Microsoft

Orca

Differential Bug Localization in Large-Scale Services

Ranjita Bhagwan, Rahul Kumar, Chandra Maddila, Adithya Philip
Microsoft Research India

Devops Pipeline for Orion

Code

1000s of devs,
100s of commits/day

Build

Dozens of
builds/week

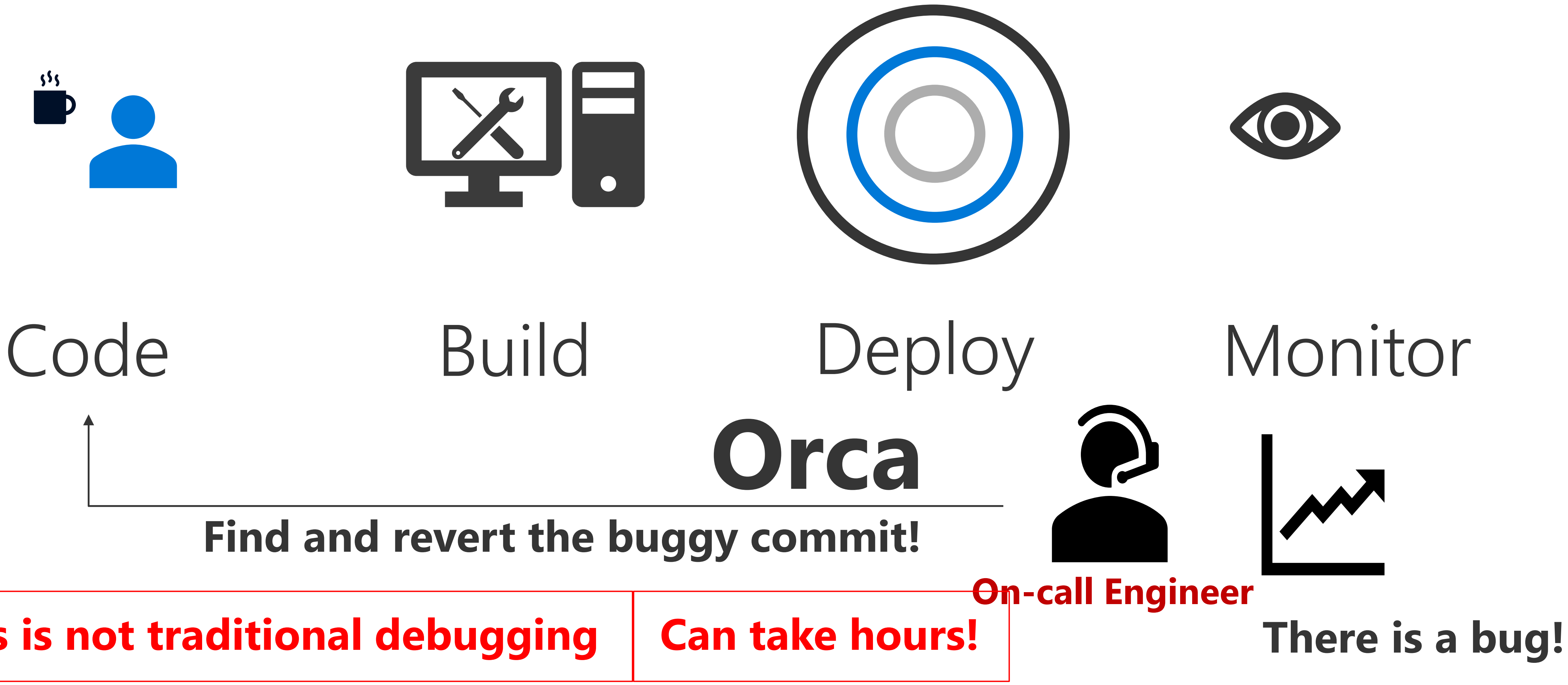
Deploy

"Ring"-based
deployment

Monitor

Fine-grained
monitoring

Commit-level Bug Localization



Code

Build

Deploy

Monitor

Orca

Find and revert the buggy commit!

On-call Engineer

There is a bug!

This is not traditional debugging

Can take hours!

State-of-the-art

- Look for related bugs [Lo et al., Srinivasa et al.]
  - Services are very dynamic and continuously changing. Similarity is limited!
- Instrument code and collect data [Liblit et al.]
  - Cannot disrupt current procedures
- Use extensive static/dynamic analysis [Lal et al.]
  - Too resource-intensive, the odd bug WILL slip through!

