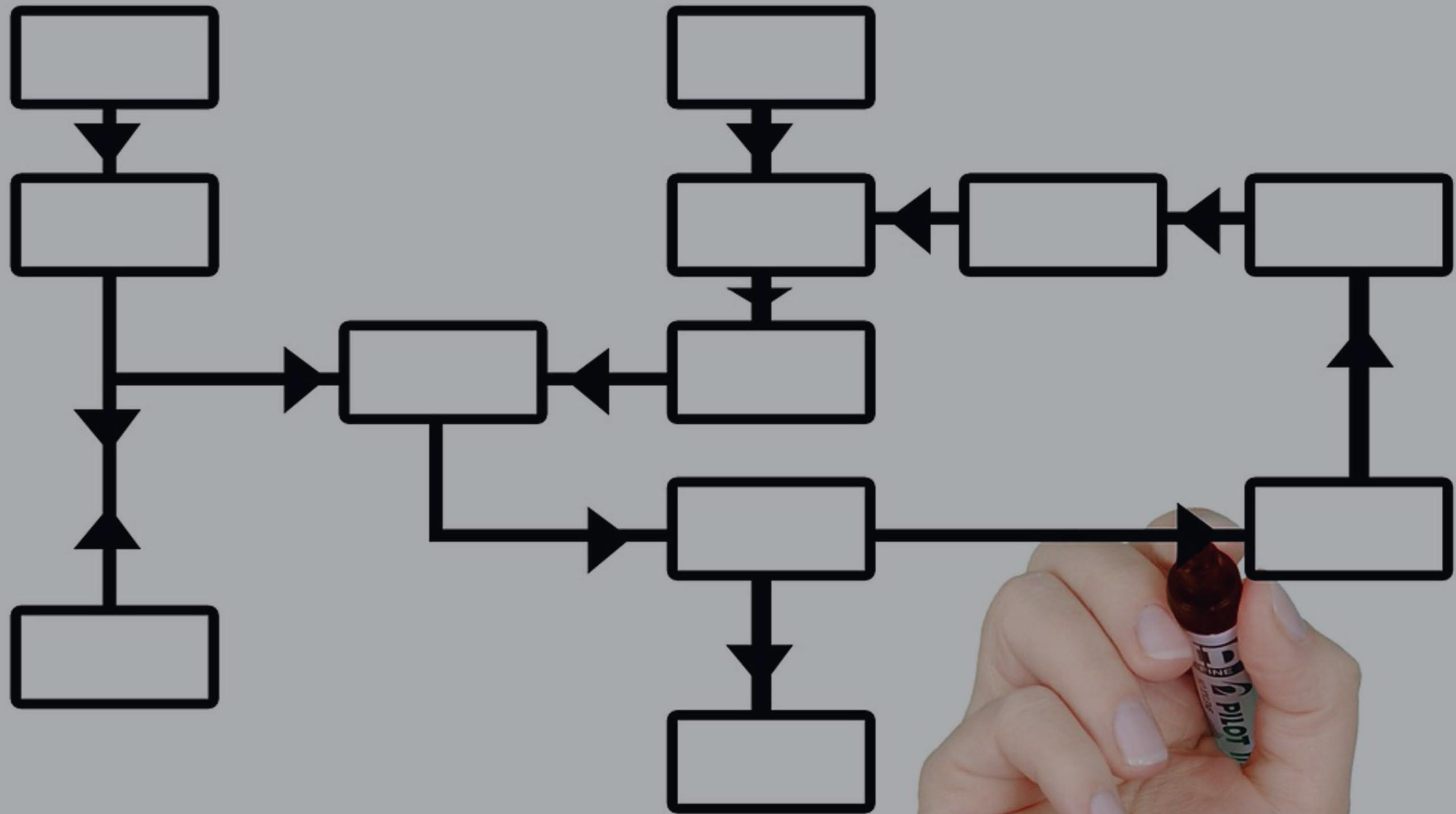


Business

Security

Development

Operations



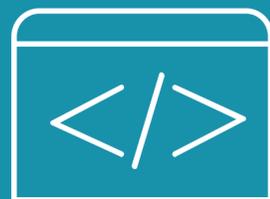




## Design

Developer initiates security work at design time.

Best practices and recommendations are documented.



## Develop

Developer documents the API contract with OpenAPI/Swagger.

API Contract security is evaluated from VSCode using 42Crunch plugin.



