# Example Error Budget Policy

| **Status** | **Published** |
|---|---|
| Author | Steven Thurgood |
| Date | 2018-02-19 |
| Reviewers | David Ferguson |
| Approvers | Betsy Beyer |
| Approval Date | 2018-02-20 |
| Revisit Date | 2019-02-01 |

## Service Overview

The *Example Game Service* allows Android and iPhone users to play a game with each other. New releases of the backend code are pushed daily. New releases of clients are pushed weekly. This policy applies both to backend and client releases.

# Goals

The goals of this policy are to:

- Protect customers from repeated SLO misses
- Provide an incentive to balance reliability with other features

# Non-Goals

This policy is not intended to serve as a punishment for missing SLOs. Halting change is undesirable; this policy gives teams permission to focus exclusively on reliability when data indicates that reliability is more important than other product features.

# SLO Miss Policy

If the service is performing at or above its SLO, then releases (including data changes) will proceed according to the release policy.

If the service has exceeded its error budget for the preceding four-week window, we will halt all changes and releases other than P0[1] issues or security fixes until the service is back within its SLO.

Depending upon the cause of the SLO miss, the team may devote additional resources to working on reliability instead of feature work.

The team *must* work on reliability if:

- A code bug or procedural error caused the service itself to exceed the error budget.
- A postmortem reveals an opportunity to soften a hard dependency.
- Miscategorized errors fail to consume budget that would have caused the service to miss its SLO.

The team *may* continue to work on non-reliability features if:

- The outage was caused by a company-wide networking problem.
- The outage was caused by a service maintained by another team, who have themselves frozen releases to address their reliability issues.
- The error budget was consumed by users out of scope for the SLO (e.g., load tests or penetration testers).
- Miscategorized errors consume budget even though no users were impacted.

# Outage Policy

If a single incident consumes more than 20% of error budget over four weeks, then the team must conduct a postmortem. The postmortem must contain at least one P0 action item to address the root cause.

If a single class of outage consumes more than 20% of error budget over a quarter, the team must have a P0 item on their quarterly planning document[2] to address the issues in the following quarter.

# Escalation Policy

In the event of a disagreement between parties regarding the calculation of the error budget or the specific actions it defines, the issue should be escalated to the CTO to make a decision.

# Background

### Note

This section is boilerplate, intended to give a succinct overview of error budgets to those unfamiliar with them.

Error budgets are the tool SRE uses to balance service reliability with the pace of innovation. Changes are a major source of instability, representing roughly 70% of our outages, and development work for features competes with development work for stability. The error budget forms a control mechanism for diverting attention to stability as needed.

An error budget is 1 minus the SLO of the service. A 99.9% SLO service has a 0.1% error budget.

If our service receives 1,000,000 requests in four weeks, a 99.9% availability SLO gives us a budget of 1,000 errors over that period.

[1] P0 is the highest priority of bug: all hands on deck; drop everything else until this is fixed.

[2] At Google, quarterly planning is public, and teams are held accountable to their plans.

PREVIOUS
Appendix A - Example SLO Document

NEXT
Appendix C - Results of Postmortem Analysis