Post-Deployment Bugs: Observation

Textual similarity between symptom and code changes

Client-side

```
Operation not supported
   for type MailId
```

Support for datatype **MailId** added
on server-side but not on a particular client

Server-side

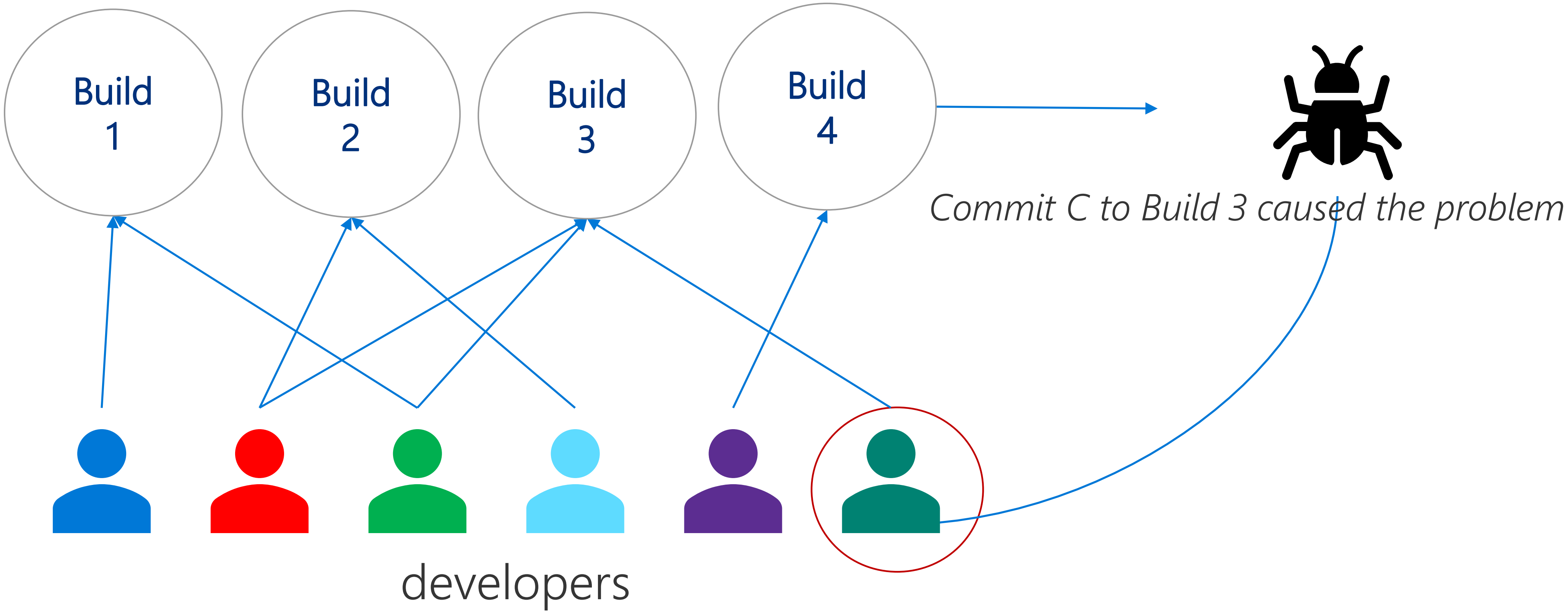
```
Public MailSession GetMailSession(Object o, ClientType c,  bool returnMailIds)
```

Post-Deployment Bugs: Challenge

Textual similarity occurs between symptom and CLUSTER of commits made around the same time.

Post-Deployment Bugs: Observation

Buggy commit may not be part of symptomatic build



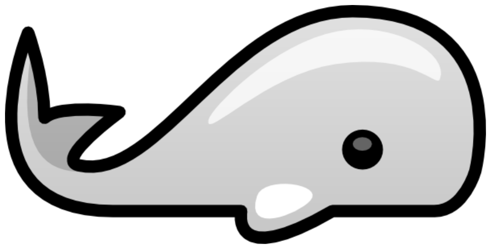
Orca: Search Engine for Localizing Bugs

On-call Engineer

1. Differential Code Analysis of the Abstract Syntax Tree

Customer complaints
Alerts
Log error messages
Exceptions

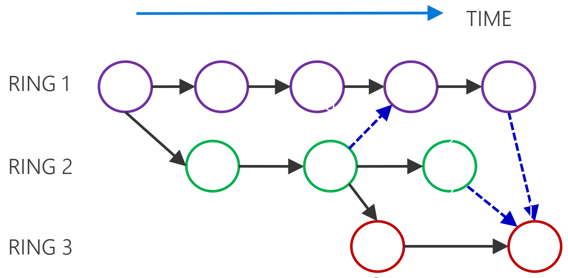
Documents



Queries

Ranking

2. Build Provenance Graph



3. Commit Risk Estimation
Term relevance

1. Differential Code Analysis

Before

```
Namespace Storage {
  ...
  class Connection {
    ...
    public Transaction BeginTx() {
    return Transaction.New(this);
}
```

