Implementing Site Reliability Engineering (SRE) Reliability Best Practices

Implementing Effective Incident Response

Karun Subramanian
IT Operations Expert

@karunops www.karunsubramanian.com

Overview

SRE Overview

Design an effective on-call system

Understand managed vs. unmanaged incidents

Build and implement an effective postmortem process

Learn the tools and templates for postmortems

SRE Overview

> “Site Reliability Engineering (SRE) is what happens when you ask a software engineer to design an operations team.”

Benjamin Treynor Sloss (Founder of Google SRE)

Responsibilities of an SRE Organization

Availability

Performance

Incident management

Monitoring

Activities of an SRE

- **Write code**
- **Lead war room**
- **Automate**
- **Be on-call**
- **Perform postmortem**
- **Implement best practices**

Designing an Effective on-call System

On-call Engineer

- Protector of production systems
- Responds to emergencies within acceptable time
- Involves team members and escalates issues
- May work on non-emergencies such as email alerts
- Writes postmortems

Three Tenets of Effective on-call System

Engineering Focus

Spend only about 25% of time managing incidents

Balanced Workload

Avoid burnouts by designing proper rotations

Positive and Safe Environment

Clearly defined escalation and blameless postmortem procedures

Engineering Focus

Write Code

Engineers should be looking to design solutions rather than stitching up band-aids

Automate

Automation not only saves time, but reduces failures due to human errors

Balanced Workload

Multi-region support

- Avoid night-shifts if possible
- Caution: Handoffs and coordination can create overhead
- 6-8 engineers per site is ideal