## Integrate & Test

API Contract quality is enforced via CI/CD pipeline. Builds are blocked when minimal security requirements defined by security teams are not met.

API implementation is tested via Conformance Scan



## Deploy & Protect

API Firewall is automatically configured from OAS file and deployed in line of traffic.

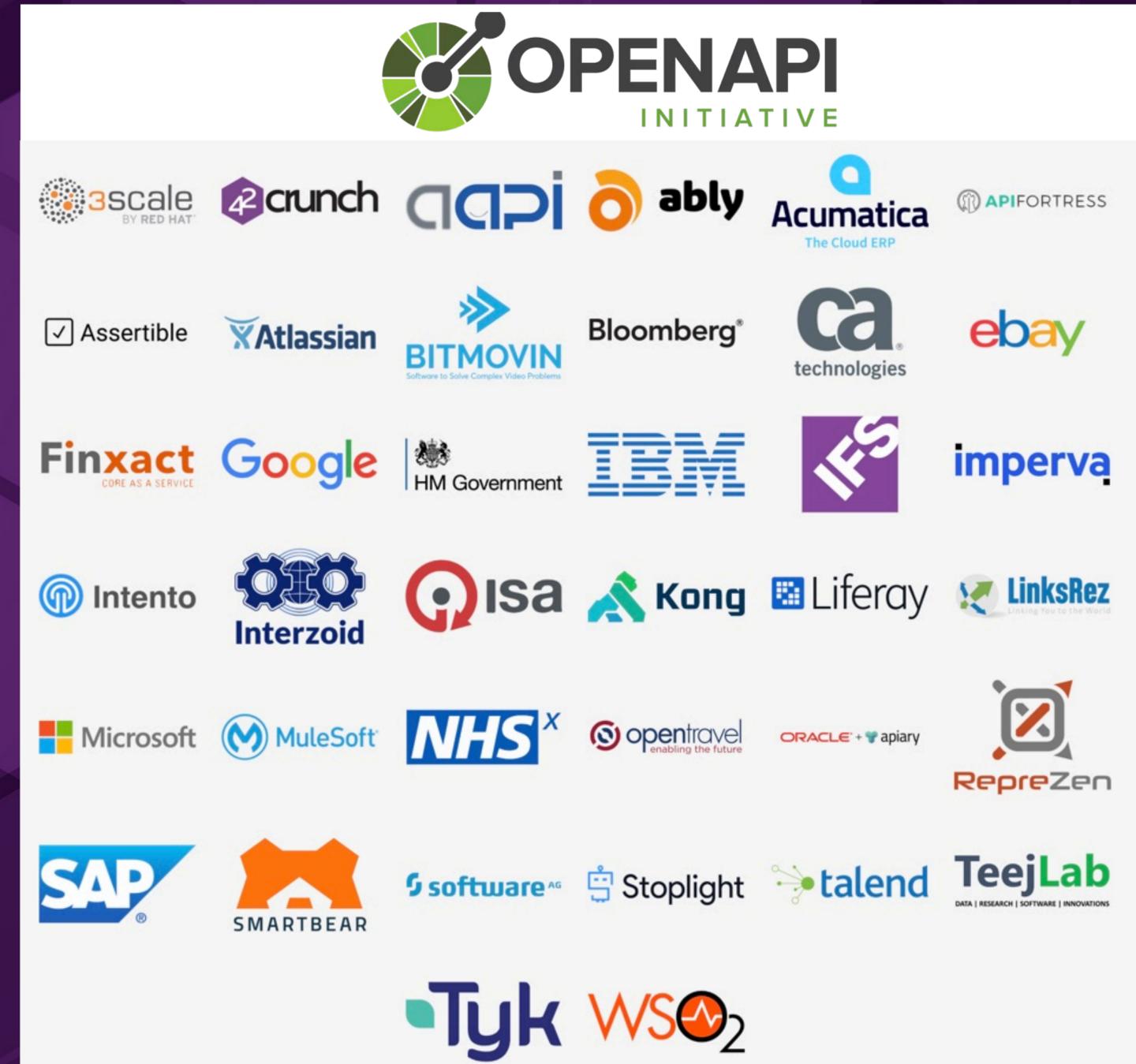
The firewall can be deployed as sidecar in Kubernetes or reverse proxy in front of API Management solutions.