After

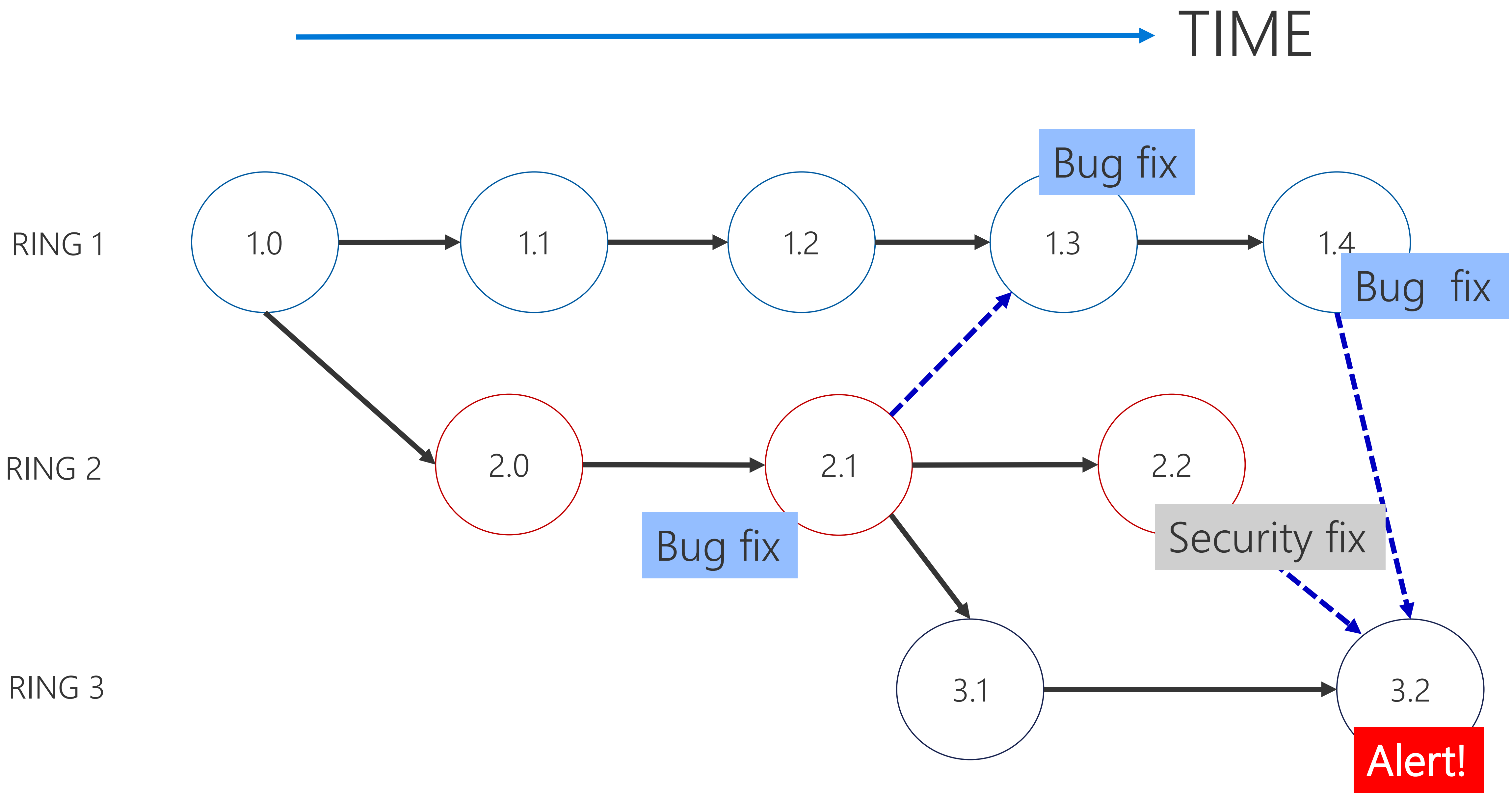
```
Namespace Storage {
  ...
  class Connection {
    ...
    public Transaction BeginTx() {
      if (this.filter != null)    {
      Transaction.MailFilter.Apply();

this.source.PerfCounters.UpdateMailFilterCt
();
    }
    return Transaction.New(this);
}
```