Avoid operational underload

- Engineers can get out-of-touch with production systems

Compensate

- Comp time off
- Cash awards

Positive and Safe Environment

Incident management system

A well-defined procedure for handling significant incidents

Blameless postmortems

Postmortems should focus on root cause and prevention

Understanding Managed Vs. Unmanaged Incidents

Managed Vs. Unmanaged Incidents

Unmanaged

Typically led by the on-call engineer with random team members participating

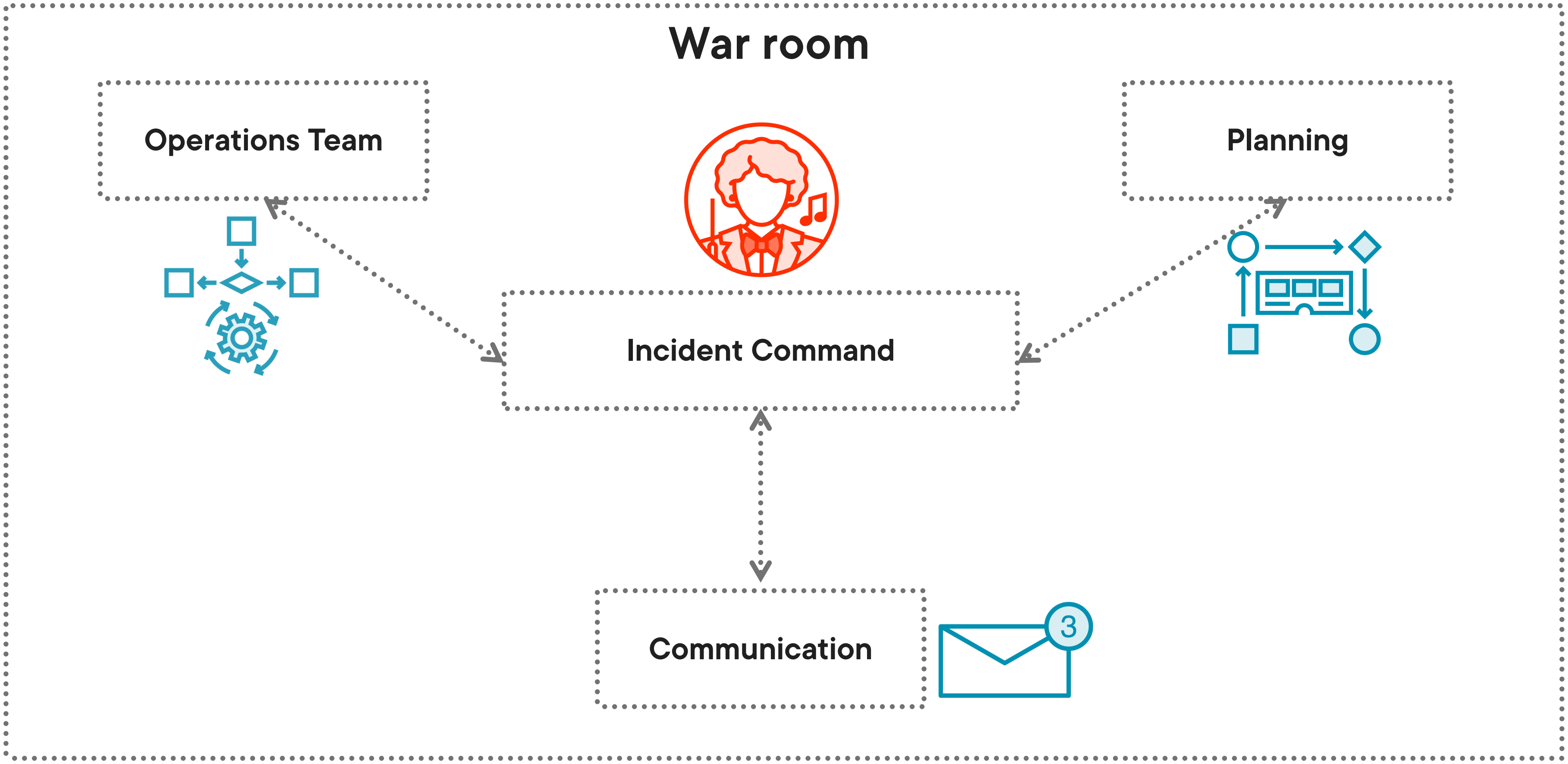
Managed

Led by incident command with clearly defined procedure and roles

Managed Vs. Unmanaged Incidents

Unmanaged	Managed
No clear roles	Clearly defined roles
No incident command	Incident command leads the resolution
Random team members involved (Freelancing)	Only the ops-team defined by the incident command can update systems
Poor (or lack of) communication	A dedicated role for communication
No central body that runs the troubleshooting	A recognized command post such as war room

Incident Management Process



War room

Operations Team

Planning

Incident Command

Communication

Incident Management Roles

Incident Command

Runs the war room, assigning responsibilities to others

Operations Team

Only role allowed to make changes to the system

Communication

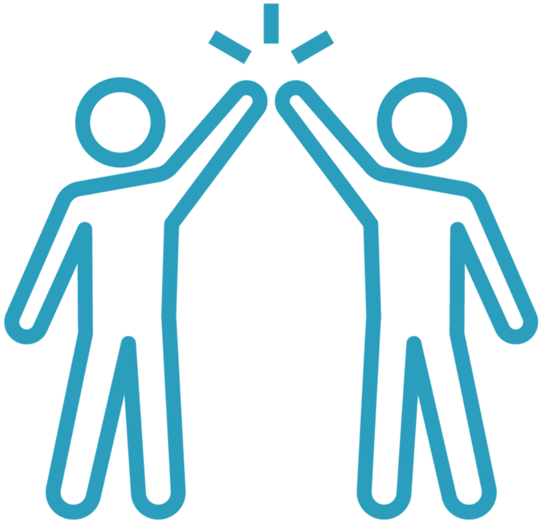
Periodic updates to stakeholders

Planning

Support operations by handling long-term items such as setting up bug fixes and postmortems

Building and Implementing an Effective Postmortem Process

Why Postmortem?



Fully understand/document the incident

What could have been done differently?

Root cause analysis

Learn from the incident

Opportunities for prevention

Plan and follow through assigned activities

Blameless Postmortem

- An important tenet of SRE
- No finger-pointing
- Focus is on systems and processes and not on individuals
- Isolating individuals/teams can create unhealthy culture
- Must call out where improvements can be made

When to Do a Postmortem?

End user experience impact beyond a threshold (SLO)

- Service unavailable
- Unacceptable performance
- Erratic functionality

Data loss

Organization/group specific

Content of a Postmortem

- Summary
- Impact (include any financial impact)
- Root cause(s)
- Resolution
- Monitoring (How was the issue detected?)
- Action items with due dates and owners

Supervisor or senior team member(s) must review postmortems before publishing

Learning the Tools and Templates for Postmortems

Tools

Existing ITSM tools

Servicenow, Remedy, Atlassian ITSM

Opensource

https://github.com/etsy/morgue

Develop your own

Remember: SREs are also software engineers!

Google

Pagerduty

Atlassian

Victorops

Your own

Templates

Demo

Walk through a postmortem

Review postmortem templates

- Google
- Atlassian

Summary

Effective on-call system is necessary to ensure service availability and health

Balance workload for on-call engineers

- Allocate resources
- Use multi-region support
- Promote safe and positive environment

Incident management must facilitate clear separation of duties

- Incident command, operations, planning and communication

Blameless postmortems help prevent repeated incidents

Up Next:

Implementing Effective Change Management