
OWASP 2023
GLOBAL
AppSec
SINGAPORE
VIRTUAL CONFERENCE
OCTOBER 3-5

OWASP 2023 GLOBAL AppSec

SINGAPORE
VIRTUAL CONFERENCE
OCTOBER 3-5

Using LLMs and Generative AI to Fix Software Vulnerabilities

Asankhaya Sharma

Asankhaya Sharma, Co-Founder & CTO, https://patched.codes

2003 — 2007 — 2010 — 2014 — 2018 — 2019 — 2022 — 2023

Microsoft

Code Analysis Tool for .NET v2.0

DIDAR – Database Intrusion Detection with Automated Recovery

NUS National University of Singapore

HIP/SLEEK :Automatic Verification and Specification Inference System

[:] SourceClear

Botwall4J

GramTest

ca technologies

VERACODE

SRC CLR SCA Agent

AutoFix

Static Analysis + LLM = AutoFix

Securade.ai

patched

Agenda

Evolution of Application Security
(The Pledge)

- Persistence of software vulnerabilities
- Changing software development practices

Rise of Generative AI
(The Turn)

- Code generation, bug fixing and vulnerability remediation
- RAG, SAG and SAGA

Developer Less Security
(The Prestige)

- Patched Coder
- Static Analysis Eval

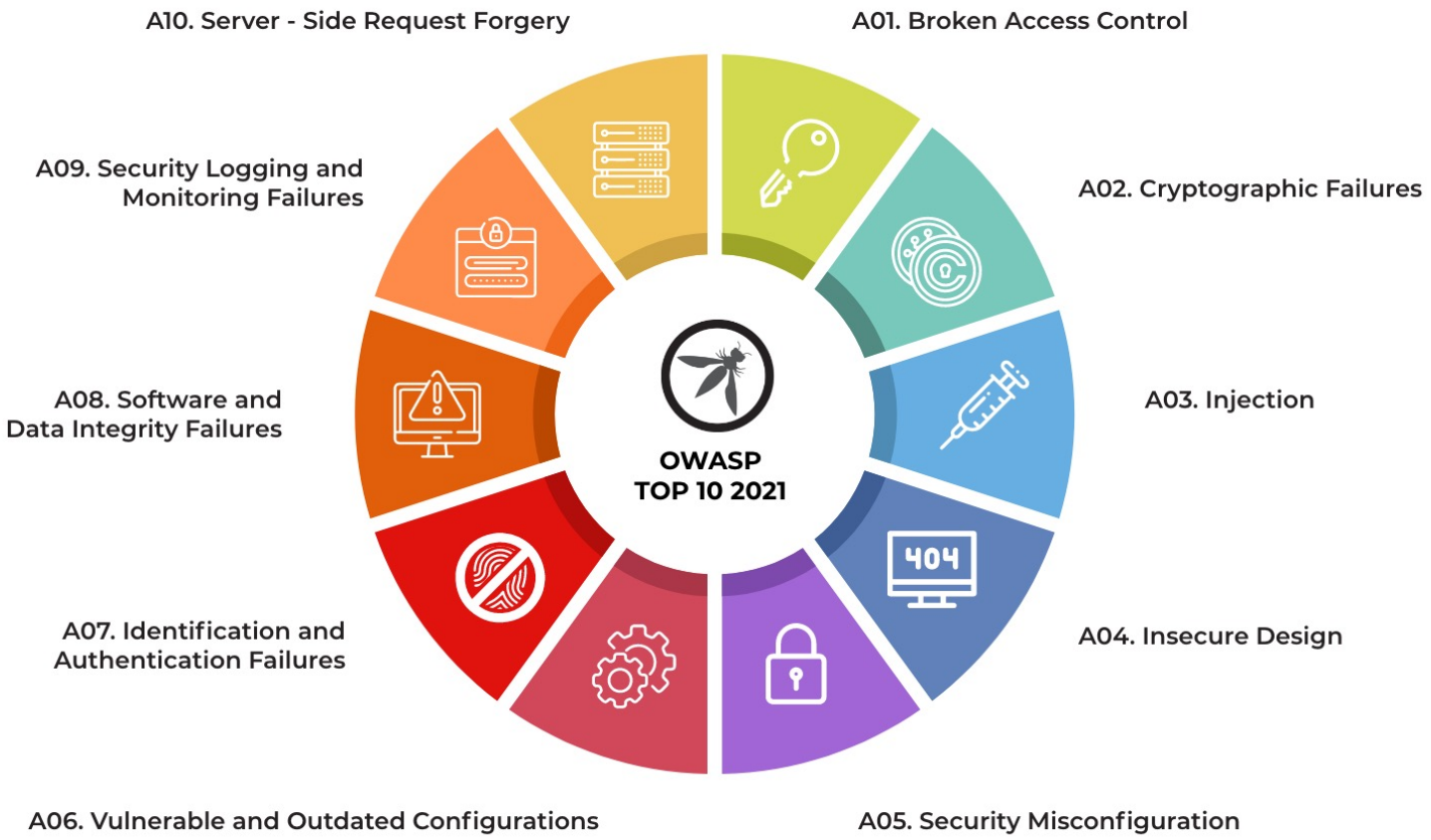
"Every magic trick consists of three parts, or acts." ~ John Cutter, The Prestige

Evolution of Application Security

(The Pledge)

- Persistence of software vulnerabilities
- Changing software development practices

OWASP Top Ten



https://www.horangi.com/blog/real-life-examples-of-web-vulnerabilities

OWASP Top Ten

	2003/2004	2007	2010	2013	2017	2021
A1	Unvalidated Input	Cross-Site Scripting (XSS)	Injection	Injection	Injection	Broken Access Control
A2	Broken Access Control	Injection Flaws	Cross-Site Scripting (XSS)	Broken Authentication and Session Management	Broken Authentication	Cryptographic Failures
A3	Broken Authentication and Session Management	Malicious File Execution	Broken Authentication and Session Management	Cross-Site Scripting (XSS)	Sensitive Data Exposure	Injection
A4	Cross-Site Scripting (XSS)	Insecure Direct Object References	Insecure Direct Object References	Insecure Direct Object References	XML External Entities (XXE)	Insecure Design
A5	Buffer Overflow	Cross-Site Request Forgery (CSRF)	Cross-Site Request Forgery (CSRF)	Security Misconfiguration	Broken Access Control	Security Misconfiguration
A6	Injection Flaws	Information Leakage and Improper Error Handling	Security Misconfiguration	Sensitive Data Exposure	Security Misconfiguration	Vulnerable and Outdated Components
A7	Improper Error Handling	Broken Authentication and Session Management	Insecure Cryptographic Storage	Missing Function Level Access Control	Cross-Site Scripting	Identification and Authentication Failures
A8	Insecure Storage	Insecure Cryptographic Storage	Failure to Restrict URL Access	Cross-Site Request Forgery (CSRF)	Insecure Deserialization	Software and Data Integrity Failures
A9	Application Denial of Service	Insecure Communications	Insufficient Transport Layer Protection	Using Components with Known Vulnerabilities	Using Components with Known Vulnerabilities	Security Logging and Monitoring Failures
A10	Insecure Configuration Management	Failure to Restrict URL Access	Unvalidated Redirects and Forwards	Unvalidated Redirects and Forwards	Insufficient Logging & Monitoring	Server-Side Request Forgery (SSRF)