# ENABLING DEVELOPERS TO INITIATE SECURITY

- ▶ Developers know how the application was built!
- ▶ OpenAPI specification is leveraged to describe the API contract.
- ▶ Once the API contract is defined by the developer, the security process becomes clear and straight forward !

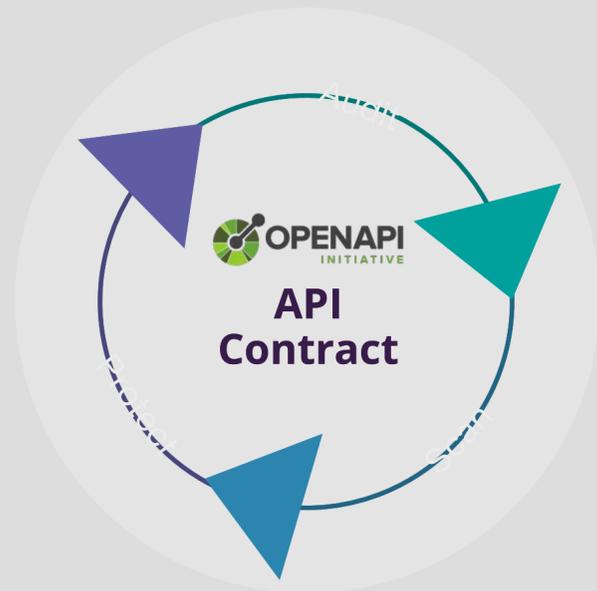


“If you describe your API, we will secure it”

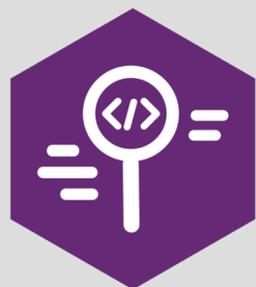
# EMPOWER DEVELOPERS TO BUILD THE ULTIMATE WHITELIST

## VALIDATE OPENAPI CONTRACT CONTENTS

- ▶ Does it comply to best practices ?
- ▶ Does it comply to security requirements ?
  - ✓ Using API Keys ? OAuth ? Basic Auth ?
- ▶ How well is the data defined ?
  - ✓ Headers, query params, path params, form data
  - ✓ Input/output payloads format (JSON)
  - ✓ Is the data constrained ?
    - Min/Max/Patterns/Max Items