Namespace, Class, Method Changed: Storage, Connection., BeginTx

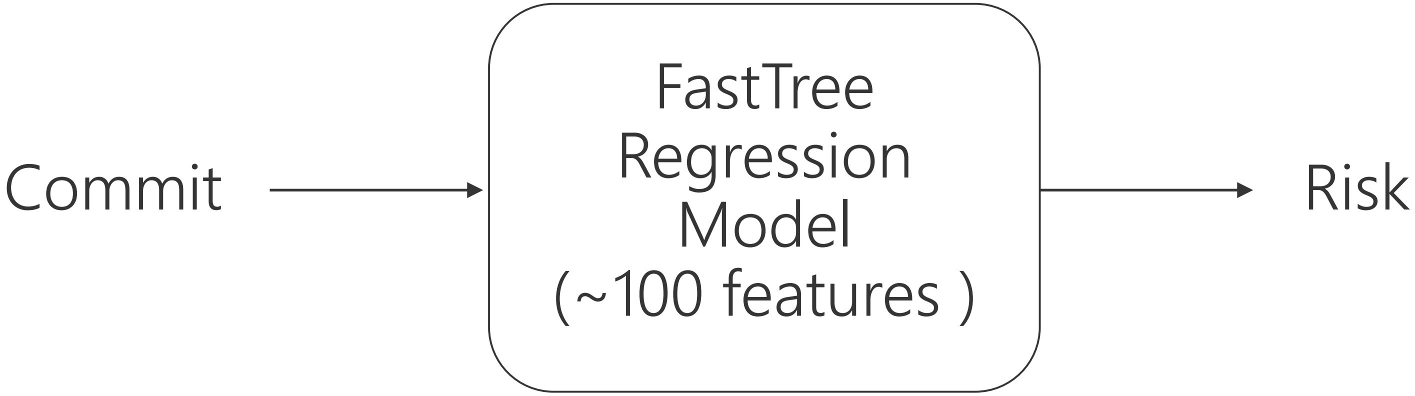
Custom tokenizer extracts important Keywords: `Storage, Mail, Tx, MailFilter,`