6 Major Changes Witnessed by Software Development

Proprietary to
Open Source Software

Waterfall to
Agile Methodology

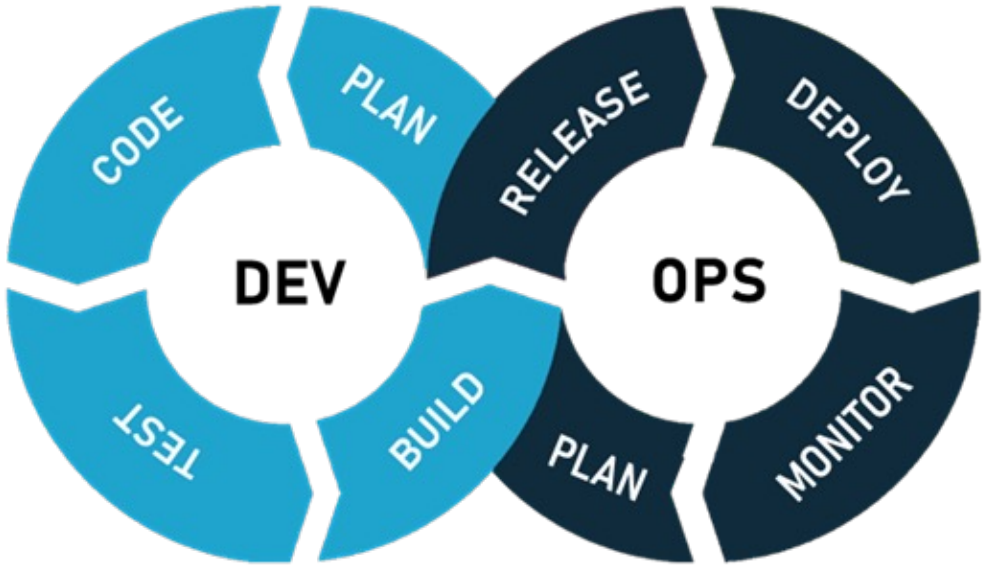
Silos to DevOps
Philosophy

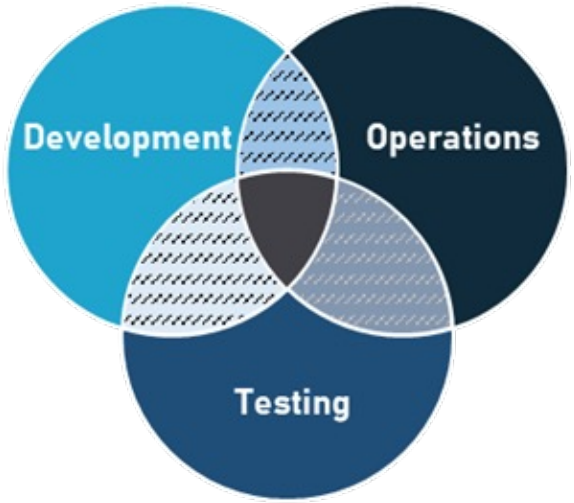
On-Premise to
Cloud Computing

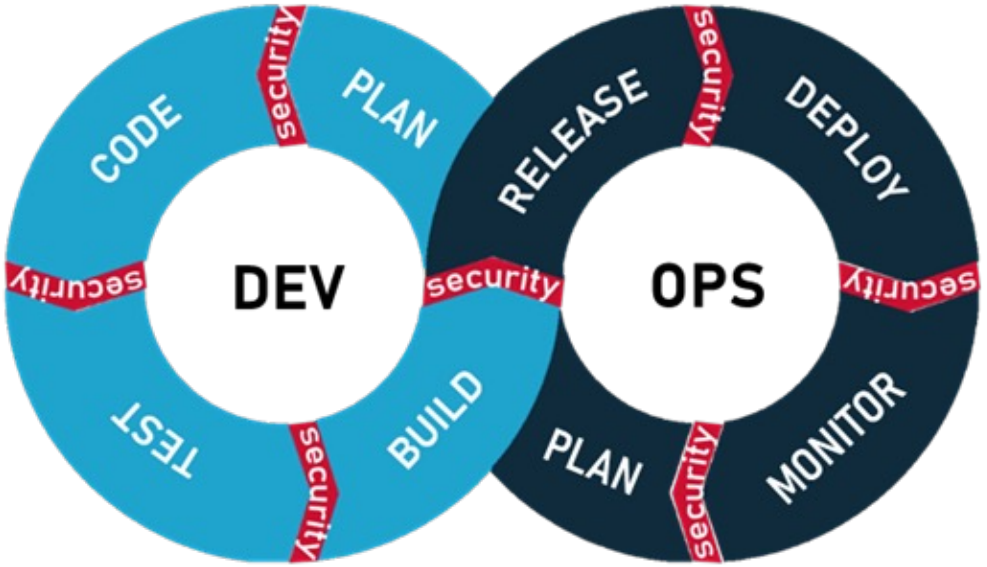
Isolated Models to
Connected APIs

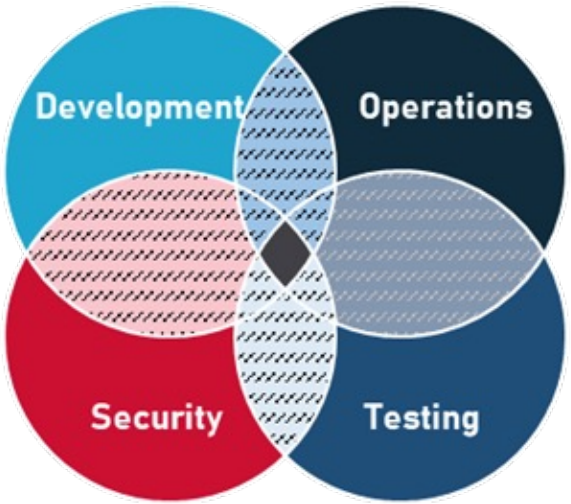
In-house to
Outsourcing

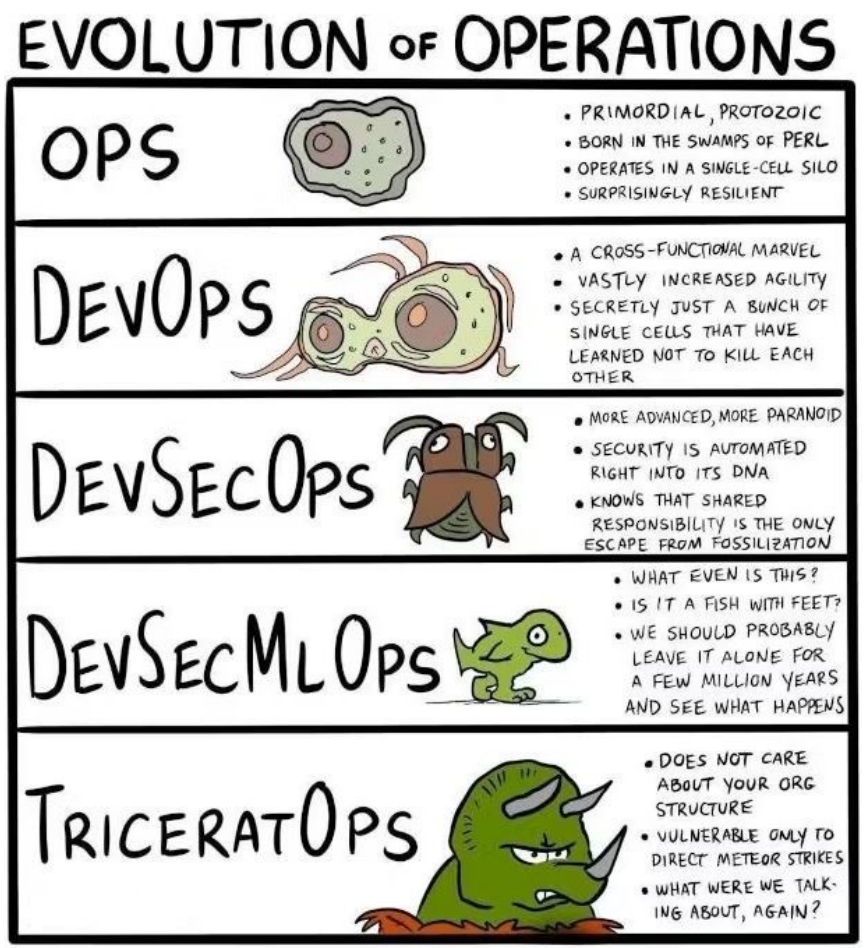
www.techtic.com

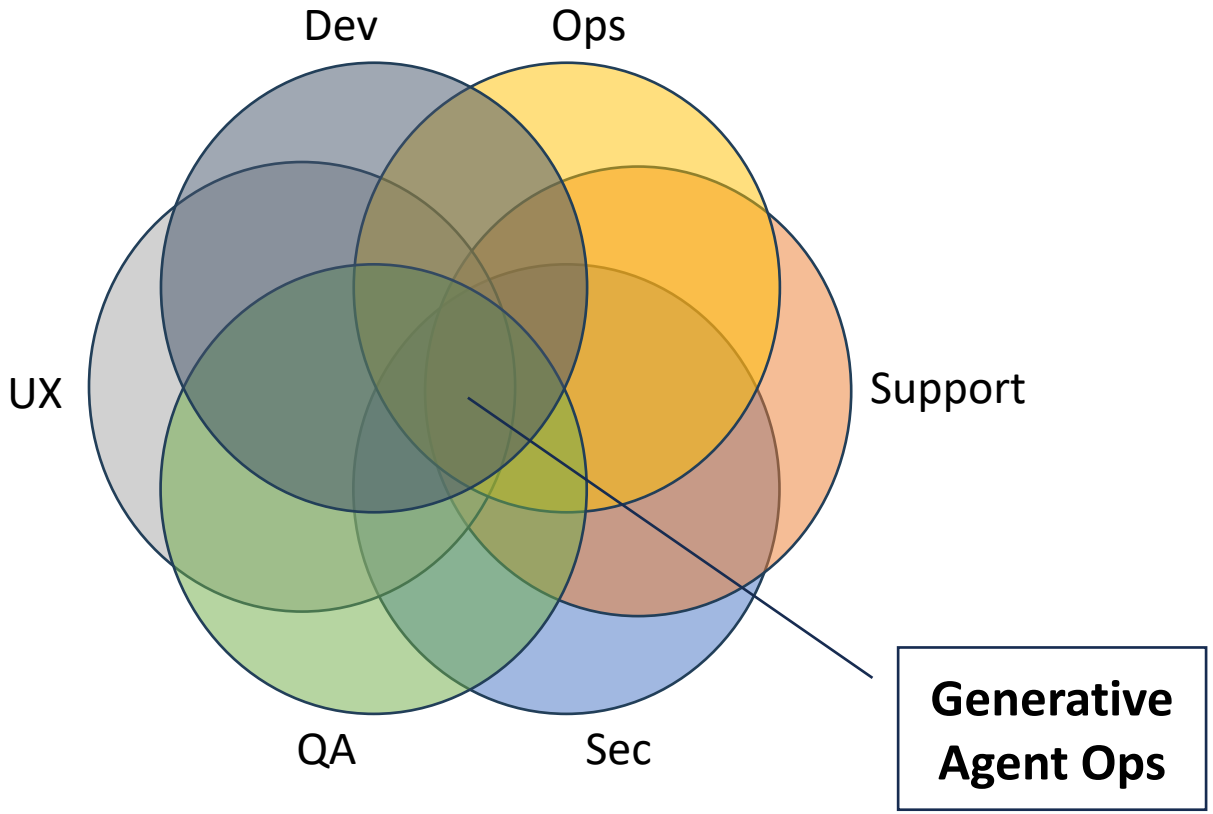

CODE
PLAN
RELEASE
DEPLOY
DEV
OPS
TEST
BUILD
PLAN
MONITOR


Development
Operations
Testing

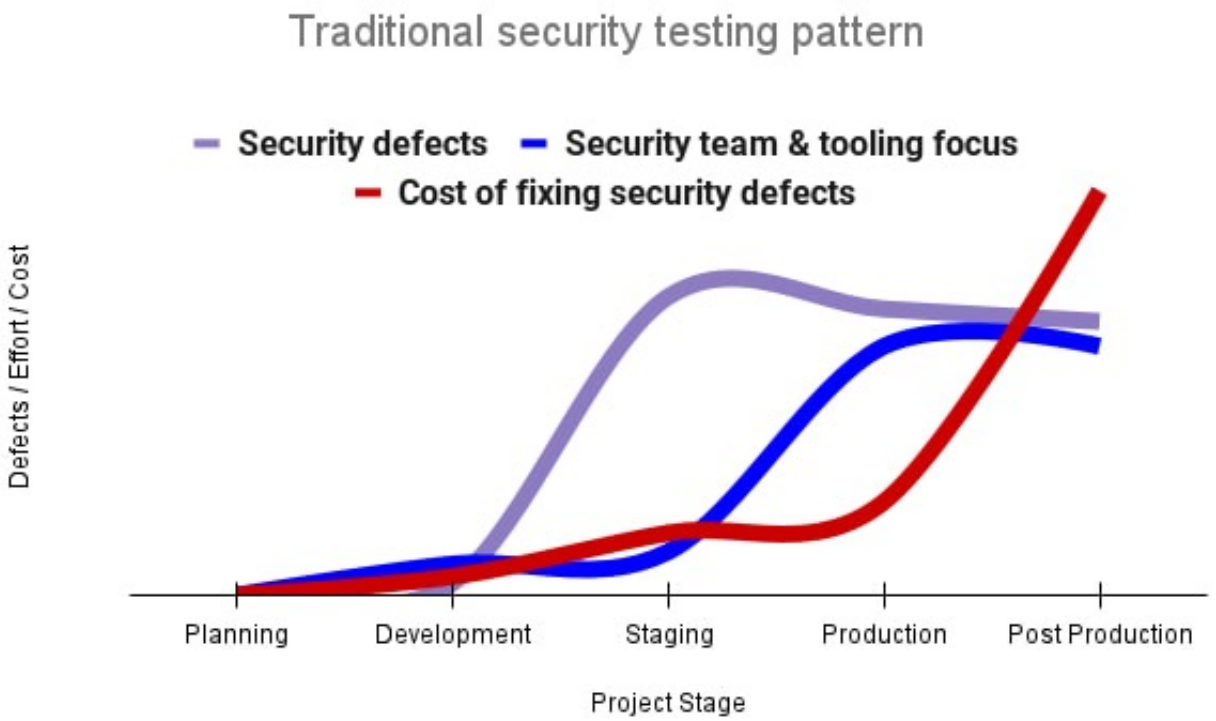

CODE
security
PLAN
RELEASE
security
DEPLOY
security
DEV
security
OPS
security
TEST
security
BUILD
PLAN
security
MONITOR

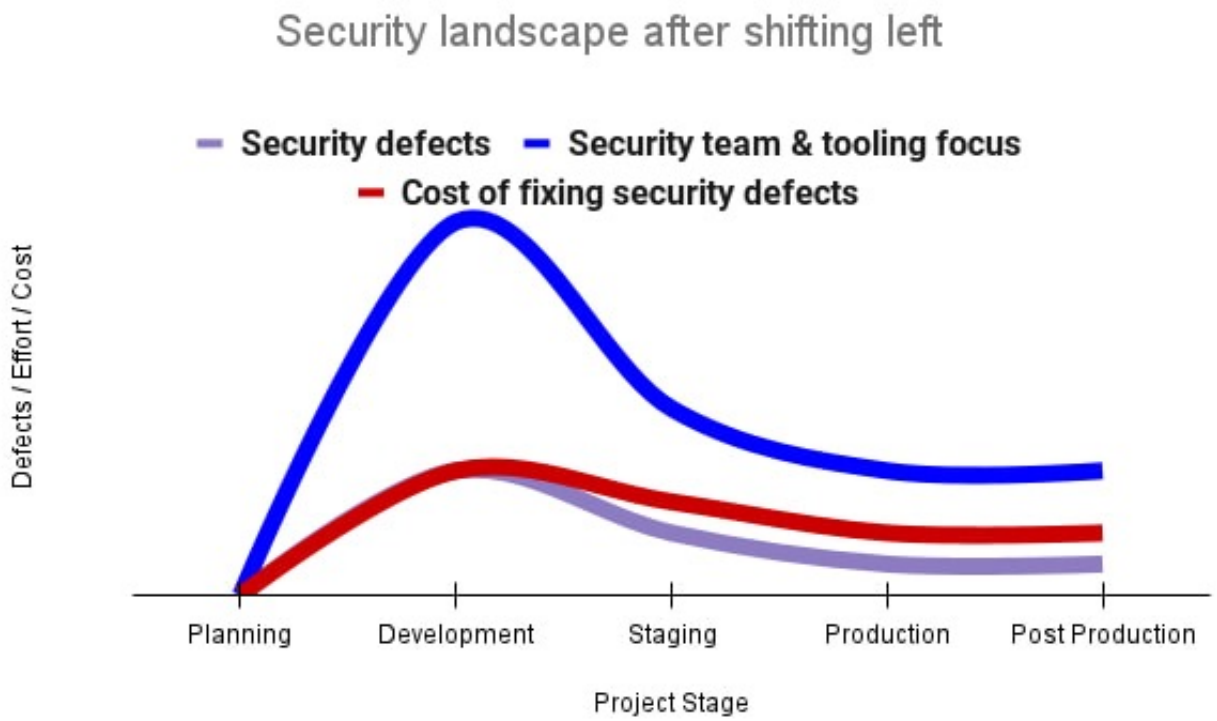

Development
Operations
Security
Testing


EVOLUTION OF OPERATIONS
OPS
• PRIMORDIAL, PROTOZOIC
• BORN IN THE SWAMPS OF PERL
• OPERATES IN A SINGLE-CELL SILO
• SURPRISINGLY RESILIENT
DEVOPS
• A CROSS-FUNCTIONAL MARVEL
• VASTLY INCREASED AGILITY
• SECRETLY JUST A BUNCH OF SINGLE CELLS THAT HAVE LEARNED NOT TO KILL EACH OTHER
DEVSECOPS
• MORE ADVANCED, MORE PARANOID
• SECURITY IS AUTOMATED RIGHT INTO ITS DNA