## AUDIT



# 42C Platform Architecture

## 42crunch API Security Platform

### 42Crunch Platform Services



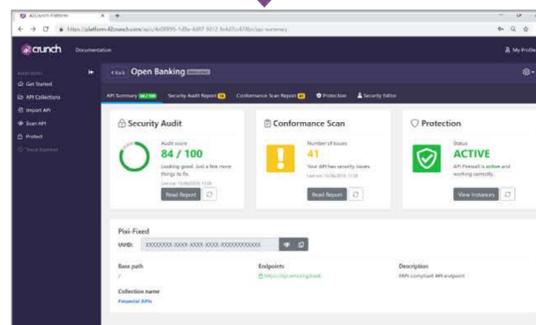
### Platform Reports / Dashboards



### 42Crunch API Firewalls



### REST API



IDEs



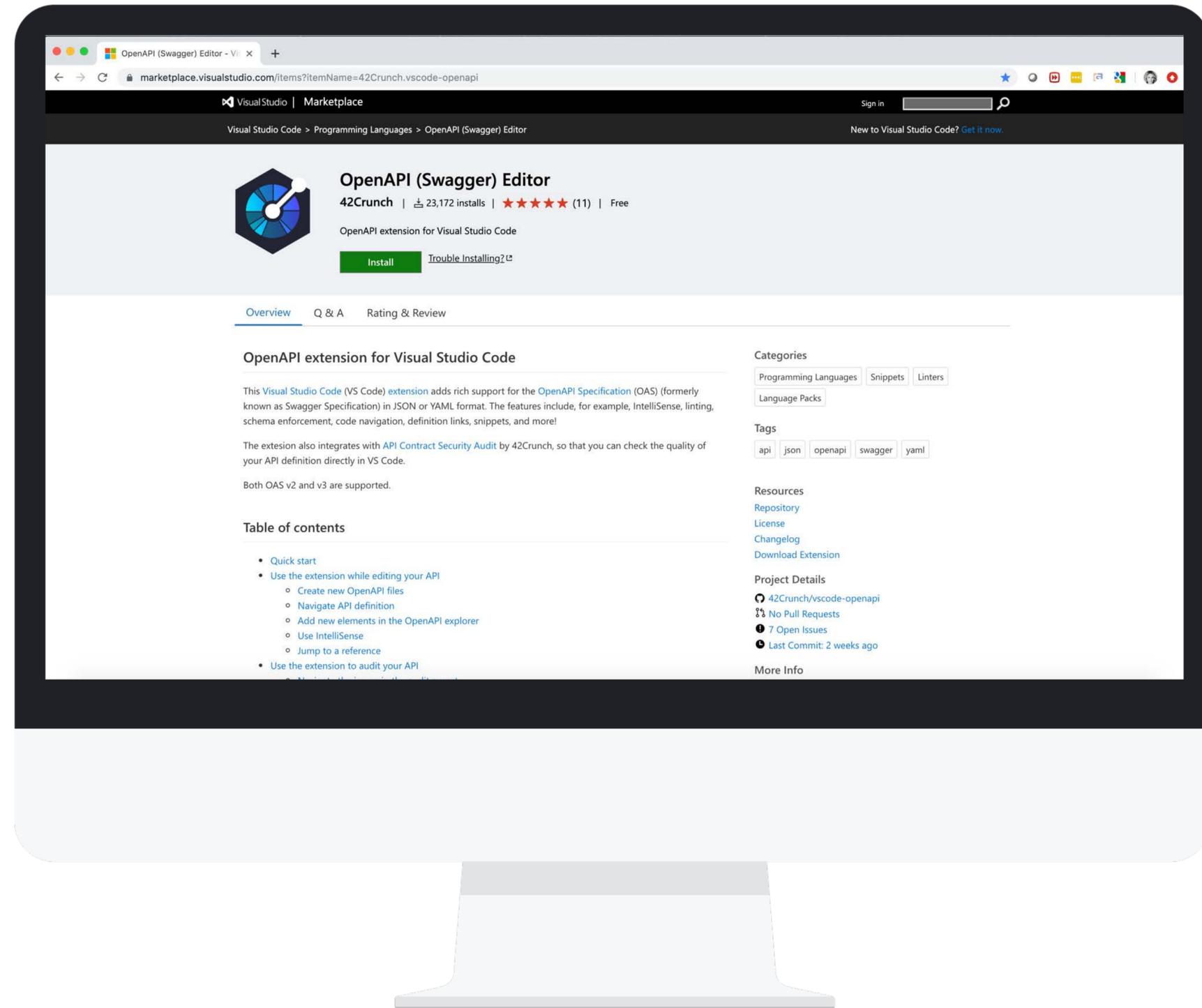
CI /CD Plugins



Scripting

# DEMO PART 1:

# OPENAPI EDITOR/AUDIT FOR VSCODE



<https://marketplace.visualstudio.com/items?itemName=42Crunch.vscod-openapi>

Pixel-Initial.json — OASFiles

OPENAPI

- swagger
- host
- info
- schemes
- produces
- tags
- PATHS
  - > /api/admin/all\_users
  - > /api/admin/users/se...
  - > /api/login
  - > /api/register
  - > /api/user/edit\_info
  - > /api/user/info
- PARAMETERS
- RESPONSES
- DEFINITIONS
  - UserRegistrationData
  - UsersItem**
  - UsersListItem
  - UserUpdateData
- SECURITY
  - access-token
- SECURITY DEFINITIONS
  - access-token

```
{} Pixel-Initial.json > {} definitions > {} UsersItem
502 "UsersItem": {
503   "type": "object",
504   "additionalProperties": false,
505   "properties": {
506     "_id": {
507       "type": "number",
508       "format": "integer",
509       "minimum": 0,
510       "maximum": 999999
511     },
512     "pic": {
513       "type": "string",
514       "format": "uri",
515       "pattern": "(\\w+:(\\/?\\/?)[^\\s]+)",
516       "minLength": 0,
517       "maxLength": 200
518     },
519     "email": {
520       "type": "string",
521       "format": "email",
522       "pattern": "^[a-zA-Z0-9_\\-\\.]+(@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5}))?$",
523       "minLength": 5,
524       "maxLength": 50,
525       "example": "email@email.com"
526     },
527     "password": {
528       "type": "string",
529       "format": "string",
530       "pattern": "[a-zA-Z0-9&@#!?]{4,12}$",
531       "minLength": 4,
532       "maxLength": 12,
533       "example": "p@ssword1"
534     },
535     "name": {
536
```

