Isabelle Mauny
Field Chief Technology Officer
42crunch

Steven Murawski
Cloud Developer Advocate
Microsoft

REST API Security by Design with Azure Pipelines

Security Matters in DevOps

THE J-CURVE OF TRANSFORMATION

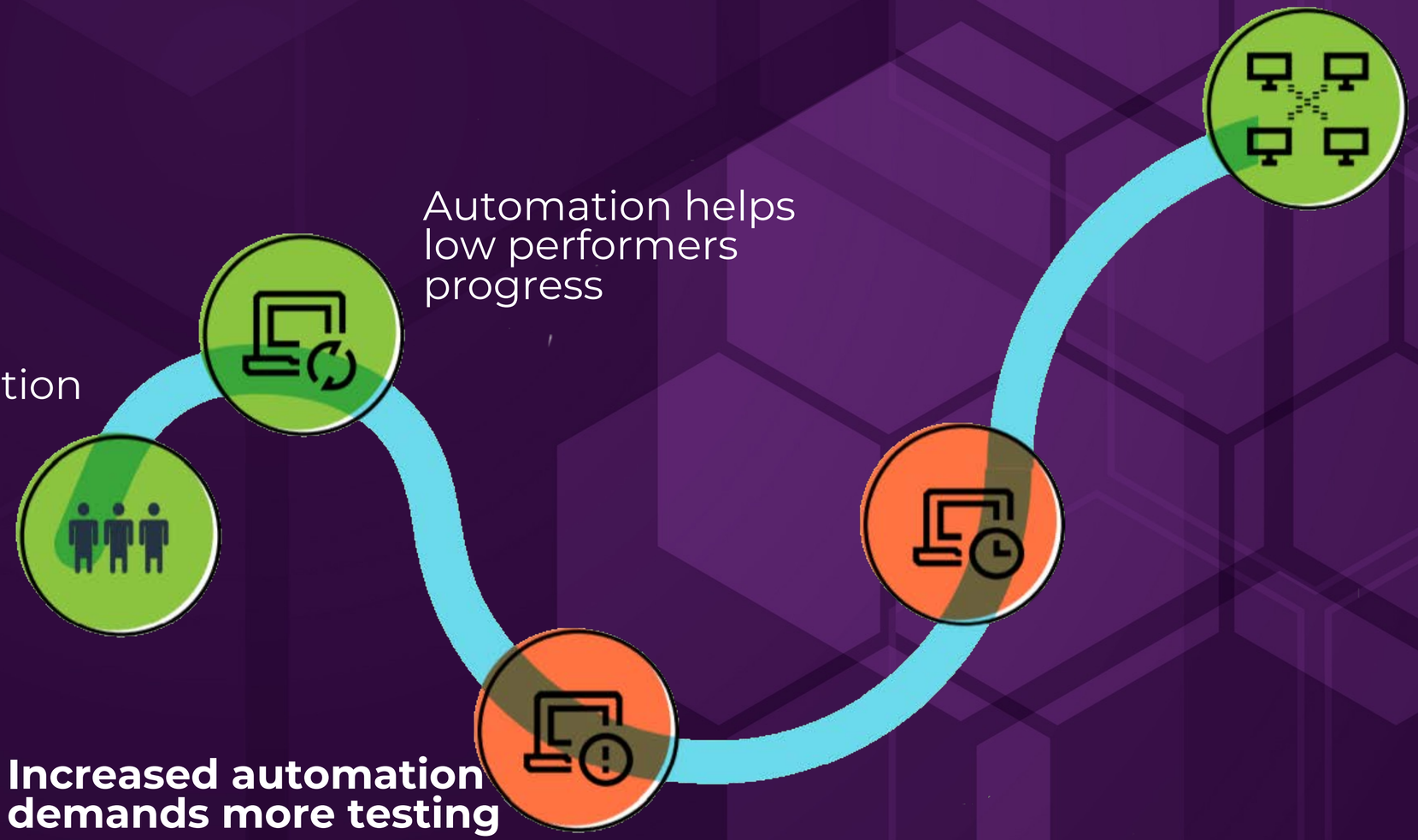
The transformation begins

From the State of DevOps 2018 Report by DORA

THE J-CURVE OF TRANSFORMATION

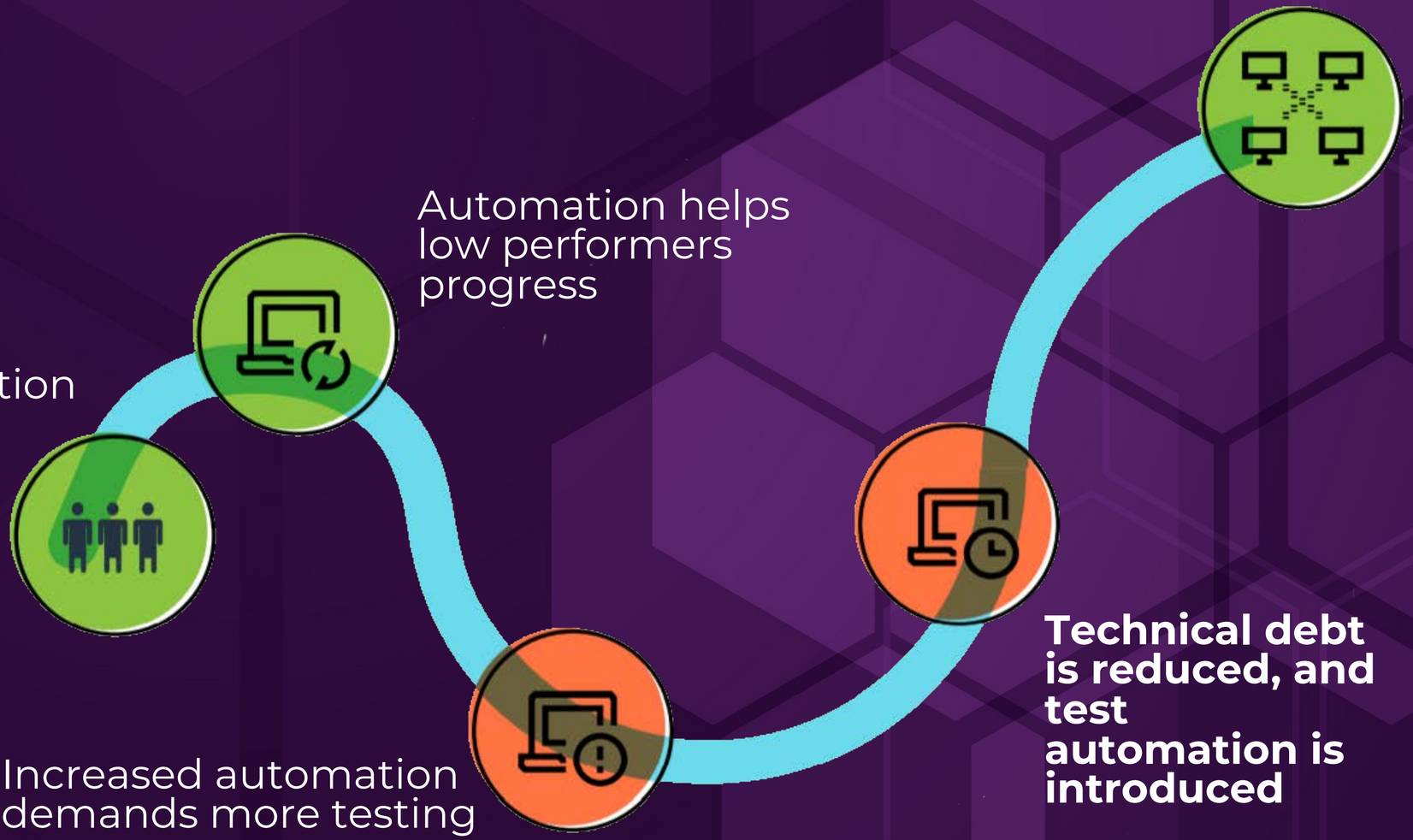
From the State of DevOps 2018 Report by DORA