• KNOWS THAT SHARED RESPONSIBILITY IS THE ONLY ESCAPE FROM FOSSILIZATION
DEVSECMLOPS
• WHAT EVEN IS THIS?
• IS IT A FISH WITH FEET?
• WE SHOULD PROBABLY LEAVE IT ALONE FOR A FEW MILLION YEARS AND SEE WHAT HAPPENS
TRICERATOPS
• DOES NOT CARE ABOUT YOUR ORG STRUCTURE
• VULNERABLE ONLY TO DIRECT METEOR STRIKES
• WHAT WERE WE TALKING ABOUT, AGAIN?


Dev
Ops
UX
Support
QA
Sec
Generative Agent Ops

Shift Left





https://cloud.google.com/blog/products/identity-security/shift-left-on-google-cloud-security-invest-now-save-later

building security tools for developers

v/s

developer tools for security

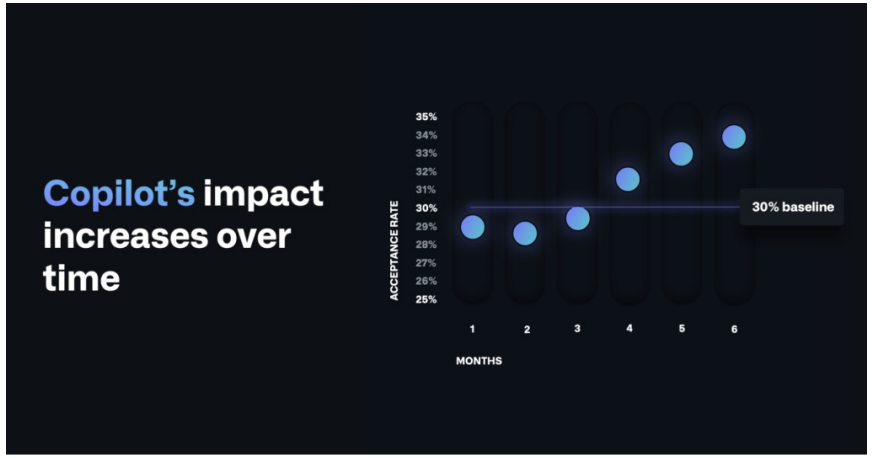
Rise of Generative AI

(The Turn)

- Code generation, bug fixing and vulnerability remediation
- RAG, SAG and SAGA

Code LLMs

OpenAI

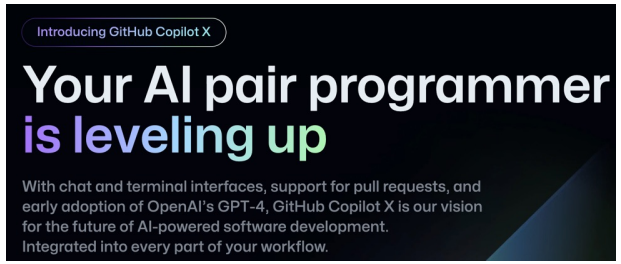


https://eventyay.com/e/7cfe0771/session/8146

`codex` `code-davinci-002` `GPT-3.5-turbo` `GPT-4`

GitHub Copilot

Copilot for Business new



foss asia

Open-access Code LLMs

StarCoderBase is a 15B parameter decoder trained on 1T tokens of code in 80+ programming languages

Trained on additional 30B tokens of Python

StarCoder

StarCoder: MAY THE SOURCE BE WITH YOU!