Ln 502, Col 18 Spaces: 2 UTF-8 LF JSON

Pixi-Initial.json — OASFiles
API Security Audit

EXPLORER

OPEN EDITORS

- GROUP 1
- GROUP 2

OASFILES

- Pixi-Initial.json 9+
- PixiBasic-forScan.json
- PixiBasic-v1.0.json

OUTLINE

BALLERINA PROJECT OV...

```

416     },
417     "required": [
418       "message"
419     ]
420   }
421 }
422 },
423 "security": [
424   {
425     "access-token": []
426   }
427 ],
428 "operationId": "userSearch"
429 }
430 },
431 "/api/admin/all_users": {
432   "get": {
433     "responses": {
434       "200": {
435         "description": "",
436         "schema": {
437           "type": "array",
438           "minItems": 1,
439         }
440       },
441       "403": {
442         "description": "No token provided or i
443       },
444       "default": {
445         "description": "unexpected error",
446         "schema": {
447           "type": "object"
448         }
449       }
450     }
451   }
452 }

```

## Array schema has no maximum number of items defined

Line 436. Severity: High. Score impact 0.5

### Description

An array schema does not specify the maximum number of items it can contain.

For more details, see the [OpenAPI Specification](#).

### Example

The following is an example of how this type of risk could look in your API definition:

```

"post": {
  "description": "Creates a new pet in the store",
  "operationId": "addPet",
  "parameters": [
    {
      "name": "pet",
      "in": "body",
      "description": "Pet to add to the store",
      "required": true,
      "schema": {
        "type": "object",
        "additionalProperties": "false",
        "required": [
          "name"
        ],
        "properties": {
          "name": {
            "type": "string"
          }
        }
      }
    }
  ]
}

```

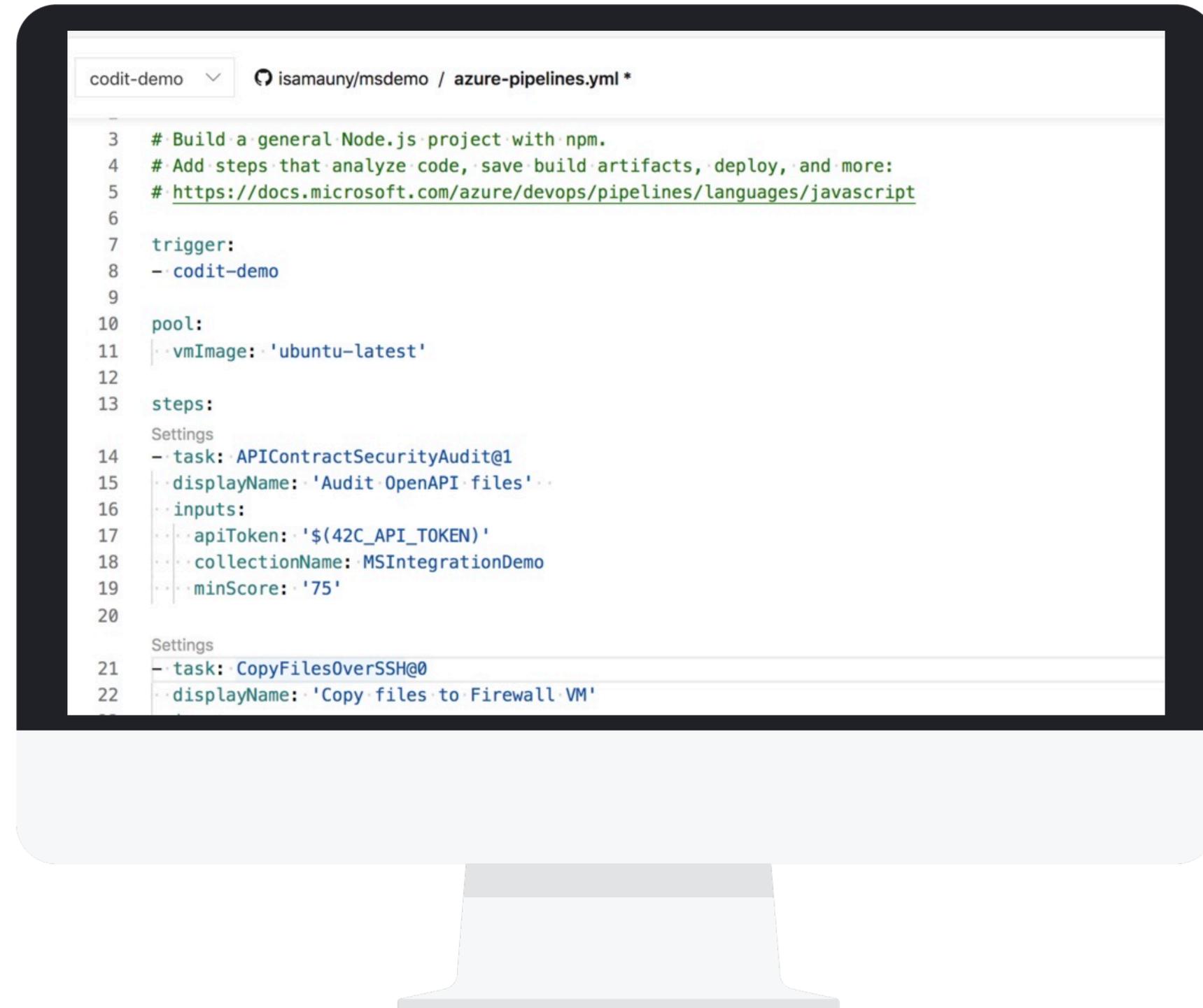
View detailed report for 1 OpenAPI issue(s)