2. Build Provenance Graph

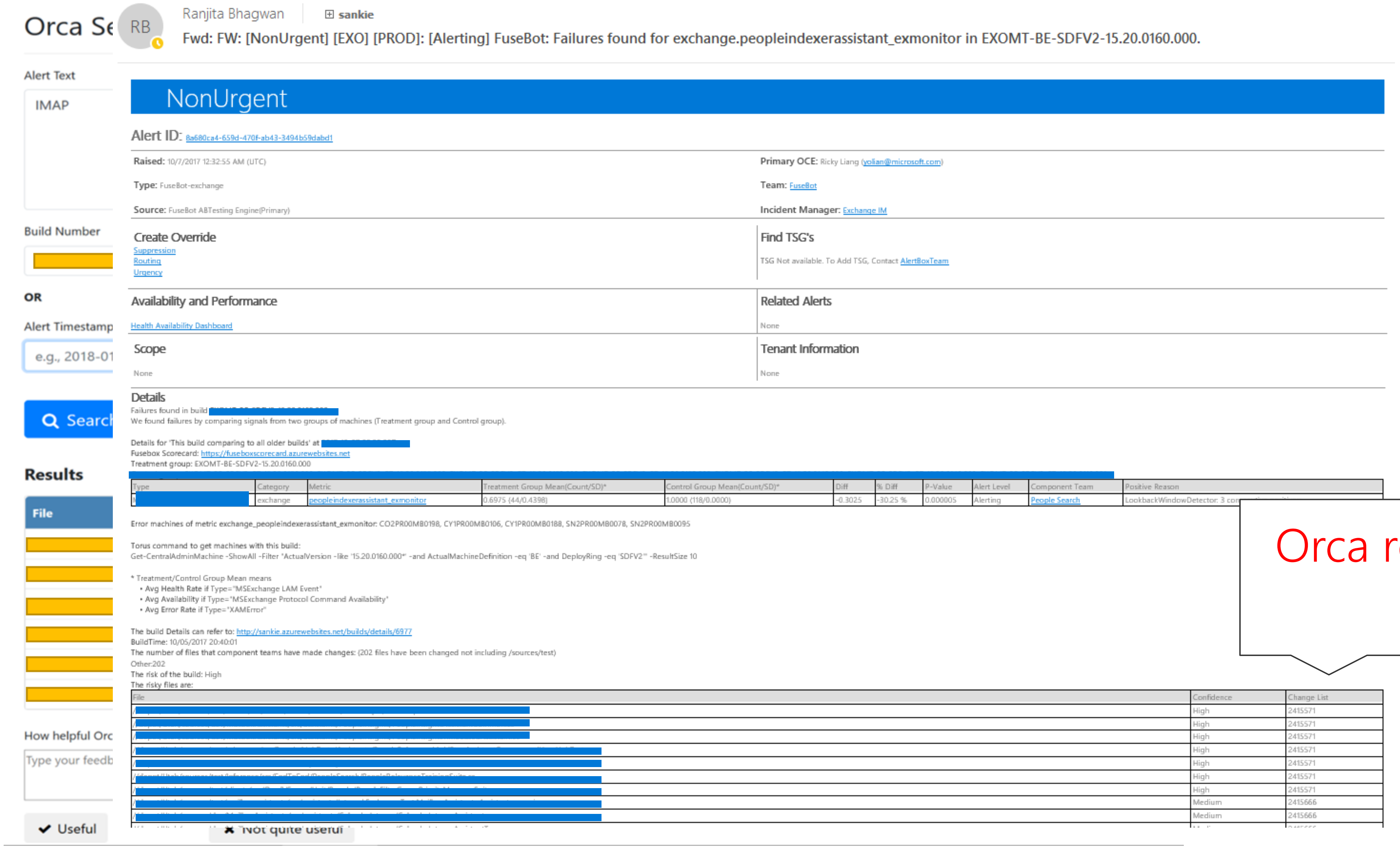


3. Commit Risk Estimation

- Commits to some codepaths are more likely to cause bugs
- A developer with < 10 total commits is twice as likely to make a buggy commit
- Buggy commits had twice as many reviewer-comments as safe commits



Presenting Orca Results



[redacted]

[redacted]

Ranjita Bhagwan | ⊞ sankie

Fwd: FW: [NonUrgent] [EXO] [PROD]: [Alerting] FuseBot: Failures found for exchange.peopleindexerassistant_exmonitor in EXOMT-BE-SDFV2-15.20.0160.000.

NonUrgent

Alert ID: 8a680ca4-659d-470f-ab43-3494b59dabd1

Raised: 10/7/2017 12:32:55 AM (UTC)

Type: FuseBot-exchange

Source: FuseBot ABTesting Engine(Primary)

Primary OCE: Ricky Liang (yolian@microsoft.com)

Team: FuseBot

Incident Manager: Exchange IM

Create Override

Suppression
Routing
Urgency

Find TSG's

TSG Not available. To Add TSG, Contact AlertBoxTeam

Availability and Performance

Health Availability Dashboard

Related Alerts

None

Scope

None

Tenant Information

None

Details

Failures found in build [redacted]
We found failures by comparing signals from two groups of machines (Treatment group and Control group).

Details for 'This build comparing to all older builds' at [redacted]
Fusebox Scorecard: https://fuseboxscorecard.azurewebsites.net
Treatment group: EXOMT-BE-SDFV2-15.20.0160.000
[redacted]

Type	Category	Metric	Treatment Group Mean(Count/SD)*	Control Group Mean(Count/SD)*	Diff	% Diff	P-Value	Alert Level	Component Team	Positive Reason
[redacted]	exchange	peopleindexerassistant_exmonitor	0.6975 (44/0.4398)	1.0000 (118/0.0000)	-0.3025	-30.25 %	0.000005	Alerting	People Search	LookbackWindowDetector: 3 con

Error machines of metric exchange_peopleindexerassistant_exmonitor: CO2PR00MB0198, CY1PR00MB0106, CY1PR00MB0188, SN2PR00MB0078, SN2PR00MB0095

Torus command to get machines with this build:

```
Get-CentralAdminMachine -ShowAll -Filter "ActualVersion -like '15.20.0160.000*' -and ActualMachineDefinition -eq 'BE' -and DeployRing -eq 'SDFV2'" -ResultSize 10
```