THE J-CURVE OF TRANSFORMATION



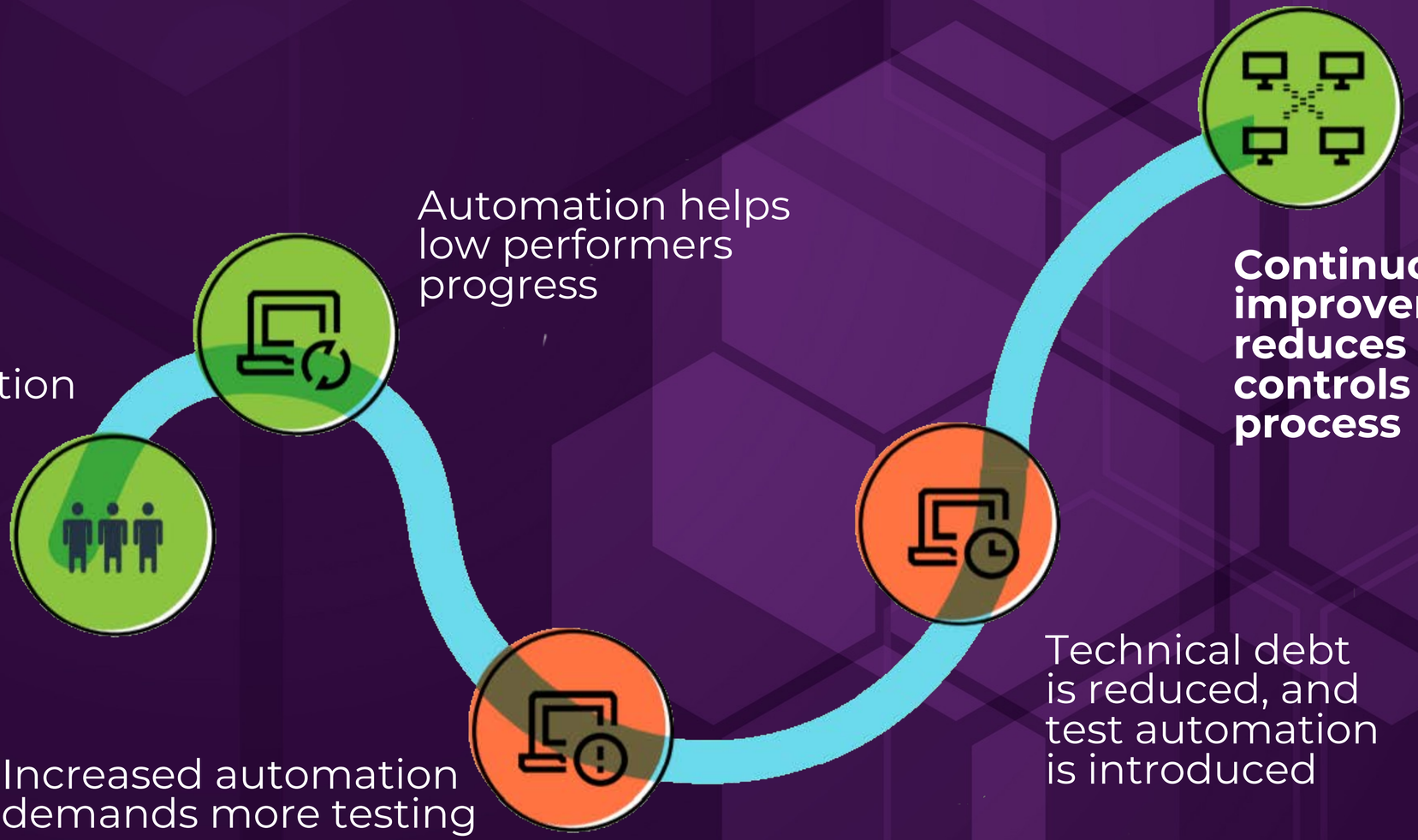
From the State of DevOps 2018 Report by DORA

THE J-CURVE OF TRANSFORMATION



From the State of DevOps 2018 Report by DORA

THE J-CURVE OF TRANSFORMATION



From the State of DevOps 2018 Report by DORA

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%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

AUTOMATION AND INTEGRATION - BUILD

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

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

AUTOMATION AND INTEGRATION – DEPLOYMENT

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%
Automated provisioning and deployment to test environments	39%	54%	68%	72%
Automated deployment to production	17%	38%	60%	69%

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

AUTOMATION AND INTEGRATION - SECURITY

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

From the State of DevOps 2019 Report by DORA – https://aka.ms/2019-state-of-devops

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%
Remediating security issues	10%	5%	5%	5%
Working on end user reported issues	20%	10%	10%	10%
Customer support work	15%	10%	10%	5%