https://arxiv.org/abs/2305.06161

StarCoderBase

Different sizes

starcoderbase-1b
starcoderbase-3b
starcoderbase-7b

StarCoderPlus

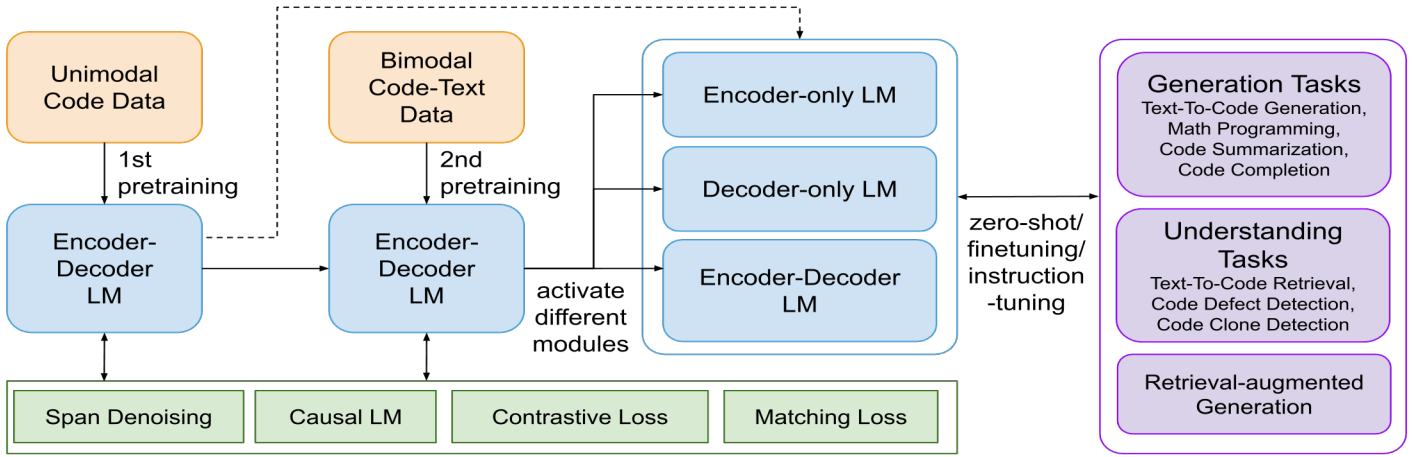
Trained on additional 600B tokens of natural text from RefinedWeb and Wikipedia

StarChat-Beta

fine-tuned StarCoderPlus with an "uncensored" variant of the openassistant-guanaco dataset

The Stack - a 6.4TB of source code in 358 programming languages from permissive licenses.

Open-access Dataset



CodeT5+: Open Code Large Language Models for Code Understanding and Generation

Yue Wang*, Hung Le*, Akhilesh Deepak Gotmare, Nghi D.Q. Bui, Junnan Li, Steven C.H. Hoi
Salesforce AI Research
https://github.com/salesforce/CodeT5/tree/main/CodeT5+

https://arxiv.org/abs/2305.07922

CodeT5+

Different sizes
220M, 770M

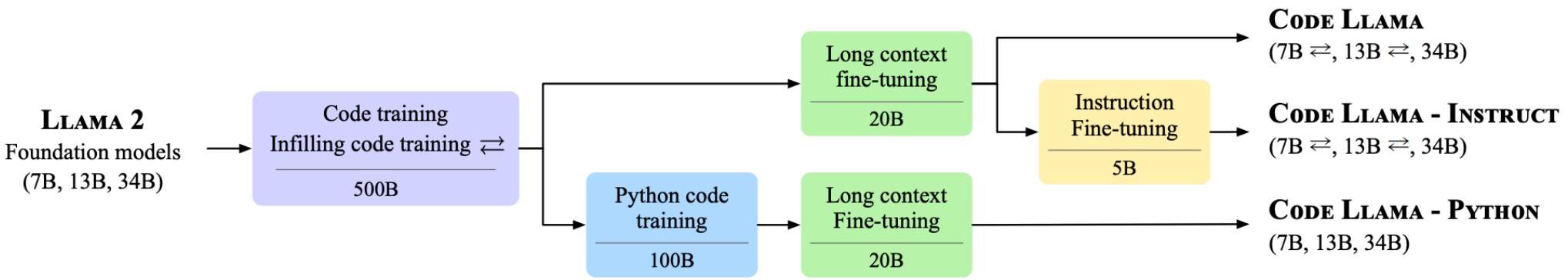
CodeT5+

Different sizes
2B, 6B, 16B
initialized from
CodeGen model

InstructCodeT5+

Fine-tuned with
data generated by
using OpenAI's API

Code Llama



Code Llama: Open Foundation Models for Code

Baptiste Rozière†, Jonas Gehring†, Fabian Gloeckle†,*, Sten Sootla†, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi◇, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, Gabriel Synnaeve†

Meta AI

https://arxiv.org/abs/2308.12950

How do we evaluate Code LLMs?

HumanEval

A dataset of 164 python programs with unit tests to measure functional correctness for synthesizing programs from docstrings

Evaluating Large Language Models Trained on Code

https://arxiv.org/abs/2107.03374

```python
def incr_list(l: list):
    """Return list with elements incremented by 1.
    >>> incr_list([1, 2, 3])
    [2, 3, 4]
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])
    [6, 4, 6, 3, 4, 4, 10, 1, 124]
    """
    return [i + 1 for i in l]
```

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Code Generation Closed v/s Open Models

HumanEval	Zero-shot pass@1 (%)
GPT-4	86.6
CodeLlama-34b-Python	53.29
InstructCodeT5+	37
StarCoder	33.6

⭐ Big Code Models Leaderboard

T	Models	humaneval-python
🔶	Phind-CodeLlama-34B-v2	71.95
🔶	WizardCoder-Python-34B-V1.0	70.73
🔶	Phind-CodeLlama-34B-Python-v1	70.22
🔶	Phind-CodeLlama-34B-v1	65.85
🔶	WizardCoder-Python-13B-V1.0	62.19
🔶	WizardCoder-15B-V1.0	58.12
🟢	CodeLlama-34b-Python	53.29

https://huggingface.co/spaces/bigcode/bigcode-models-leaderboard

Code Generation (HumanEval)

HumanEval	Zero-shot pass@1 (%)
GPT-4	86.6
Phind-CodeLlama-34B-v2	71.95
WizardCoder-Python-34B-v1.0	70.73
CodeLlama-34b-Python	53.29

Infilling with Code Generation

```
// some code
<FILL-HERE>
// some more code
```

```
<prefix>
// some code
<suffix>
// some more code
<middle>
```

```
<prefix>
// some code
<suffix>
// some more code
<middle>
// generated code
```

```
// some code
// generated code
// some more code
```

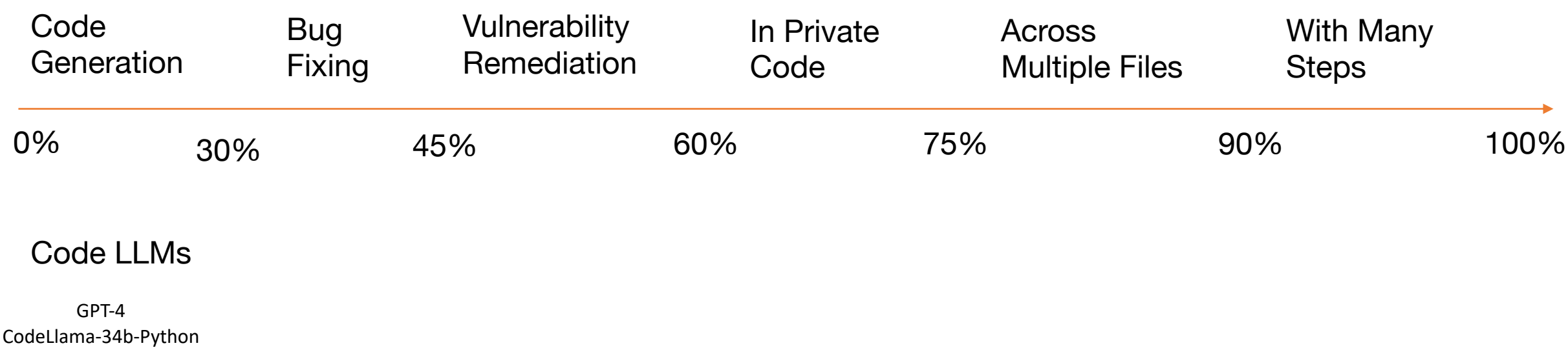
Infilling to Fix Vulnerabilities

```
String output = Launcher.RESOURCES.getString("WinstoneResponse.ErrorPage",
// BUG: CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
// new String[] { sc + "", (msg == null ? "" : msg), sw.toString(),
// FIXED:
new String[] { sc + "", URIUtil.htmlEscape(msg == null ? "" : msg),

URIUtil.htmlEscape(sw.toString()),Launcher.RESOURCES.getString("ServerVersion"),"" + new Date() });

response.setContentLength(output.getBytes(response.getCharacterEncoding()).length);
Writer out = response.getWriter();
```

Examining Zero-Shot Vulnerability Repair with Large Language Models

https://arxiv.org/abs/2112.02125

Fixing Software Vulnerabilities



Bug Fixing is Harder

Fix bug in fibonacci

```
def fibonacci(n):
        if n == 0:
                return 0
        elif n == 1 or n == 2:
                return 1
        else:
                return fibonacci(n-1) - fibonacci(n-2)
```

Requires an LLM that can follow instructions (or is chatty)

```
def fibonacci(n):
        if n == 0:
                return 0
        elif n == 1 or n == 2:
                return 1
        else:
                return fibonacci(n-1) + fibonacci(n-2)
```

HumanEvalFix

A dataset created by adding a bug to each of the 164 HumanEval solutions. Bugs are written such that the code still runs but produces an incorrect result leading to at least one unit test failing.

https://arxiv.org/abs/2308.07124

HumanEvalFix	Zero-shot pass@1 (%)
GPT-4	47
Phind-CodeLlama-34B-v2	39.57
WizardCoder-Python-34B-v1.0	38.66
CodeLlama-34b-Instruct	33.14

```
from typing import List


def has_close_elements(numbers: List[float], threshold: float) ->
bool:
    for idx, elem in enumerate(numbers):
        for idx2, elem2 in enumerate(numbers):
            if idx != idx2:
                distance = elem - elem2
                if distance < threshold:
                    return True

    return False

def check(has_close_elements):
    assert has_close_elements([1.0, 2.0, 3.9, 4.0, 5.0, 2.2], 0.3) ==
True
    assert has_close_elements([1.0, 2.0, 3.9, 4.0, 5.0, 2.2], 0.05) ==
False
    assert has_close_elements([1.0, 2.0, 5.9, 4.0, 5.0], 0.95) == True
    assert has_close_elements([1.0, 2.0, 5.9, 4.0, 5.0], 0.8) == False
    assert has_close_elements([1.0, 2.0, 3.0, 4.0, 5.0, 2.0], 0.1) ==
True
    assert has_close_elements([1.1, 2.2, 3.1, 4.1, 5.1], 1.0) == True
    assert has_close_elements([1.1, 2.2, 3.1, 4.1, 5.1], 0.5) == False

check(has_close_elements)

Fix bugs in has_close_elements.
```

```
from typing import List


def has_close_elements(numbers: List[float], threshold: float) ->
bool:
    for idx, elem in enumerate(numbers):
        for idx2, elem2 in enumerate(numbers):
            if idx != idx2:
                distance = abs(elem - elem2)
                if distance < threshold:
                    return True

    return False
```

Are commits a good data source for instruction tuning code LLMs?

