Engineer

Ціль рівня обслуговування

вимірювана ціль надійності, якої має досягати сервіс

Що це таке

Ціль рівня обслуговування — це вимога до того, наскільки добре працює сервіс, сформульована так точно, що будь-хто може перевірити, чи її виконано: частка запитів, які мають завершитися успішно або отримати відповідь досить швидко, за визначений період. Майже всі називають її коротко SLO, від англійського service level objective. Книга Google про Site Reliability Engineering визначає її як цільове значення або діапазон значень для певного рівня обслуговування, виміряного індикатором.

Цей індикатор — індикатор рівня обслуговування, або SLI: ретельно визначена величина, що описує один бік обслуговування, яке користувачі справді отримують. Найчастіше беруть затримки запитів, а одразу за ними йдуть частка помилок і пропускна здатність. SRE Workbook радить записувати кожен SLI як відношення: кількість добрих подій, поділена на кількість усіх подій, щоб значення йшло від 0%, коли нічого не працює, до 100%, коли нічого не зламано. Сама ціль — це риска на цій шкалі: індикатор має лишатися з правильного боку порогу або між нижньою і верхньою межею.

Повна ціль називає все, що потрібно для перевірки. Приклад із книги: 99% викликів Get RPC, усереднених за одну хвилину, завершуються менш ніж за 100 мс, якщо міряти на всіх бекенд-серверах. Один індикатор може мати кілька цілей одночасно, наприклад 90% викликів швидше за 1 мс, 99% — за 10 мс і 99,9% — за 100 мс, бо один перцентиль рідко описує весь розподіл. SRE Workbook також розрізняє специфікацію, тобто результат, важливий для користувачів, і реалізацію, тобто місце, де його вимірюють: журнал сервера, зонд, що запускає браузер, або код на самій сторінці. Кожне з цих місць бачить свої збої, і вибір місця належить до визначення.

SLO — це не SLA. Угода про рівень обслуговування (service level agreement) — це явний чи неявний договір із користувачами, який прив’язує наслідки до виконання або невиконання записаних у ньому цілей, найчастіше знижку чи штраф. Книга пропонує простий тест: запитати, що станеться, якщо ціль не досягнуто. Якщо явних наслідків немає, це SLO. У виносці додано, що люди, які говорять про порушену SLA, майже завжди мають на увазі недосягнуту SLO, бо справжнє порушення SLA може закінчитися судом за порушення договору. Google Search, зазначає книга, не має SLA для широкої публіки, але його недоступність однаково б’є по репутації та доходах від реклами.

Застосуйте поняття до рішення

Застосуйте «Що таке ціль рівня обслуговування?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Запустила перший дашборд спостережуваності — перший огляд власних систем компанії

Запустила перший дашборд спостережуваності Зв'язатися з нами →

Чому це важливо

Без заявленої цілі користувачі вигадують власну. Книга SRE описує обидва способи, як це йде не так: одні люди покладаються на сервіс більше, ніж слід, а потенційні користувачі вважають його менш надійним, ніж він є, і не приходять. Приклад — Chubby, сервіс блокувань Google для розподілених систем. Глобальні збої в ньому траплялися так рідко, що інші сервіси непомітно почали розраховувати, що він ніколи не відмовить, і ламалися в ті рідкісні рази, коли це ставалося. Розв’язок полягав у тому, щоб глобальний екземпляр виконував свою SLO, але не перевиконував її надто сильно: у кожному кварталі, коли справжні збої не опустили доступність нижче цілі, команда SRE навмисно вимикала систему.

Ціль також закриває суперечку про те, наскільки надійно — це достатньо надійно. SRE Workbook називає 100% надійності хибною ціллю і пояснює це практично. Компоненти відмовляють одночасно, хоч як їх збудовано. Ланцюжок мереж і пристроїв між сервісом і клієнтами теж відмовляє, тож клієнти однаково ніколи не побачили б цих 100%. Кожна додаткова дев’ятка коштує дорожче за попередню, а її цінність для користувачів наближається до нуля. І оскільки зміни — головна причина збоїв, сервіс, який не має права на жодну відмову, не можна й оновлювати. Щойно ціль опускається нижче 100%, проміжок між нею і досконалістю стає запасом, який можна витратити

на релізи, експерименти й планове обслуговування. Цей запас називають бюджетом помилок (error budget), і він дорівнює просто 100% мінус SLO. Хтось має відповідати за компроміс між новими функціями та стабільністю; SRE Workbook називає CTO в невеликій організації і власника продукту (product owner) чи продакт-менеджера — у більшій.

Нарешті, ціль перетворює моніторинг на рішення. Книга описує цикл: виміряти індикатори, порівняти їх із цілями, вирішити, чи треба щось робити, і зробити це. Якщо затримка зростає і за кілька годин перетне межу, команда перевіряє, чи сервери впираються в процесор, і додає ще кілька. Без цілі, наголошує книга, ніхто не скаже, чи треба діяти і коли.