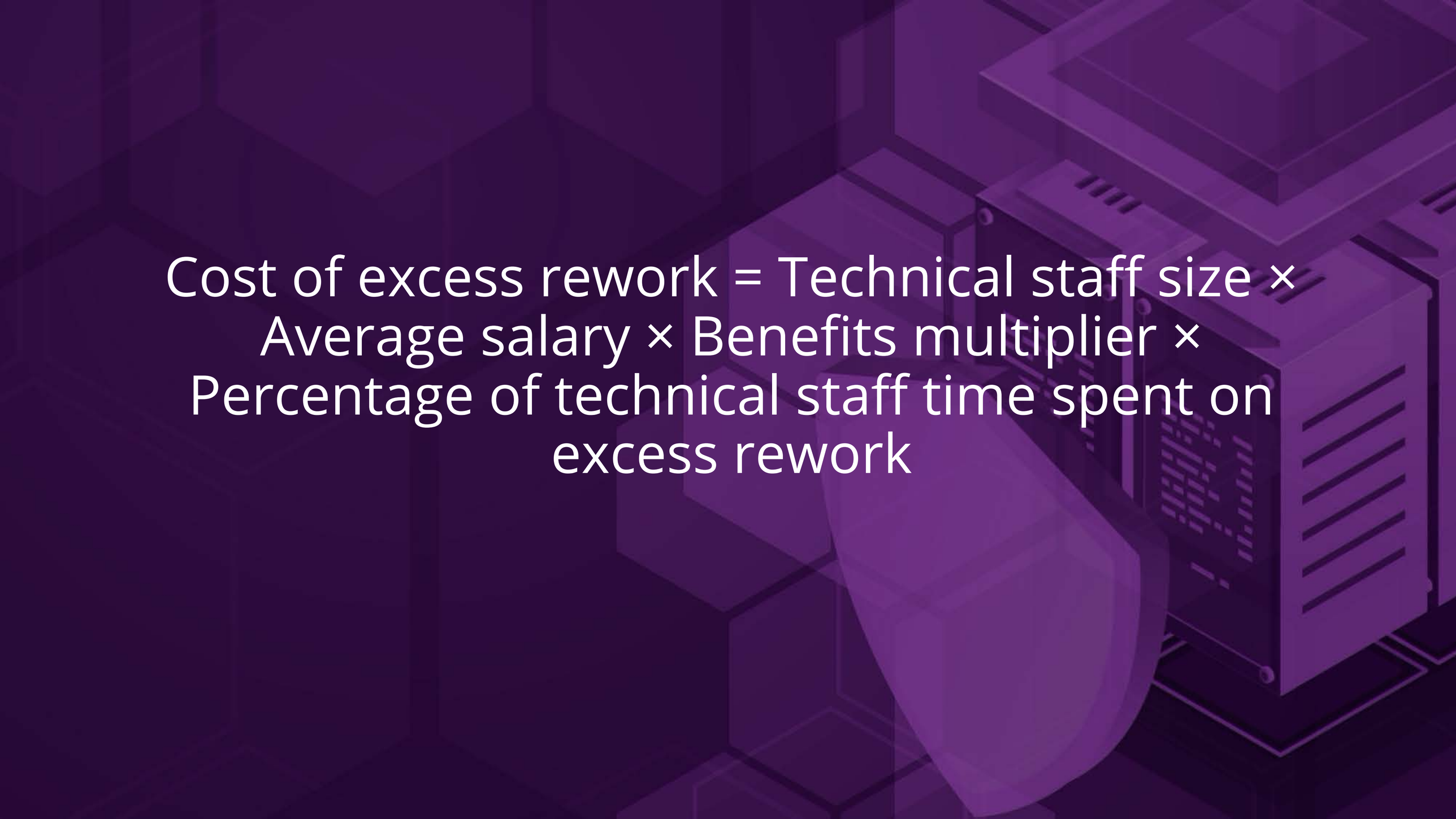
From the State of DevOps 2018 Report by DORA

WHERE IS WORK TIME SPENT

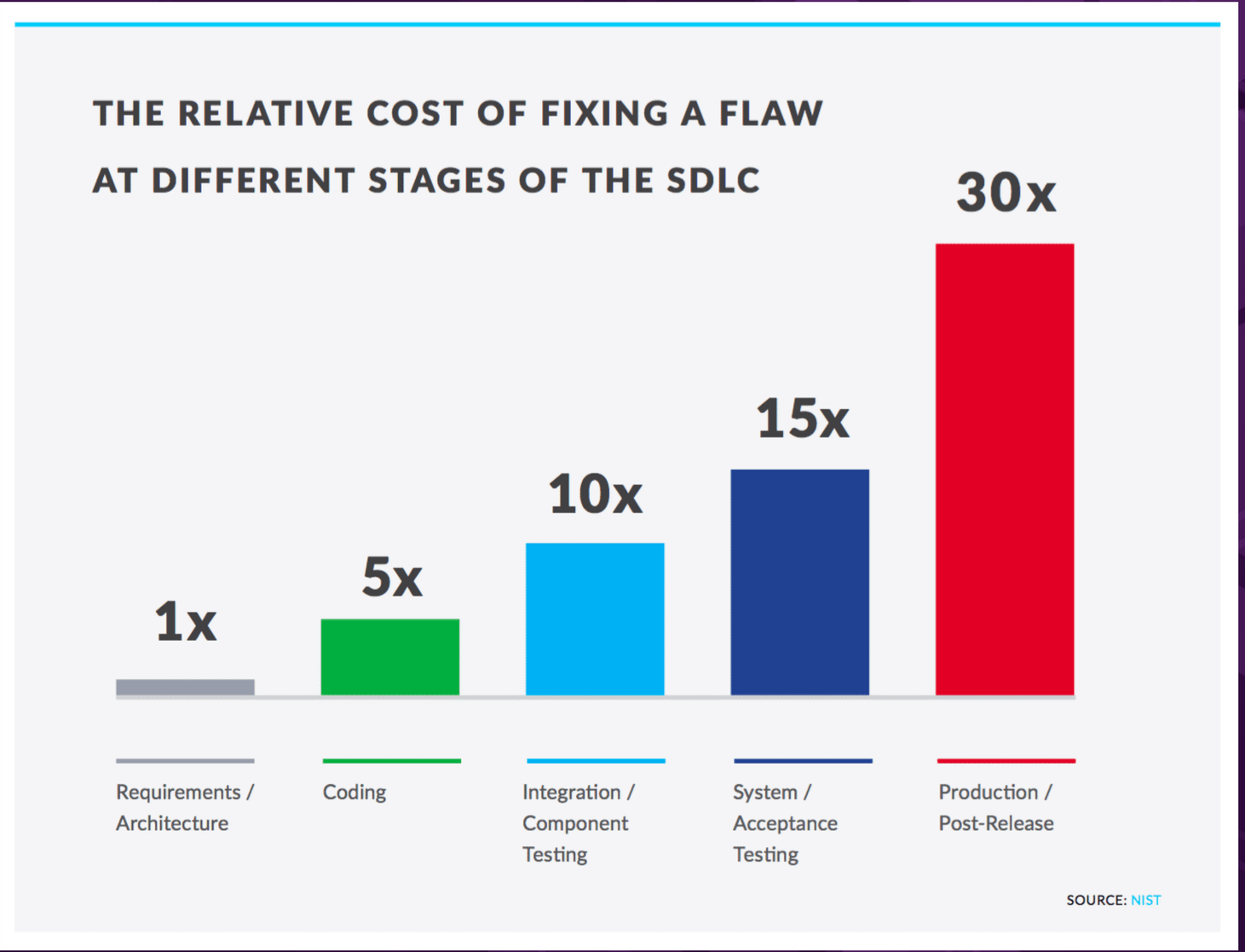
Time Spent	Low	Medium	High	Elite
New work	30%	40%	50%	50%
Unplanned work and rework	20%	20%	20%	19.5%
Remediating security issues	**10%**	**5%**	**5%**	**5%**
Working on end user reported issues	20%	10%	10%	10%
Customer support work	15%	10%	10%	5%