A Machine Learning Approach for Vulnerability Curation

Yang Chen
Veracode
ychen@veracode.com

Andrew E. Santosa
Veracode
asantosa@veracode.com

Ang Ming Yi
Veracode
mang@veracode.com

Abhishek Sharma
Veracode
absharma@veracode.com

Asankhaya Sharma
Veracode
asharma@veracode.com

David Lo
Singapore Management University
davidlo@smu.edu.sg

https://dl.acm.org/doi/10.1145/3379597.3387461

Code Before

```
import numpy as np
import matplotlib.pyplot as plt

# generate sample data
x_data = np.linspace(-5, 5, 20)
y_data = np.random.normal(0.0, 1.0, x_data.size)

plt.plot(x_data, y_data, 'o')
plt.show()
```

Commit Message

Change to sin() function with noise

Code After

```
import math
import numpy as np
import matplotlib.pyplot as plt

# generate sample data
x_data = np.linspace(-math.pi, math.pi, 30)
y_data = np.sin(x_data) + np.random.normal(0.0, 0.1, x_data.size)

plt.plot(x_data, y_data, 'o')
plt.show()
```

Patched Coder

CodeLlama-34b-Python → patched-coder-34b

CommitPackFT is a 2GB filtered version of CommitPack to contain only high-quality commit messages that resemble natural language instructions.

https://hf.co/datasets/bigcode/commitpackft

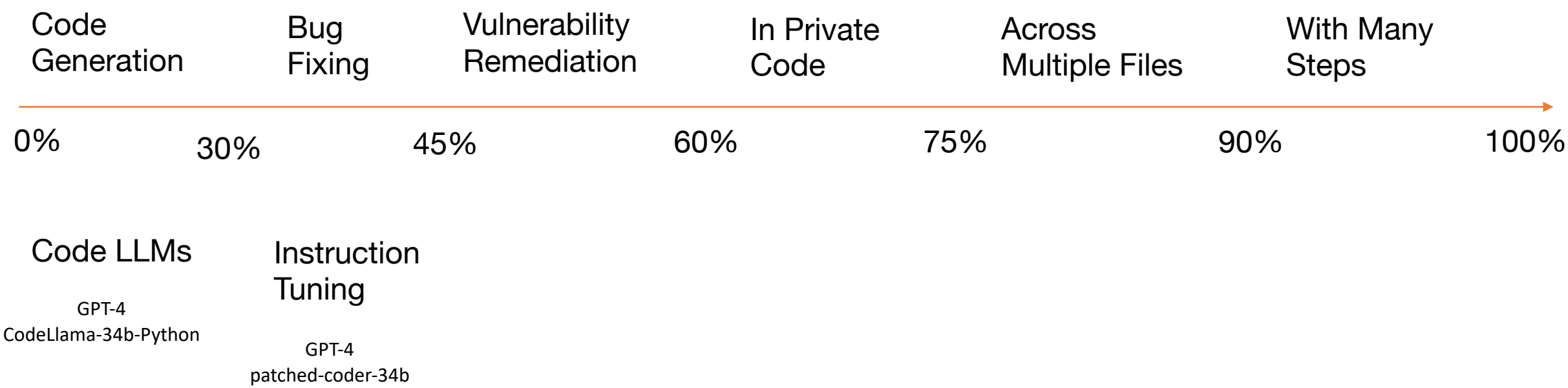
```
### Instruction:
commit_msg

### Input:
code_before

### Response:
code_after
```

https://hf.co/patched-codes/patched-coder-34b

Patched Coder is the SOTA Open Code LLM

Code LLM	HumanEval	HumanEvalFix
GPT-4	86.6	47
Phind-CodeLlama-34B-v2	71.95	39.57
WizardCoder-Python-34B-v1.0	70.73	38.66
patched-coder-34b	53.57	41.34
CodeLlama-34b	53.29	33.14

Fixing Software Vulnerabilities



Static Analysis Eval

A dataset of 76 Python programs taken from real Python open source projects (top 1000 on GitHub), where each program is a file that has exactly 1 vulnerability as detected by a particular static analyzer (Semgrep).

https://hf.co/datasets/patched-codes/static-analysis-eval

```python
import os
import requests


def download_file(url, path):
    """
    def download_model(model_url)
    download pretrained h5 __model file
    Args:
        url (str): __model download url
        path (str): download path
    Returns:
        True if download succeed
        False otherwise
    """
    try:
        request = requests.get(url, allow_redirects=True)
        path_parent = os.path.abspath(os.path.join(path, os.pardir))
        os.makedirs(path_parent, exist_ok=True)
        open(path, 'wb').write(request.content)


        return True
    except:
        return False


def update_model(model_path):
    pass
```

Static Analysis Eval

1. Scan with static analyzer (Semgrep)
2. Extract <CWE>, <vulnerable line(s)> and <error message> from the output of the analyzer
3. Prompt the code LLM to generate fix for the vulnerability
4. Scan again with the static analyzer to check if the error message goes away

Static Analysis Eval	Zero-shot pass@1 (%)
GPT-4	55.26
patched-coder-34b	51.32

```
### Instruction:
<error message>

Fix vulnerability <CWE> in

<vulnerable line(s)>

### Input:
vulnerable_code


### Response:
fixed_code
```

Fixing Software Vulnerabilities

Code Generation
Bug Fixing
Vulnerability Remediation
In Private Code
Across Multiple Files
With Many Steps

