# OnCall Handbook for Koorier IT Team

**Objective** : This document focuses on bringing a uniform approach to investigate, resolve and retrospect production issues collectively by the development team.

**Why are we reading this** – As of today each team and team members has their own approach to trouble-shoot and investigate an ongoing production issues that results in (a) longer Investigation time (b) Breach of SLS’s (c) Lack or documentation to address similar issues in future.

**Proposal**

In order to eliminate on-going operational challenge for handling PROD issues, its recommended that

**Monitors and Alarms**

IT team should create a 3-tier monitor system for the services they own and publish on their home page (Wiki or Sharepoint)

1. **Primary indicators** – This dashboard should include time-series graph which captures CPU, Memory, API through-put/minute, API Latency, Fatal per minute (aggregate of 5xx exception in service logs).
2. **Secondary indicators** – This dashboard should include service specific time series graphs like concurrent manifest ingestion rate induction/sortation /min, package delivery/hr, etc. These metrics would indicate operational load on the PROD Kapp server. In case of deviations or spikes alarms can be configured to investigate the delta.
3. **Tertiary Monitors** – This dashboard can include service level communication to upstream or downstream services or cluster level assets requests sent to Here Maps/ Map box per minutes, Street Perfect reqs/ min, S3 uploads / min etc.

**Identifying problems**

OnCall engineer should capture level 1 data for the issues through the above monitoring system. As a level 2 analysis logs captured in PROD server should be referenced to isolate the problem and cause.

**Building the Solution**

Once the problem is identified a solution should be created on Patch branch, duly code-reviewed and pushed to the branch *(\*\* This section needs elaboration)* for pipeline deployment. Based on the severity of the issue and customer impact fix could be directly patched to the PROD server or sent through pipeline which takes usually xxx days of deployment (Journey from pre-prod/integ to QA to PROD).

**OnCall Rotation and Retrospective**

This section outlines the method to collectively develop a knowledge base and a standard operating procedure for handling a PROD ticket withing the development team.

On the last day of OnCall roster rotation a retrospective meeting should be organized and headed by the current oncall engineer. It should be mandatorily attended by next onCall engineer and entire Development team. The agenda of the meeting should be

1. **Discuss about all Tickets received during the week** – Current onCall should share all the tickets received during the week and elaborate on resolved, closed and open tickets with root cause analysis. This will (1) help the next onCall to know about all open tickets from last week. (2) Entire dev team would be aware of the issues and its RCA
2. **Fixes and their current deployment stage** – All the tickets resolved by the current onCall should be detailed out along with RCA and deployment stage.
3. **RCA’s that point to** - Need for new alarms or matrices or technical debt or design short-coming in the service. This should lead to an AI for creating JIRA/EPIC by taking the leadership in confidence by the team.
4. **New feature/bug deployment during the week** – The onCall engineer should update the next onCall and the team for any deployments or changes that happened during the current week and its effect on the service during observation window.
5. **Review open TT queue** - The team should collectively review the queue and take action through inclusion of some high priority tickets in the sprint plan of the OnCall engineer’s plate is projected full.

**Conclusion**

The development team owns the design, develop, deploy and maintenance of a cloud software. By introducing/piloting the above method it would help bring (a) A common run-book and standardized way of solutioning (c) Reduced time to identify and fix a PROD issues (d) Collective retrospectives would help identify and address core issues in the service.