From the State of DevOps 2018 Report by DORA

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



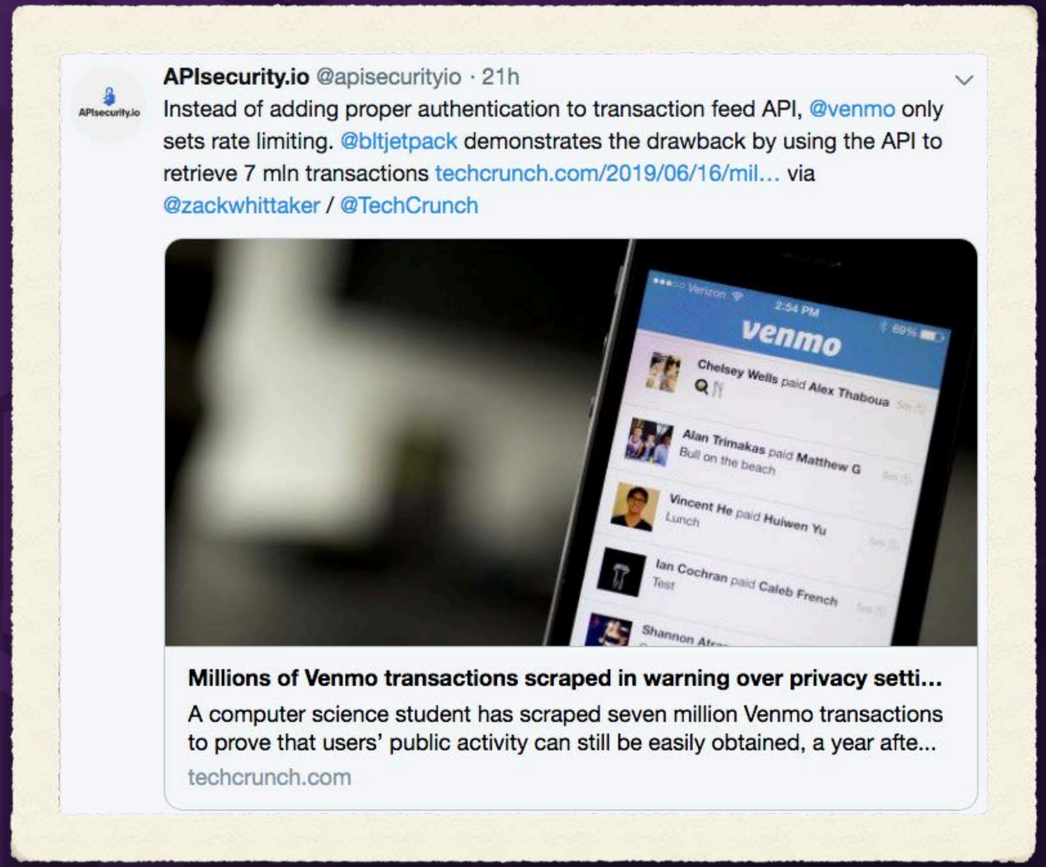
COST OF DEFECTS ALONG THE LIFECYCLE




AUTOMATING API
THREAT PROTECTION

APIS ARE THE NEW ATTACKS VECTOR…

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

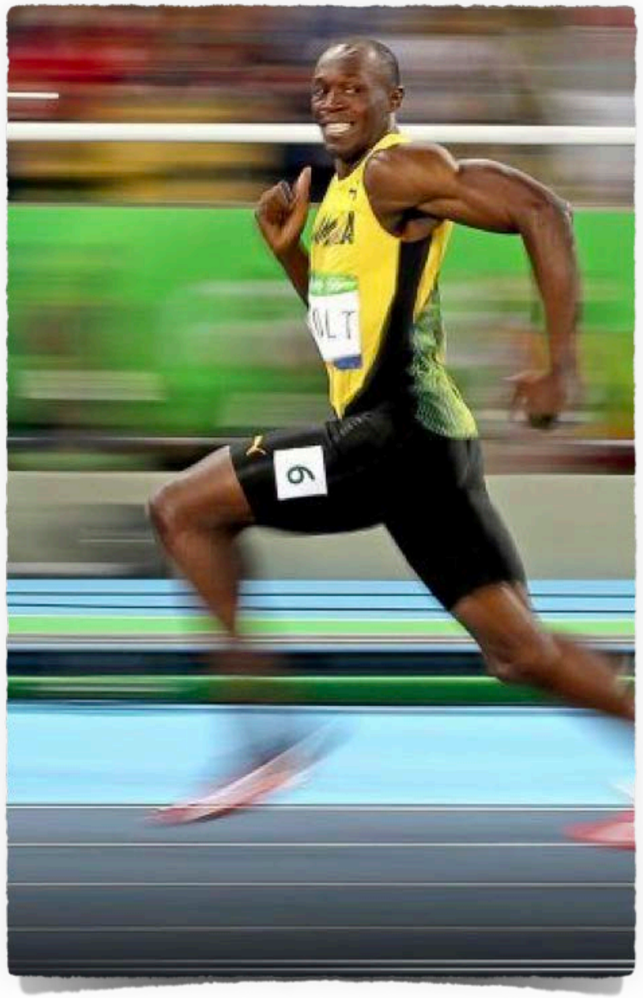


“By 2022, APIs will become the #1 attack vector.”

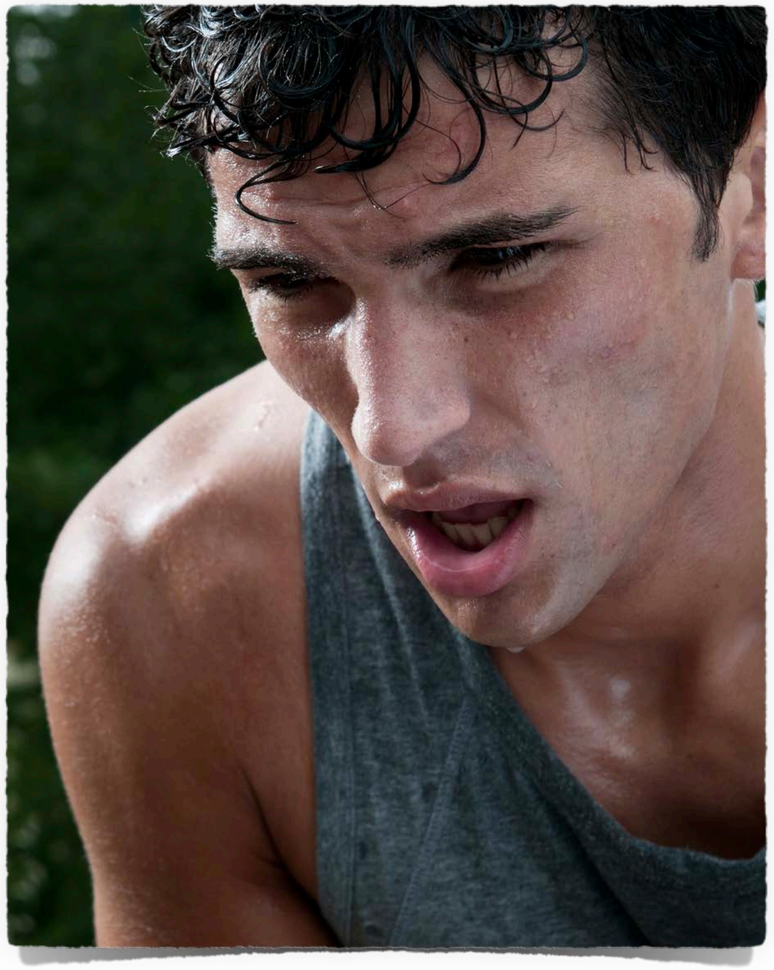
- Gartner, How to Build an Effective API Security Strategy -

*□□□□□://□□□.□□□□□□□□.□□□/□□/□□□□□□□□□□/3834704/□□□-□□-□□□□□-□□-□□□□□□□□□-□□□-□□□□□□□□-□□□□□□□□