0% 30% 45% 60% 75% 90% 100%

Code LLMs

GPT-4
CodeLlama-34b-Python

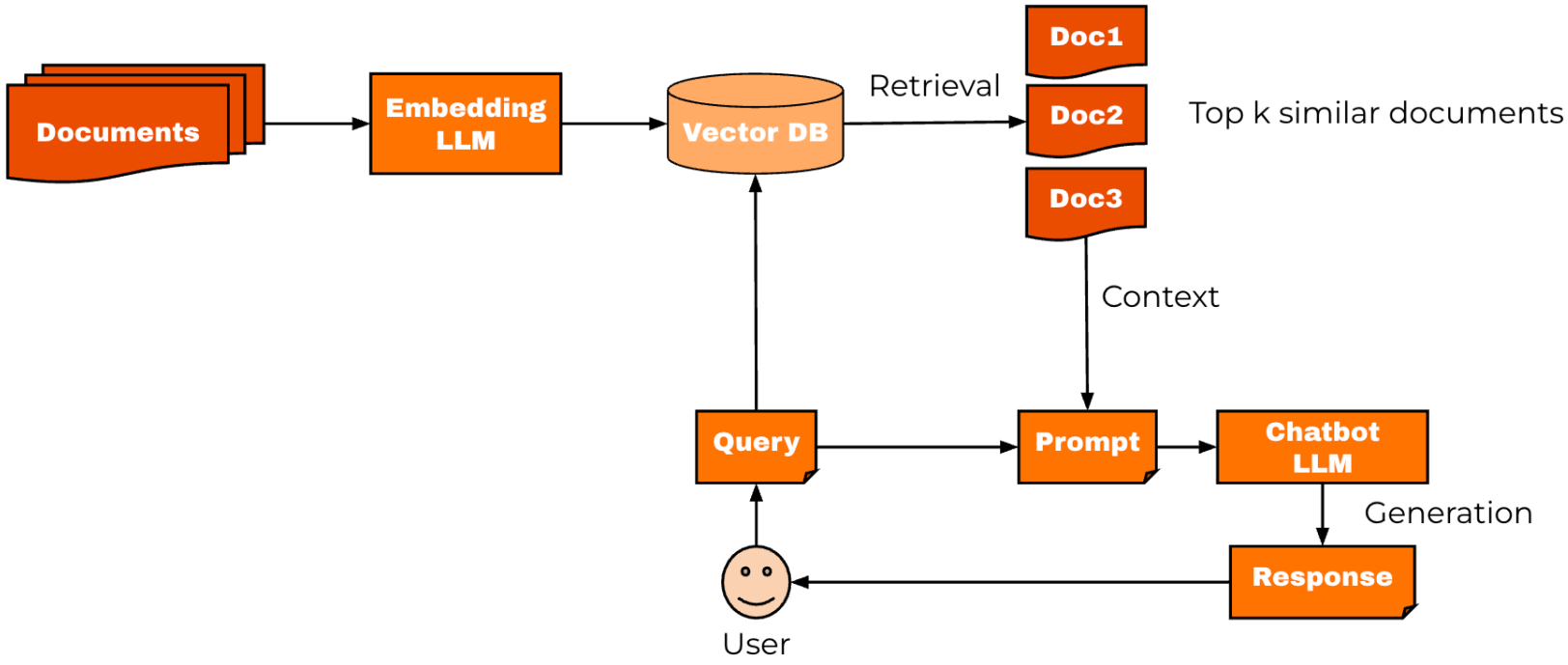
Instruction Tuning

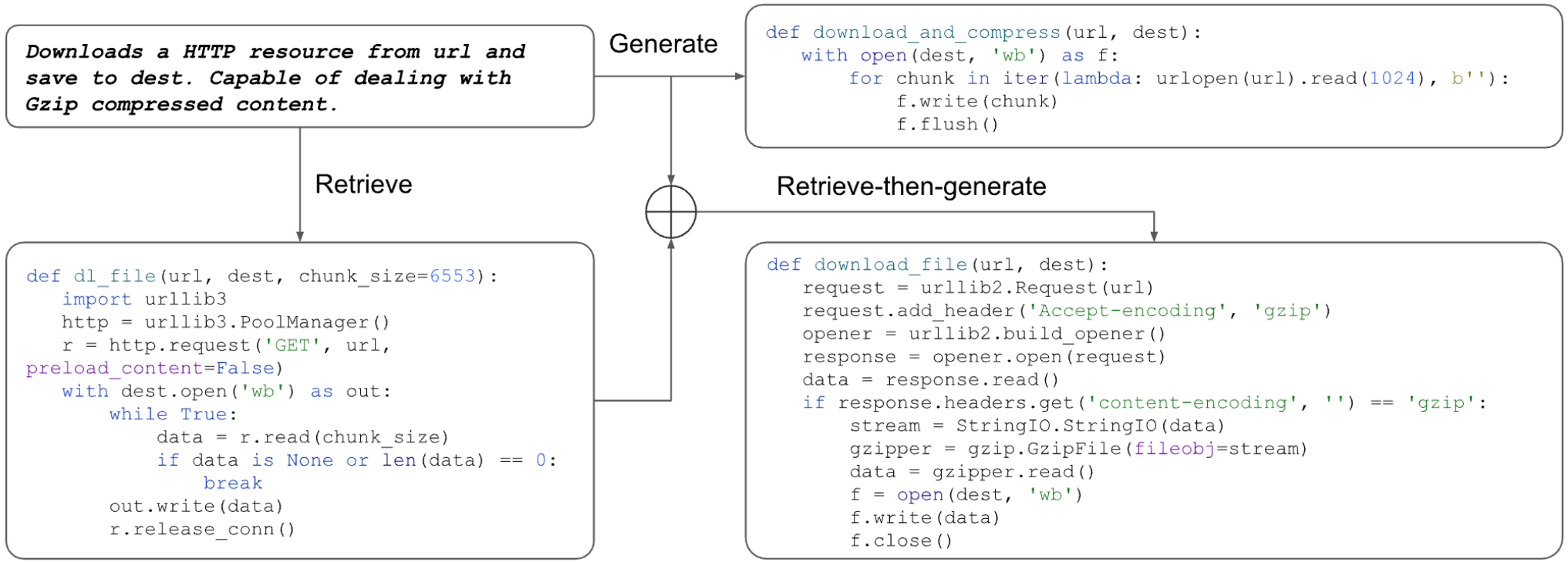
GPT-4
patched-coder-34b

Prompting with Security Context

GPT-4
patched-coder-34b

Retrieval Augmented Generation (RAG)





CodeT5+’s encoder-decoder architecture enables end-to-end retrieval-augmented code generation

Retrieval Augmented Generation (RAG)

1. Unimodal (text or code)
2. Bimodal (code and description pairs)
3. Bimodal with context (instruction with before_code and after_code pairs)

Build a few-shot prompt

```
// Buggy code [snippet 1] // Fixed code [completion 1]
// Buggy code [snippet 2] // Fixed code [completion 2]
// Buggy code [snippet X]
```

```
### Instruction:
<retrieved_similar_commit_message>

### Input:
<retrieved_similar_vulnerable_code>

### Response:
<retrieved_fix>

### Instruction:
<error message>

Fix vulnerability <CWE> in <vulnerable line(s)>

### Input:
vulnerable_code

### Response:
fixed_code
```

Use obfuscation to preserve structural fix

```
private Designer getDesigner(Object adaptable) {
    ResourceResolver resolver = getResourceResolver(adaptable);
    if (resolver != null) {
        return resolver.adaptTo(Designer.class);
    }
    return null;
}
```

```
private CLASS_1 METHOD_1(CLASS_2 VAR_1) {
    CLASS_3 VAR_2 = METHOD_2(VAR_1);
    if (VAR_2 != null) {
        return VAR_2.METHOD_3(CLASS_1.METHOD_4);
    }
    return null;
}
```

Fixing vulnerabilities with RAG

Approach	NPD		RL		TSV	
	Java	*C#*	*Java*	*C#*	*Java*	*C#*
Demonstration (Codex)	20.3	30.1	25.3	29.1	19.0	16.7
Completion (Codex)	6.7	6.1	7.8	5.7	3.9	0.0
Instruction (Davinci)	40.5	22.2	53.8	19.7	41.3	33.3
Finetuning (Codex)	49.7	58.1	60.0	51.9	64.4	70.0
InferFix	**59.5**	**66.7**	**71.2**	**57.0**	**77.4**	**82.5**

InferFix: End-to-End Program Repair with LLMs over Retrieval-Augmented Prompts

Matthew Jin
Microsoft
Redmond, WA, USA

Syed Shahriar
UCLA
Los Angeles, CA, USA

Michele Tufano
Microsoft
Redmond, WA, USA

Xin Shi
Microsoft
Redmond, WA, USA

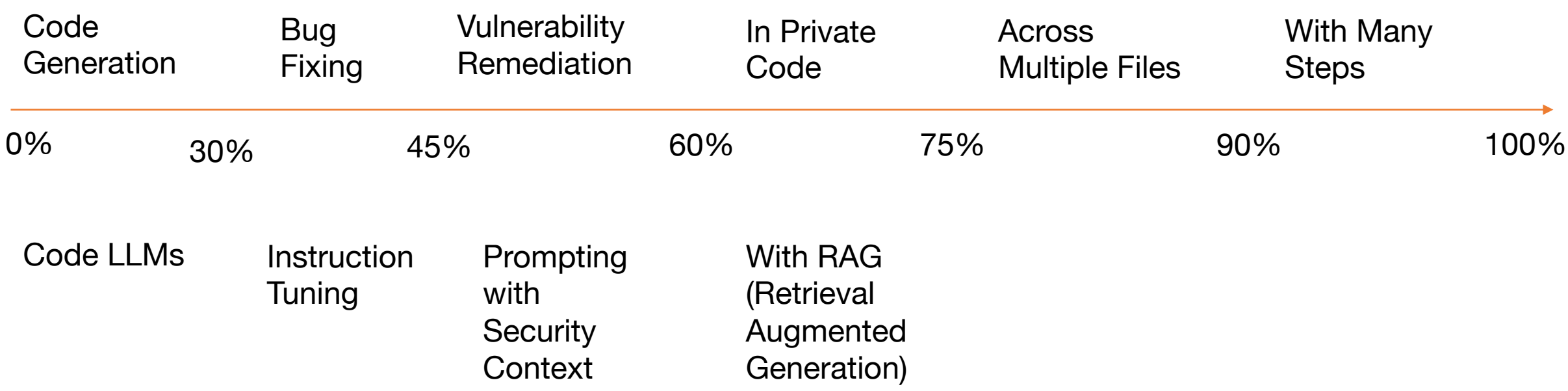
Shuai Lu
Microsoft Research
Beijing, China

Neel Sundaresan
Microsoft
Redmond, WA, USA

Alexey Svyatkovskiy
Microsoft
Redmond, WA, USA

https://arxiv.org/abs/2303.07263

Fixing Software Vulnerabilities



Static Analysis-augmented Generation (SAG)

```
package com.adobe.acs.commons.models.injectors;
import com.adobe.granite.xss.XSSAPI;
import com.day.cq.wcm.api.Page;
import com.day.cq.wcm.api.PageManager;
...
public class DefineObjectsInjector implements Injector {

    private static Designer getDesigner(Object adaptable) {}

    private ResourceResolver getResourceResolver(Object adaptable) {
        if (adaptable instanceof SlingHttpServletRequest) {
            return ((SlingHttpServletRequest)adaptable).getResourceResolver();
           }
        if (adaptable instanceof Resource) {
            return ((Resource)adaptable).getResourceResolver();
           }
        return null;
    }
```

```
    private Designer getDesigner(Object adaptable) {
        <START_BUG>
        return getResourceResolver(adaptable).adaptTo(Designer.class);
        <END_BUG>
    }
```

```
### Instruction:
<error message>

Fix vulnerability <CWE> in

<vulnerable line(s)>

### Input:
related_code

vulnerable_code

### Response:
```

SAG

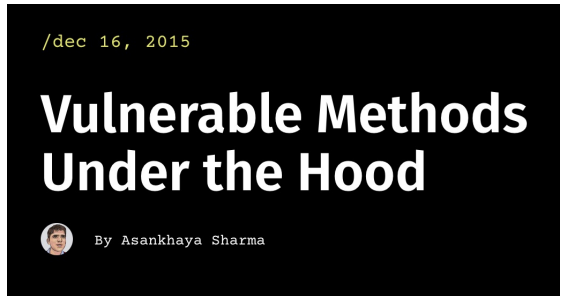
1. Reachability analysis
2. Impact analysis

```
607     -       private byte[] crypt_raw(byte password[], byte salt[], int log_rounds) {
610     +       public byte[] crypt_raw(byte password[], byte salt[], int log_rounds,
611     +           int cdata[]) {
```

Vulnerable Method crypt_raw has the following call chain

Class Name: org/mindrot/jbcrypt/BCrypt
Method Name: crypt_raw
Parameters: ([B[BI)
1 Path(s)

Paths	Line #
LChrisMethodsTest;->**main**([Ljava/lang/String;)	23
LChrisMethodsTest;->**execute**(Ljava/lang/String;)	44
Lorg/mindrot/jbcrypt/BCrypt;->**checkpw**(Ljava/lang/String;Ljava/lang/String;)	763
Lorg/mindrot/jbcrypt/BCrypt;->**hashpw**(Ljava/lang/String;Ljava/lang/String;)	692
Lorg/mindrot/jbcrypt/BCrypt;->**crypt_raw**([B[BI)	



https://www.veracode.com/blog/managing-appsec/vulnerable-methods-under-hood

Type	Library	From	To	Breaking
MAVEN	`commons-fileupload:commons-fileupload`	1.3.2	1.5	No
MAVEN	`org.keycloak:keycloak-saml-core`	1.8.1.Final	2.5.5.Final	No
MAVEN	`org.apache.commons:commons-collections4`	4.0	4.1	No
MAVEN	`org.mindrot:jbcrypt`	0.3m	0.4-atlassian-1	No
MAVEN	`mysql:mysql-connector-java`	5.1.48	8.0.28	Yes

