Playbook Methods Repository

# **Performance and Error Monitoring**

Implement technologies and processes to streamline real-time monitoring, tracing, and alerting of errors and performance bottlenecks across the tech stack. This might include capabilities such as enriched stack traces, distributed querying of event data, visualisations of infrastructure resources, etc.

### Remote Agility: **•** High

### Linked Tactic(s): Release Management

## Why we do it:

Since we've come a long way, customers have pretty high expectations for software and applications. Many customers tend not to report bugs or issues that arise. They simply move on to the next product. By monitoring errors, you can identify and fix problems before they occur, or at the very least as soon as they occur, which in turn helps reduce customer outrage and increase uptime.

If your outsourced infrastructure and software slows or fails, it can also have a huge impact on your business. Speed isn't just a nice-to-have quality today, it's a must-have quality. The need for error monitoring tools is therefore extremely high since it can directly affect the user.

By using tools, you can prevent these errors from occurring in the first place. With custom keys and logs, you can debug crashes faster and gain insight into users. The data can be visualized as well as you can receive notifications in real-time.

## 

## When to apply it:

## Best Practices & Considerations:

* Monitoring frontend interactions should be user-centric by monitoring browser events and tracking individual user interactions over time.
* Logs can help you gather valuable information before a crash. Depending on the user's activity, it can be anything from behavior (ex. user clicks download button) to details about an action (ex. image downloaded, image downloaded from).
* While code quality is critical, you must monitor a variety of physical components and networking settings that are used to run your application. Additionally, with hybrid cloud systems, you may want cloud monitoring capabilities. That is why, in order to keep on top of your applications, you need to combine server and infrastructure monitoring with Azure/AWS monitoring and database performance monitoring capabilities.
* Have a simple and holistic perspective by selecting key indicators of performance and utilization. Visual dashboards that combine infrastructure and application metrics simplify correlating and evaluating data.
* Integrating your APM tool with the CI/CD workflow adds another layer of dependability and improves test coverage. The integration enables you to execute performance tests and resolve application problems prior to deploying your application to the production environment.

## Responsible roles:

* Software Engineer

## Tools:

### Online tools/platforms/services

### Crashlytics, Sentry, bugsnag, dynatrace Instabug, Rollbar

### Websites

* + xx

### Databases

* + xx
* Other
  + xx

## 

## Thoughtworks Examples - Linked

### Client working docs, airtable, miro/mural boards

* + xx

### Client polished presentations/deliverables

* + xx

### Internal assets - clinic materials / guild docs

* + xx

## 

## Learn more: How we do this?

### How-To Resources (external or internal)

* + <https://www.softwaretestinghelp.com/application-performance-monitoring-in-devops/>
  + <https://www.datadoghq.com/blog/modern-frontend-monitoring/>
  + <https://firebase.googleblog.com/2017/09/7-tips-for-getting-most-out-of.html>
  + <https://www.techtarget.com/searchenterprisedesktop/definition/Application-monitoring-app-monitoring>
  + <https://www.tek-tools.com/apm/open-source-apm-tools>

## 