MANY APIS, DEPLOYED OFTEN



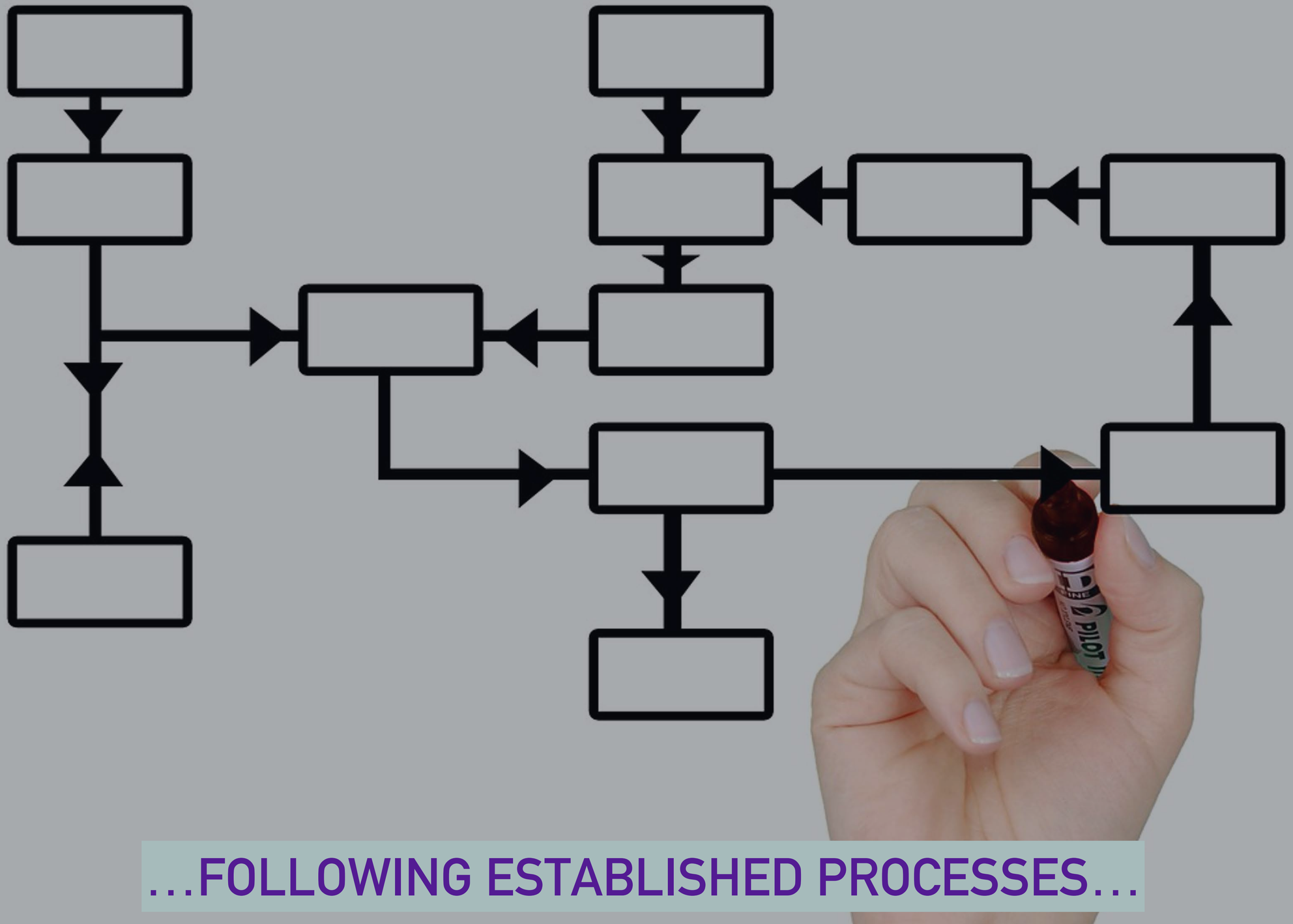
APPLICATION
DEVELOPMENT



APPLICATION
SECURITY


SECURING APIS
REQUIRES A NEW
APPROACH

A CHANGE IN CULTURE: PEOPLE COLLABORATING…



…FOLLOWING ESTABLISHED PROCESSES…

42crunch

…AND USING THE RIGHT TOOLS.

Design

Developer initiates security work at design time.

Best practices and recommendations are documented.