Efficient Static Checking of Library Updates

Darius Foo
CA Technologies
Singapore
darius.foo@ca.com

Hendy Chua
CA Technologies
Singapore
hendy.chua@ca.com

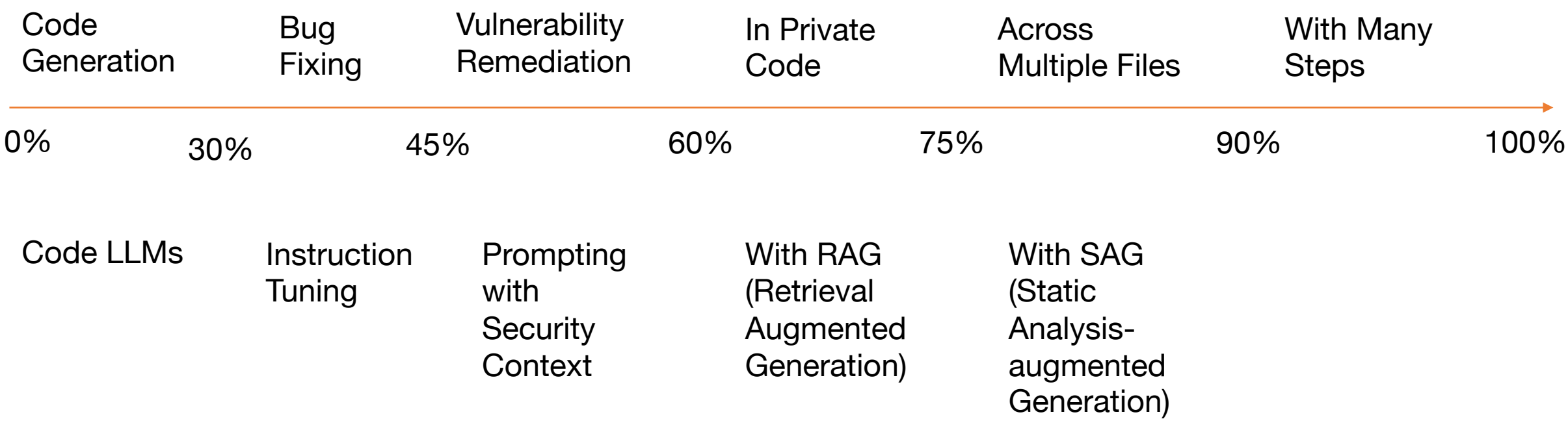
Jason Yeo
CA Technologies
Singapore
jason.yeo@ca.com

Ang Ming Yi
CA Technologies
Singapore
mingyi.ang@ca.com

Asankhaya Sharma
CA Technologies
Singapore
asankhaya.sharma@ca.com

https://dl.acm.org/doi/10.1145/3236024.3275535

Fixing Software Vulnerabilities



Static Analysis-augmented Generative Agents (SAGA)

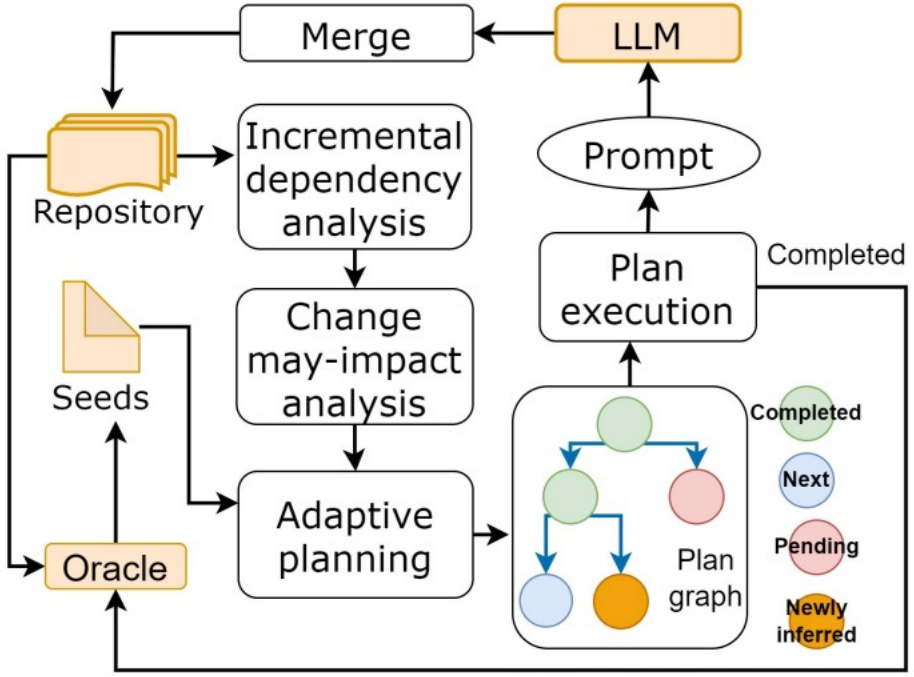


Fig. 2. Overview of CodePlan.

CodePlan: **Repository-level Coding using LLMs and Planning**

RAMAKRISHNA BAIRI, Microsoft Research, India
ATHARV SONWANE, Microsoft Research, India
ADITYA KANADE, Microsoft Research, India
VAGEESH D C, Microsoft Research, India
ARUN IYER, Microsoft Research, India
SURESH PARTHASARATHY, Microsoft Research, India
SRIRAM RAJAMANI, Microsoft Research, India
B. ASHOK, Microsoft Research, India
SHASHANK SHET, Microsoft Research, India

https://arxiv.org/abs/2309.12499

Prompt Template

p1 Task Instructions: *Your task is to . . .*

p2 Earlier Code Changes (Temporal Context): *These are edits that have been made in the code-base previously -*

```
Edit 1:
 Before: «code_before»
 After: «code_after»
...
```

p3 Causes for Change: *The change is required due to -*

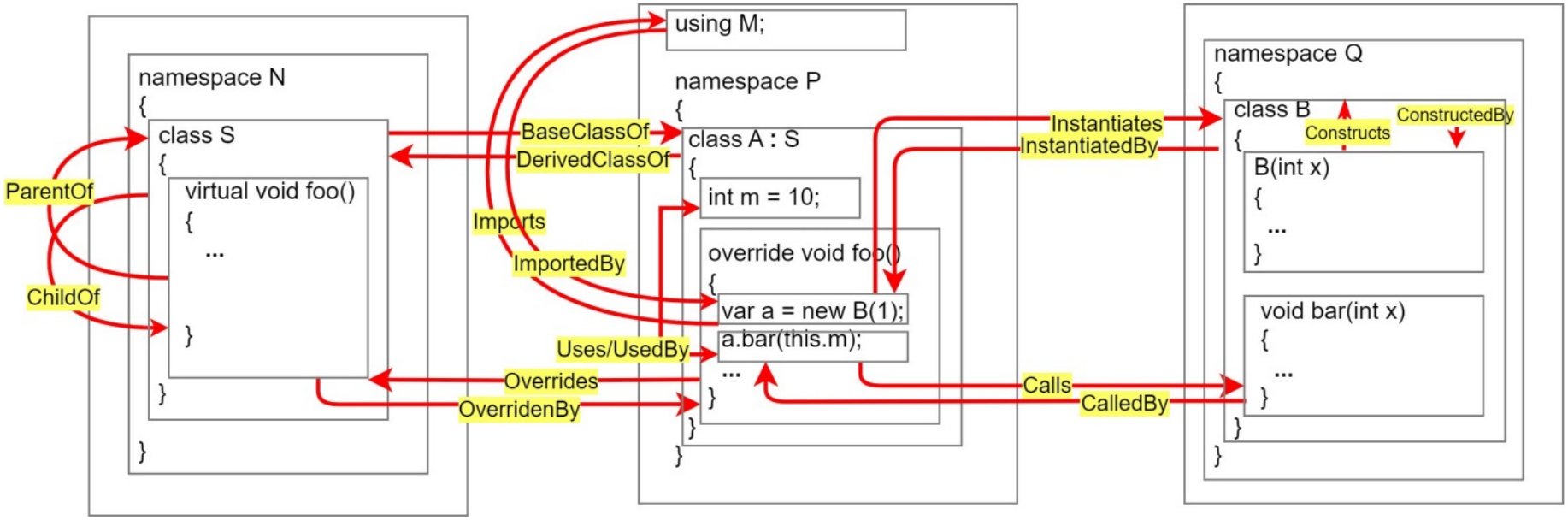
```
«code_to_be_edited» is related to «code_changed_earlier» by «cause»
...
```

p4 Related Code (Spatial Context): *The following code maybe related -*

```
«related_code_block-1»
...
```

p5 Code to be Changed Next: *The existing code is given below -*

```
«code_to_be_edited»
```


namespace N
{
class S
{
virtual void foo()
{
...
}
}
}
using M;
namespace P
{
class A : S
{
int m = 10;
override void foo()
{
var a = new B(1);
a.bar(this.m);
...
}
}
}
namespace Q
{
class B
{
B(int x)
{
...
}
void bar(int x)
{
...
}
}
}
ParentOf
ChildOf
BaseClassOf
DerivedClassOf
Imports
ImportedBy
Uses/UsedBy
Overrides
OverridenBy
Instantiates
InstantiatedBy
Constructs
ConstructedBy
Calls
CalledBy

```
public class SyncSubscriberTest : SubscriberBlackboxVerification<int?>
{
    public SyncSubscriberTest() : base(new TestEnvironment())
    {
    }

    public override ISubscriber<int?> CreateSubscriber() => new Subscriber();

    private sealed class Subscriber : SyncSubscriber<int?>
    {
        private long _acc;

        public override void OnComplete() => Console.WriteLine("Accumulated: " + _acc);
    }
}
```