* Treatment/Control Group Mean means
- Avg Health Rate if Type="MSExchange LAM Event"
- Avg Availability if Type="MSExchange Protocol Command Availability"
- Avg Error Rate if Type="XAMError"

The build Details can refer to: http://sankie.azurewebsites.net/builds/details/6977
BuildTime: 10/05/2017 20:40:01
The number of files that component teams have made changes: (202 files have been changed not including /sources/test)
Other:202
The risk of the build: High
The risky files are:

File	Confidence	Change List
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	High	2415571
[redacted]	Medium	2415666
[redacted]	Medium	2415666

Orca results

Orca Evaluation

Manually curated a list of $\boldsymbol{Q}$=48 code-related incidents

Mean Reciprocal Rank (MRR) = $\dfrac{1}{|Q|}\displaystyle\sum_{i=1}^{|Q|}\frac{1}{\mathbf{rank}_i}$

Metric	Results for top-10 ranking		
	Diff on AST	Build Graph	Commit Risk
Recall	33	37	37
MRR	0.44	0.38	0.42

Orca Evaluation

Orca reduced the average number of commits that an On-Call Engineer has to examine by a factor of 3

“Awesome! And using this tool today we found [a] regression very quickly😊”

Project Sankie

www.microsoft.com/en-us/research/project/sankie

Code

CHLR: Commit Risk Prediction

REX: Related Entities

Reviewer **Sugg**estions

Test/Build

FastLane: Data-driven Test Minimization

Rings

Deploy

Dotributor: Attributing Deployment inefficiency

Monitor

ORCA: Bug Localization

Decaf: Explaining Latency Anomalies

Summary

- Orca is a commit-level bug localization tool

- Orca uses differential code analysis, build provenance graph, commit risk estimation

- On-call Engineers have found Orca useful: it finds the buggy commit is 77% of cases.

Thank you. Questions?

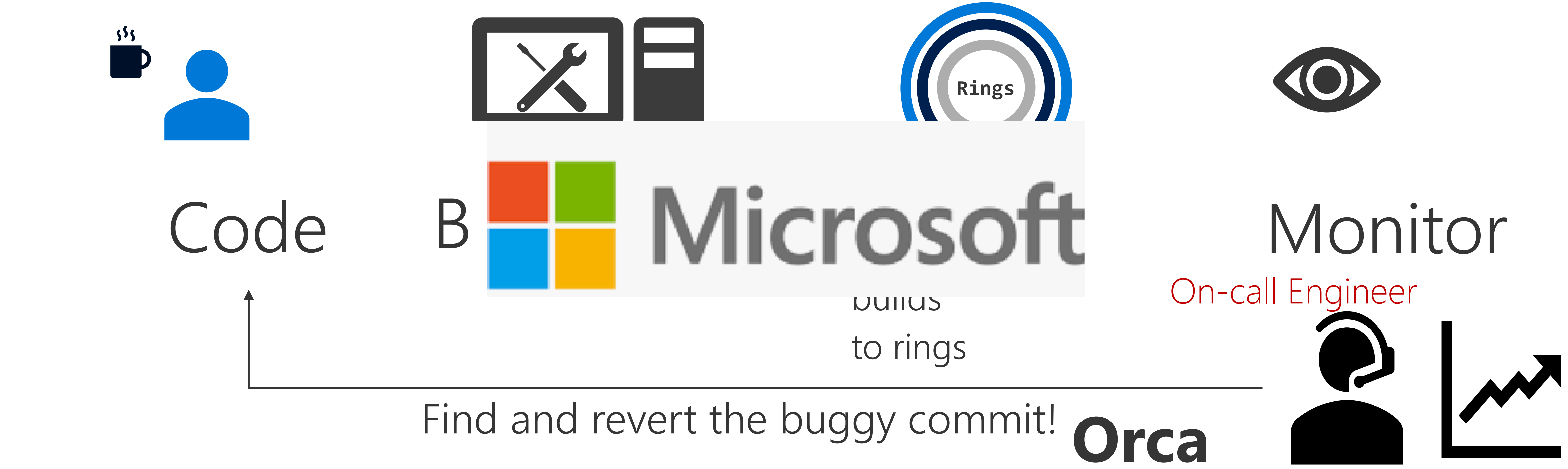
Devops Pipeline for Orion

1000s of devs,
100s of commits/day

Testing severely
restricted

Dozens of
deployments/week

Fine-grained
monitoring



This is not debugging

Can take hours!

Anomalies tell us
It is a code issue