APIsecurity.io

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

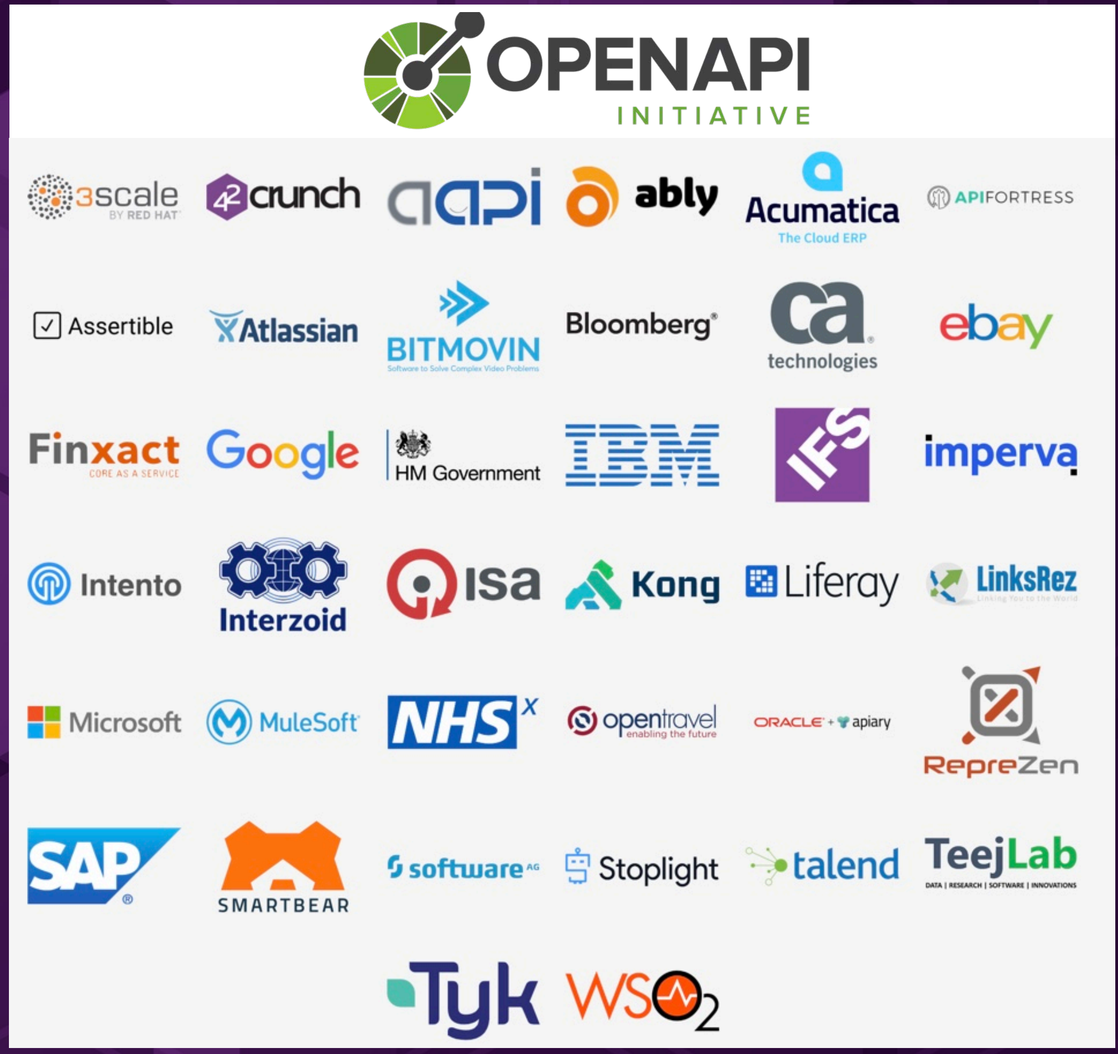
Deploy & Protect

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

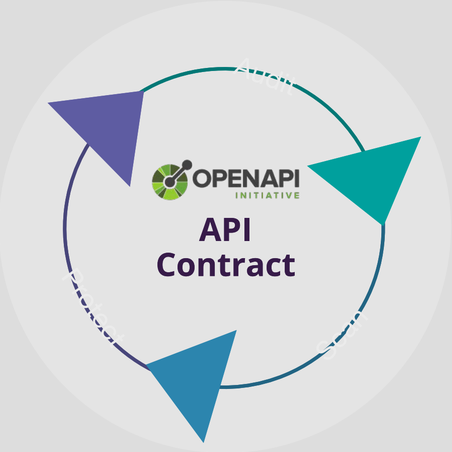
- Developers know how the application was built!
- OpenAPI specification is leveraged to describe the API contract.
- Once the API contract is defined by the developer, the security process becomes clear and straight forward !



"If you describe your API, we will secure it"