1

```
public class SyncSubscriberTest : SubscriberBlackboxVerification<int?>
{
    public SyncSubscriberTest() : base(new TestEnvironment())
    {
    }

    public override ISubscriber<int?> CreateSubscriber() => new Subscriber();

    private sealed class Subscriber : SyncSubscriber<int?>
    {
        private long _acc;
        private readonly ITestOutputHelper _output;

        public override void OnComplete() => _output.WriteLine("Accumulated: " + _acc);

    }
}
```

2

```
public class SyncSubscriberTest : SubscriberBlackboxVerification<int?>
{
    public SyncSubscriberTest() : base(new TestEnvironment())
    {
    }

    public override ISubscriber<int?> CreateSubscriber() => new Subscriber();

    private sealed class Subscriber : SyncSubscriber<int?>
    {
        private long _acc;
        private readonly ITestOutputHelper _output;

        public Subscriber(ITestOutputHelper output)
        {
            _output = output;
        }

        public override void OnComplete() => _output.WriteLine("Accumulated: " + _acc);
    }
}
```

3

```
public class SyncSubscriberTest : SubscriberBlackboxVerification<int?>
{
    private readonly ITestOutputHelper _output;

    public SyncSubscriberTest() : base(new TestEnvironment())
    {
    }

    public override ISubscriber<int?> CreateSubscriber() => new Subscriber(_output);

    private sealed class Subscriber : SyncSubscriber<int?>
    {
        private long _acc;
        private readonly ITestOutputHelper _output;

        public Subscriber(ITestOutputHelper output)
        {
            _output = output;
        }

        public override void OnComplete() => _output.WriteLine("Accumulated: " + _acc);
    }
}
```

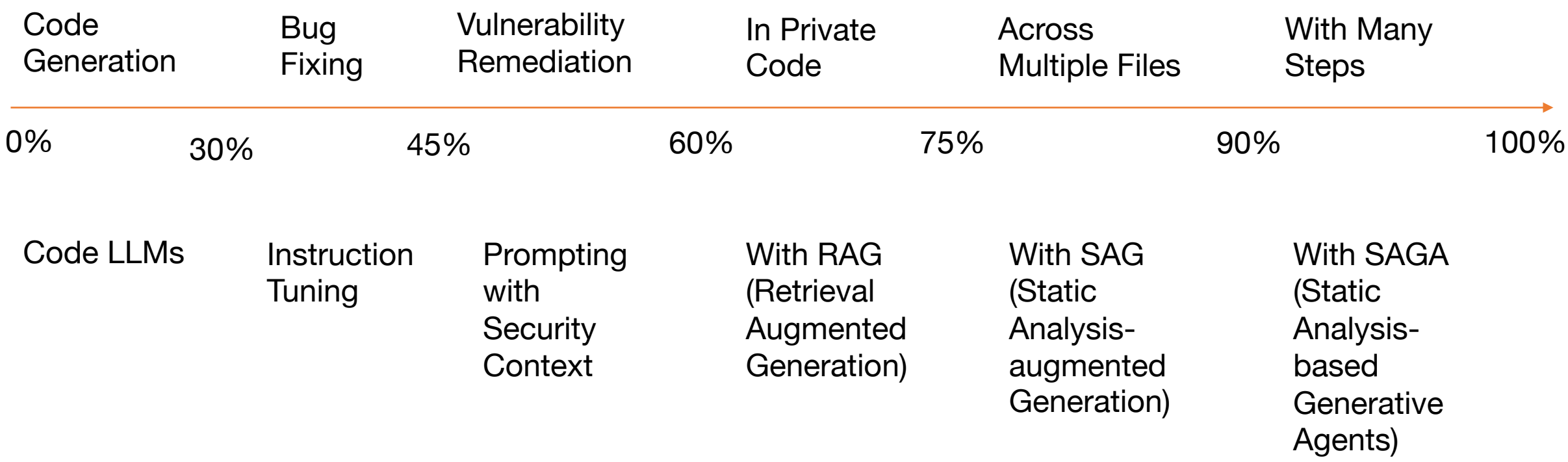
4

```
public class SyncSubscriberTest : SubscriberBlackboxVerification<int?>
{
    private readonly ITestOutputHelper _output;

    public SyncSubscriberTest(ITestOutputHelper output) : base(new TestEnvironment(output))
    {
        _output = output;
    }

    public override ISubscriber<int?> CreateSubscriber() => new Subscriber(_output);

    private sealed class Subscriber : SyncSubscriber<int?>
    {
        private long _acc;
        private readonly ITestOutputHelper _output;

        public Subscriber(ITestOutputHelper output)
        {
            _output = output;
        }

        public override void OnComplete() => _output.WriteLine("Accumulated: " + _acc);
    }
}
```

1. Console.WriteLine is migrated to ITestOutputHelper.WriteLine. This adds a member _output to Subscriber class
2. CodePlan's change-may-impact analysis detects addition of a new field and propagates the changes to the constructor of Subscriber through next
3. CodePlan's change-may-impact analysis detects changes the signature of Subsriber's constructor and propagates the changes to the instantiation of
4. CodePlan's change-may-impact analysis detects addition of a new field and propagates the changes to the constructor of SyncSubscriberTest through next LLM call. The tempral/spatial context also includes the change that

Note, Build-Repair stops after step 1, since there are no build errors. Hence fails to execute the changes in steps 2,3, and 4.

Fixing Software Vulnerabilities

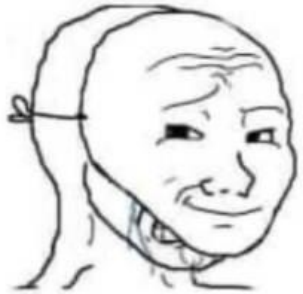


Developer Less Security

(The Prestige)

- Patched Coder
- Static Analysis Eval

Artists:



ai art will replace us

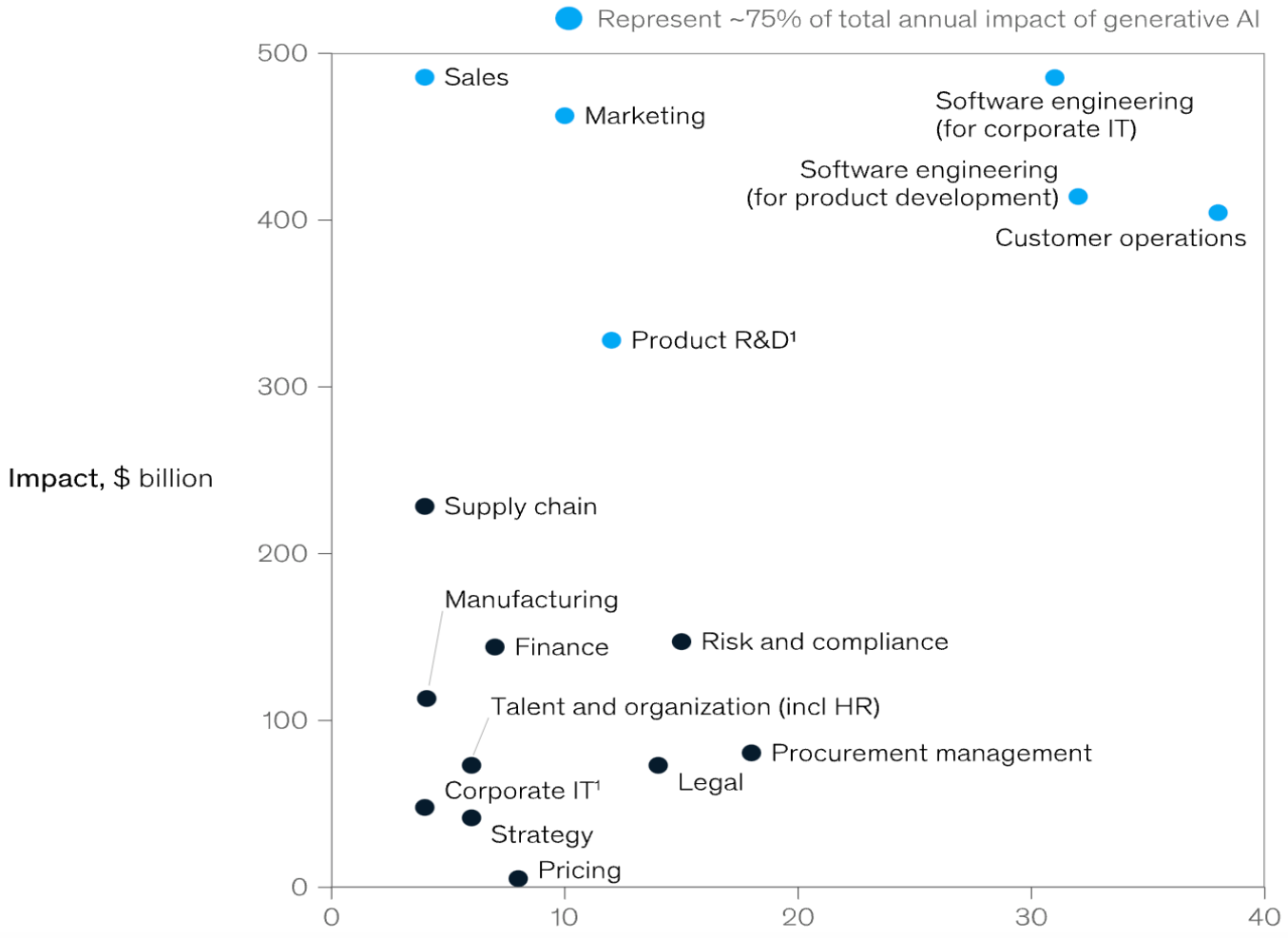
nooooooooooo

programmers:

ChatGPT will replace us

finally.

Using generative AI in just a few functions could drive most of the technology's impact across potential corporate use cases.



https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#business-value


DevSecOps
DevLess
Security


OWASP 2023
GLOBAL
AppSec
SINGAPORE
VIRTUAL CONFERENCE
OCTOBER 3-5
THANK YOU