**Error Budget Policy**

**Context**

**Scope:** *Choose from* [ORG, TEAM, APP, SLO]

**Name:** *Org name* or *Team name* or *App name* or *SLOs*

**Errors**

These are the acceptable errors that will not consume the error budget:

1. Company wide network outages.
2. Cloud vendor outages.
3. Known downtimes for maintenance.
4. Miscategorized errors when no real users were impacted.
5. Errors consumed by users out of scope for the SLOs (e.g., load tests, integration tests).
6. Outages caused by a service maintained by another team (internal dependencies).
7. Outages caused by a service maintained by a 3rd party vendor (external dependencies).

Any other errors experienced by the users are taken out of the error budget.

**Alerts**

**Budget Window**

Rolling 30-days

**Alerts on Error Budget Consumption**

| **Budget Consumption** | **Alert Type** | **Outcome** |
| --- | --- | --- |
| >= 75% and < 90% | Warning | An investigation started; ticket created |
| >= 90% | Critical | Support team paged; an incident ticket created |

**Alerts on Elevated Burn Rate**

| **Alert Window** | **Budget Consumption** | **Exhaustion Time** | **Burn Rate Condition** | **Alert Type** | **Outcome** |
| --- | --- | --- | --- | --- | --- |
| 1 hour | 2% | 2 days, 2 hours | >= 14.4 | Critical | Support team paged; an incident ticket created |
| 6 hours | 5% | 5 days | >= 6 | Critical | Support team paged; an incident ticket created |
| 3 days | 10% | 30 days | >= 1 | Warning | An investigation started; ticket created |

**Policies**

**Happy State**

**Conditions:**

No alerts are fired, and error budget consumption is below 75%.

**Policy:**

* New feature developments are allowed.
* The team can take risks and do experimentations.
* If budget consumption is consistently hovering below 10%, SLOs can be tuned with stricter targets.

**Uncertain State**

**Conditions:**

Warning alerts are fired, but not critical alerts.

**Policy:**

* Risky deployments should wait until investigation is completed.
* Tickets should be created and prioritized for investigations.
* If system recovers and operational state conditions are met, follow the "Happy State" policy.

**Sad State**

**Conditions:**

Critical alerts are fired.

**Policy:**

(This is also known as "**Out of Budget Policy**")

* SRE team should be paged.
* An incident ticket is created.
* The application code should be frozen until the remaining error budget reaches at least 25% and all the alerts are cleared up.
* The product feature releases will be completely halted. Only exceptions that are allowed will be:
  + Fixes to address the root cause of SLO miss.
  + Fixes that have highest priority (e.g., Changes that involve legal ramifications if the deadlines are not met).
  + Security fixes.
* If SLO miss happened due to reasons out of team's control, adjust the error budget using the [status correction options](https://docs.datadoghq.com/api/latest/service-level-objective-corrections/) provided by Datadog.

**Outages Caused by Dependencies**

* Internal dependencies:
  + If an outage was caused by a service maintained by another service provider team, the dependent team should create a ticket with the service provider team to investigte the issue.
  + If the service provider team has SLOs and error budgets defined, the dependent team can forego out of budget policy as long as the service provider team has enacted their own error budget policies.
* External dependencies:
  + If the 3rd party vendor's outage broke the SLA, the team should engage with the vendor through proper channels to get compensated for the reliability miss.
  + An investigation ticket should be created internally to track progress.
* If a certain dependency has been identified as the root cause for recurring SLO misses, then consider removing that unreliable dependency from the system.
  + Since this can take long time, look for other measures to bring the error budget under control in the short term.