EMPOWER DEVELOPERS TO BUILD THE ULTIMATE WHITELIST

VALIDATE OPENAPI CONTRACT CONTENTS



AUDIT

- 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

Platform Architecture

42crunch API Security Platform

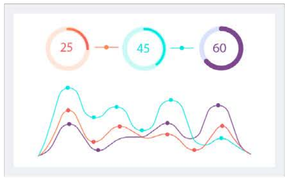
42Crunch Platform Services

Audit

Scan

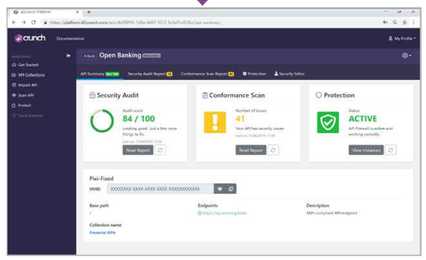
Protect

Platform Reports / Dashboards



42Crunch API Firewalls

REST API



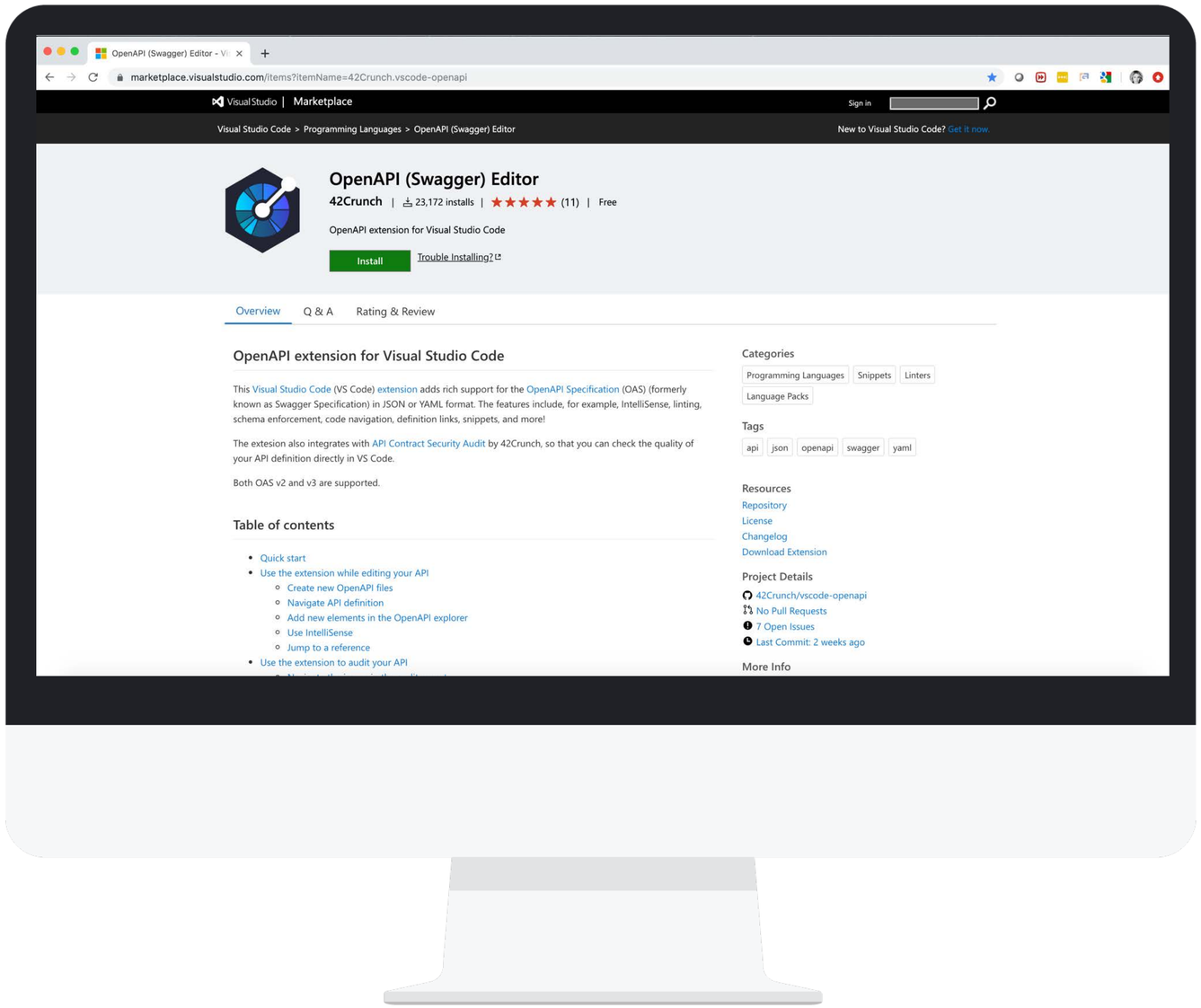
IDEs

CI /CD Plugins

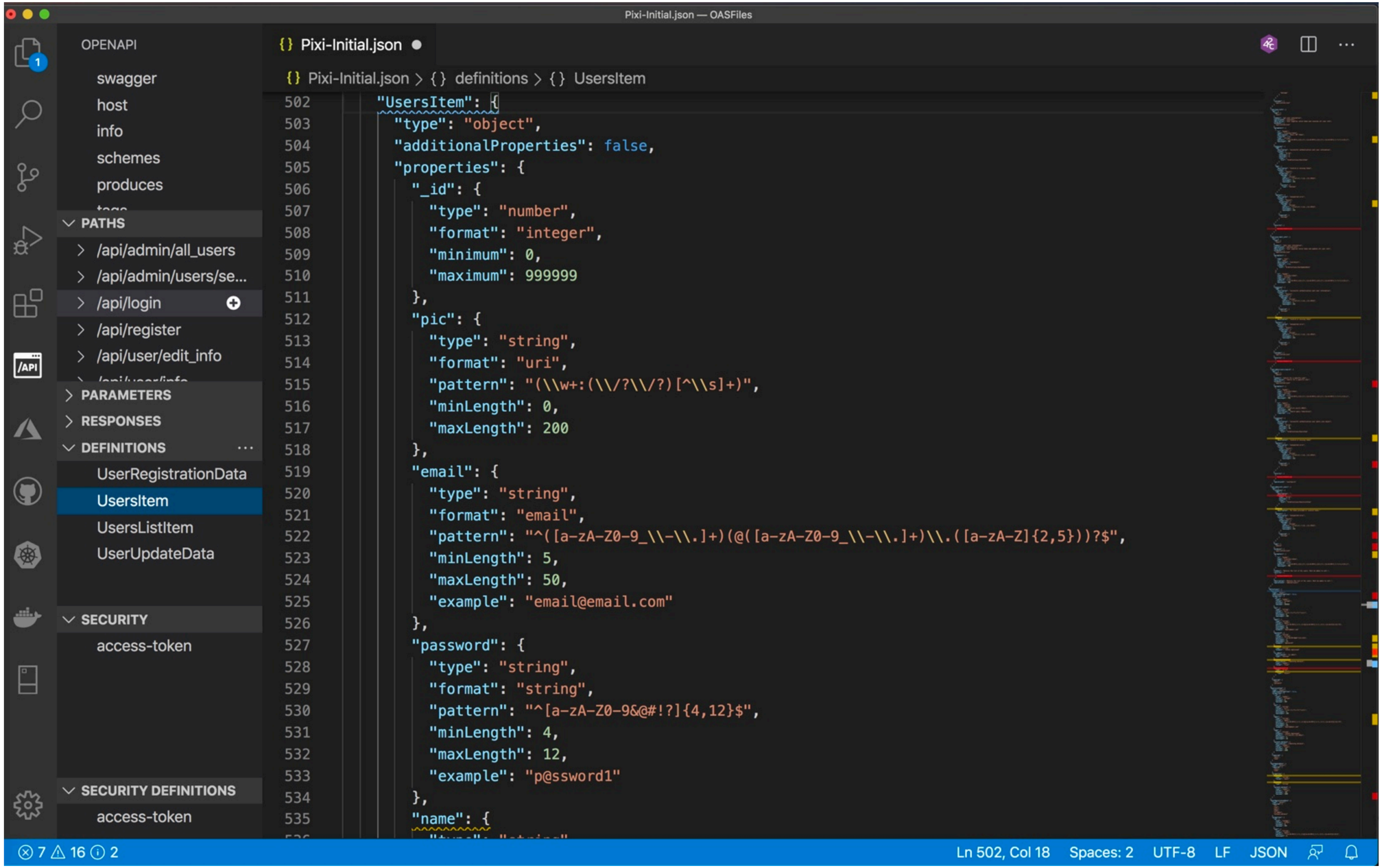
Scripting

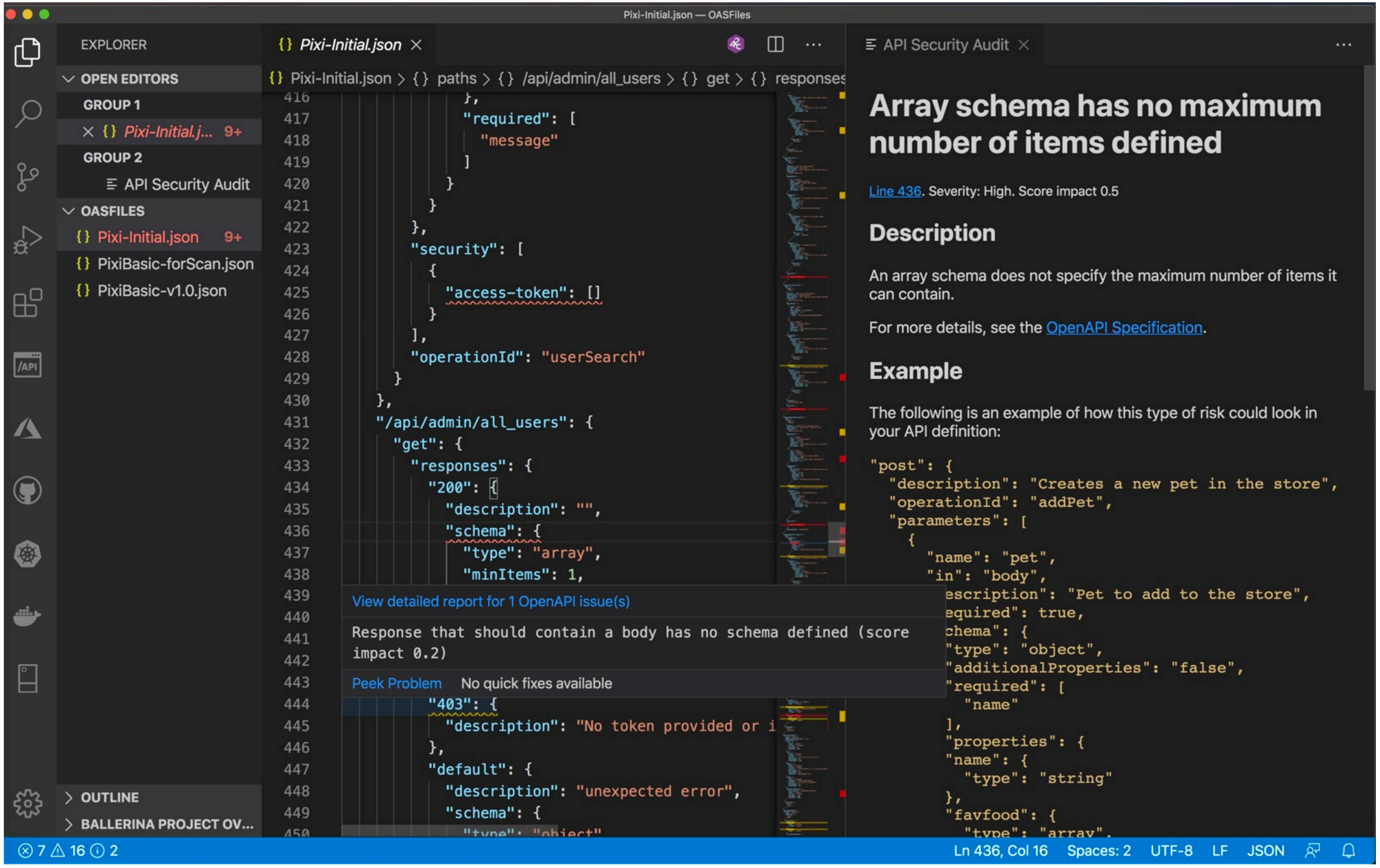
DEMO PART 1:

