



holibob



Incident and Release severity level

Version: 3



15 April 2024

Table of Contents

1 Introduction	1
2 Incidents	1
3 Risks	2



1 Introduction

This document formalises the rules used to assess the severity of a production incident or production release.

- Incidents can have a level P1 - P3
- Deployments can have a level R1 - R5 (where R=Risk)

2 Incidents

Any incident that qualifies for reporting per the table below **MUST** be created in the Teams Incident reporting channel.

This channel will then be used to report all actions taken to first mitigate and then resolve the issue. The incident will eventually be marked as CLOSED.

A post-incident review will be carried out by Tech and written to the incident log with a focus on learnings and process changes that may mitigate similar issues in future.

Any P1 incident must be reported by voice or text communication to the most senior resource available in the tech team. If it is reported by text the reporter **MUST** receive an acknowledgement from Tech that they are mounting a full response.

The level of severity of an incident controls the actions that will be taken by tech in investigating and resolving the incident.

P1 is the HIGHEST incident severity.

Severity	Definition	Actions
P1	<p>The issue is immediately apparent to ALL consumers of one or more White-Label or the main Product, Availability and Booking queries and mutations within the GraphQL-API.</p> <p>This includes any failure in the look to book process. ie if a consumer would be unable to find, book and pay for an experience.</p>	<p>Tech will immediately assess the issue.</p> <p>If root cause and an immediately deployable fix are not identified within 10 minutes of reporting then Tech will perform an immediate rollback of the relevant system(s) to the last known safe release.</p>

	If the issue relates to only one or a group of product then this is NOT a P1	AFTER successful rollback of the environment, Tech will identify root cause and work as required to test and then deploy a fix as soon as reasonable.
P2	The issue is immediately apparent in the White-Labels or GraphQL-API but impacts only a sub-set of products. The issue is immediately apparent to Partners when using Hub	Tech will immediately assess the issue. If root cause and an immediately deployable fix are not identified within 30 minutes of reporting then Tech will perform an immediate rollback of the relevant system(s) to the last known safe release.
P3	Everything else that is apparent in the White-Labels or GraphQL-API Anything that is affecting the production release of Hub but not directly impacting most partners	Tech will identify root-cause and deploy a fix as soon as reasonably possible within the normal hours for the team.

3 Risks

There is a risk associated with all software development! If we do nothing then generally we can expect all systems (subject to scale and infrastructure) to just keep doing the same thing for ever.

However, we are constantly developing new features and improving existing features. No matter whether this is a change to a single line of code or the introduction of a new feature with thousands of lines of code there is a risk of unexpected consequences, bugs or misunderstandings.

For large changes, we are now operating a reasonably extensive manual regression testing of the white-label sites and thus indirectly also testing the GraphQL code that supports the features of these sites. We have also introduced some manual regression testing of certain key features of Hub. A single run of the white-labels regression takes two staff two hours to complete and if hub is included then a further hour is consumed. It is not practical therefore to run full tests for very small changes.

We are working towards full-scale automation of both unit and regression testing but this is a massive effort. We expect to make very considerable steps to meet our goal by the end of 2021.

In the meantime, we must take a risk-based approach to release of code to production and the table below describes the methodology to assess risks and the level of testing that must be performed in advance of a production release.

R1 is the LOWEST level of risk

Risk	Definition	Process
R1	A minor bug fix generally addresses in only one or two lines of code and in a function that is isolated to a single page or API schema element	The release can be merged to master and deployed immediately to production without code review.
R2		
R3	The addition of entirely new functionality in the GQL or Hub and in some cases in Booking where this can reasonably be assessed as having zero impact on existing functionality.	The release must be developed and tested locally in a separate branch and the code must be peer-reviewed via a pull request before merge to master and development to production.
R4		
R5	Any significant change to high-level components such as database schema, context, security or functions, hooks etc that may be used across the entire architecture of a project.	As per R3. Additionally, the branch must be deployed to staging where the entire regression test is fully executed as documented and all tests pass. A release note must be published to the Team channel and other parts of the business given time to review and comment on the change. Additionally, the main functionality of WL sites should be confirmed post-deployment to production.