Response that should contain a body has no schema defined (score impact 0.2)

Peek Problem No quick fixes available

Ln 436, Col 16 Spaces: 2 UTF-8 LF JSON

# DEMO STEP 2: AZURE DEVOPS INTEGRATION



```
codit-demo  isamauny/msdemo / azure-pipelines.yml *  
3 # Build a general Node.js project with npm.  
4 # Add steps that analyze code, save build artifacts, deploy, and more:  
5 # https://docs.microsoft.com/azure/devops/pipelines/languages/javascript  
6  
7 trigger:  
8 - codit-demo  
9  
10 pool:  
11   vmImage: 'ubuntu-latest'  
12  
13 steps:  
14   Settings  
15   - task: APIContractSecurityAudit@1  
16     displayName: 'Audit OpenAPI files'  
17     inputs:  
18       apiToken: '$(42C_API_TOKEN)'  
19       collectionName: MSIntegrationDemo  
20       minScore: '75'  
21  
22   Settings  
23   - task: CopyFilesOverSSH@0  
24     displayName: 'Copy files to Firewall VM'
```

42css / APIMIntegration / Pipelines / Builds / msdemo deployment / YAML

Search

msdemo deployment

Variables Save

codit-demo isamauny/msdemo / azure-pipelines.yml \*

```
3 # Build a general Node.js project with npm.
4 # Add steps that analyze code, save build artifacts, deploy, and more:
5 # https://docs.microsoft.com/azure/devops/pipelines/languages/javascript
6
7 trigger:
8   - codit-demo
9
10 pool:
11   vmImage: 'ubuntu-latest'
12
13 steps:
14   Settings
15   - task: APIContractSecurityAudit@1
16     displayName: 'Audit OpenAPI files'
17     inputs:
18       apiToken: '$(42C_API_TOKEN)'
19       collectionName: MSIntegrationDemo
20       minScore: '75'
21
22   Settings
23   - task: CopyFilesOverSSH@0
24     displayName: 'Copy files to Firewall VM'
25     inputs:
26       sshEndpoint: '42C-VM-France'
```

Tasks

42crunch

42Crunch API Contract Security Audit  
Security Audit discovers your OpenAPI files, an...

## Automated audit and API discovery

<https://marketplace.visualstudio.com/items?itemName=42Crunch.cicd>

# DEV-SEC-OPS BENEFITS

When API security becomes **fully part** of the **API lifecycle**:

- Security is applied **automatically** and at **scale**
- Vulnerable APIs are detected early
- APIs are **automatically protected** as soon as the contract is defined



# RESOURCES

- [42Crunch Website](#)
- [Azure DevOps SignUp](#)
- [Free OAS Security Audit](#)
- [OpenAPI VS Code Extension](#)
- [OpenAPI Spec Encyclopedia](#)
- [OWASP API Security Top 10](#)
- [APIsecurity.io](#)