OPENAPI EDITOR/AUDIT FOR VSCODE



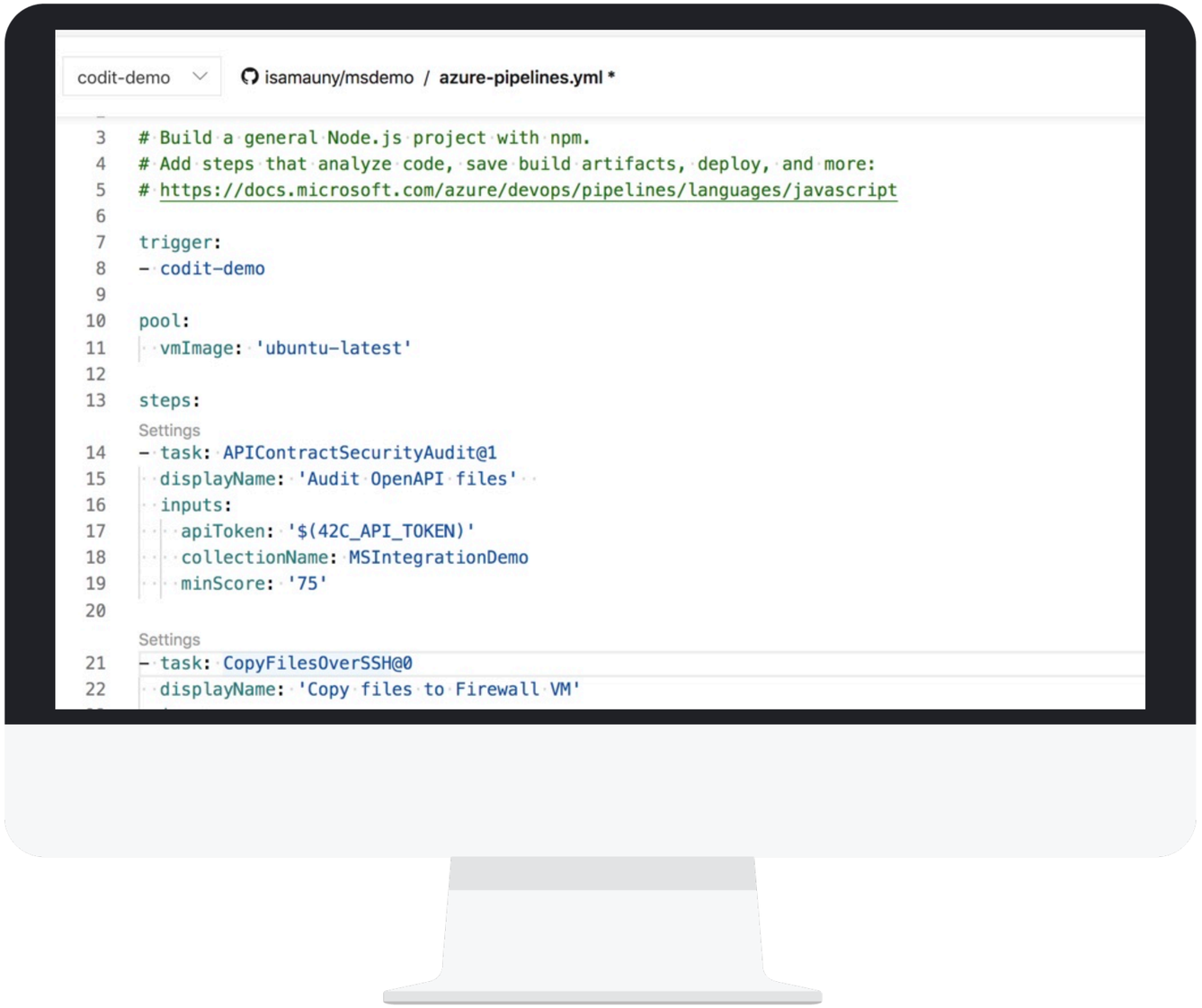
https://marketplace.visualstudio.com/items?itemName=42Crunch.vscode-openapi

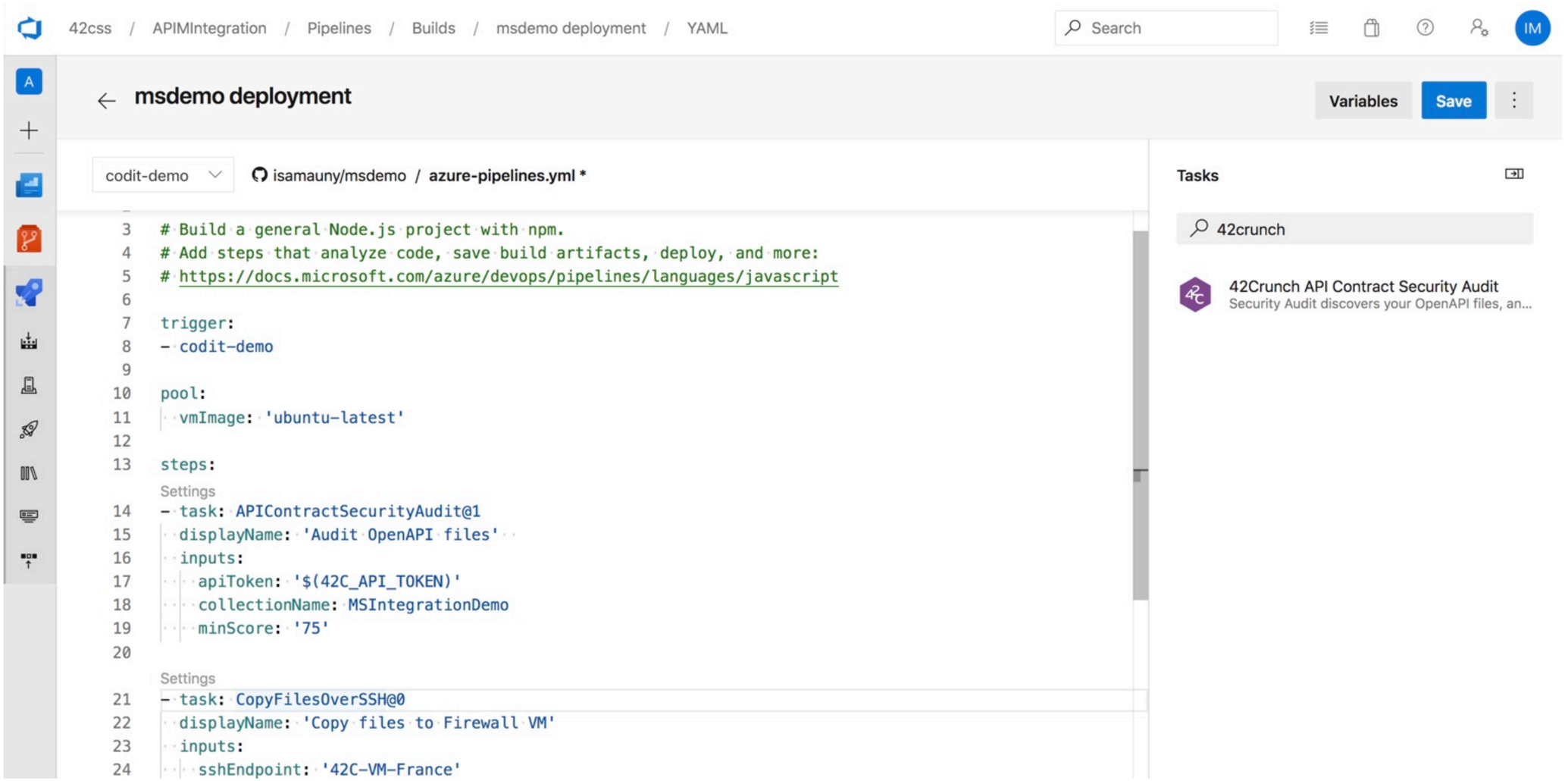




DEMO STEP 2:

AZURE DEVOPS INTEGRATION





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