Як це виглядає на практиці

Візьмімо API оформлення замовлення і ціль, записану так, як радить SRE Workbook: індикатор окремо від місця вимірювання, вікно назване, а обґрунтування лежить поруч:

```
Сервіс:          API оформлення замовлення (checkout)
SLI:             частка запитів POST /checkout без статусу 5xx,
                 пораховано в журналі запитів сервера
Ціль:            99,9% за ковзне вікно у 28 днів
Обґрунтування:   спостереження, округлено вниз із чотирьох тижнів даних
Власник:         продакт-менеджер разом із черговими інженерами
Перегляд:        щомісяця, доки ціль не усталиться, далі щокварталу
```

Вікно важить не менше, ніж відсоток. SRE Workbook віддає перевагу ковзному вікну перед календарним місяцем, бо користувачі не забувають про збій в останній день місяця лише тому, що почався новий, а також цілій кількості тижнів, щоб кожне вікно мало однакову кількість вихідних; чотири тижні — його вибір для загального призначення. Таке вікно — це 28 × 24 × 60 = 40 320 хвилин, і 0,1% від нього — трохи більше ніж 40 хвилин повного збою. Для місяця з 30 днів ті самі 99,9% дають 43,2 хвилини, і саме це число стоїть у таблиці доступності з книги.

Проте час — грубий спосіб рахувати. Книга SRE пояснює, що Google вимірює доступність як частку успішних запитів, а не як час безвідмовної роботи (uptime), бо сервіс, розкиданий по всьому світу, рідко буває повністю працездатним або повністю недоступним. У запитах приклад із SRE Workbook виглядає так: за цілі 99,9% три мільйони запитів за чотири тижні

лишають місце для 3 000 помилок, а збій, що спричиняє 1 500 із них, з'їдає половину бюджету. Підрахунок запитів враховує й часткові збої, яких не бачить uptime. За рівномірного трафіку 20 хвилин, протягом яких падає половина всіх запитів, коштують стільки ж, скільки 10 хвилин повного збою, а це приблизно чверть бюджету на 28 днів.

У Prometheus, якщо лічильник запитів має мітку зі статусом, індикатор за вікно — це одне відношення, побудоване за прикладом із SRE Workbook:

```
sum(rate(http_requests_total{job="checkout", status!~"5.."}[28d]))
  /
sum(rate(http_requests_total{job="checkout"}[28d]))
```

На цьому сайті сторінка про рівень обслуговування описує договірний бік питання. Для системи, яку компанія експлуатує для клієнта, час реагування на інцидент, години обслуговування, вимоги до доступності й вікна технічного обслуговування записують в угоду саме для цієї системи. Визначають їх для кожної системи окремо, бо нічний звіт і платіжний сервіс потребують різних відповідей. Власну систему моніторингу помилок і трасування засновниця створила в межах своєї поточної роботи в NextMariner. Дані цієї системи показано в операторських поданнях, зокрема в одному для цілей рівня обслуговування. А нотатка про те, як продати останній квиток рівно один раз, перетворює інваріант залишку на SLO: перевіряти його за розкладом, будувати графік і піднімати алерт на будь-яке розходження, щоб нуль продажів понад залишок був ціллю з алертом, а не надією.

Де виникають проблеми

Ціль скопійовано з дашборда за минулий місяць. Взяти те, чого система досягає сьогодні, здається безпечним, але книга SRE застерігає: ціль, обрана без роздумів, може приректи команду на героїчні зусилля, щоб і далі її виконувати, на системі, яку не покращити без перепроєктування. Ідіть від того, що потрібно користувачам, і записуйте обґрунтування поруч із числом: SRE Workbook просить пояснити логіку і вказати, чи взято цифри з експерименту чи зі спостереження, навіть якщо вони відверто ad hoc. Переглядайте документ щомісяця на початку і щокварталу, коли він усталиться.

Ціль — 100% або низка дев'яток, ціну яких ніхто не рахував. Система, яка завжди доступна чи масштабується без меж, дорога у створенні та експлуатації і краща, ніж помітять користувачі. Почніть із м'якої цілі й підтягуйте її, коли краще пізнаєте систему; книга віддає цьому перевагу перед жорсткою ціллю, яку згодом доводиться послаблювати, бо вона виявилася недосяжною.

Середнє заміщує те, що відчувають користувачі. У прикладі з книги типовий запит обслуговується приблизно за 50 мс, тоді як 5% запитів у двадцять разів повільніші, і моніторинг лише середньої затримки не показав би жодної зміни протягом дня. Записуйте цілі для затримки як перцентилі або як частку запитів, швидших за поріг, наприклад 300 мс, і ніколи як середнє.

Цілей більше, ніж хтось здатен обстояти. Десяток SLO на сервіс розмиває їх усі. Тест із книги прямолінійний: якщо посилання на ціль ніколи не допоможе виграти суперечку про пріоритети, ця ціль, імовірно, не варта того, щоб її мати. Залиште кілька, які покривають те, що користувачі справді роблять.

Недосягнення цілі нічого не змінює. SRE Workbook називає це SLO без зубів: дотримання цілі стає ще одним KPI у звіті замість підстави щось вирішувати. Домовтеся заздалегідь, що станеться, коли бюджет вичерпано, і нехай продакт-менеджер, розробники та SRE це підпишуть. Приклад політики бюджету помилок у SRE Workbook зупиняє всі зміни, крім виправлень помилок найвищого пріоритету та виправлень безпеки, доки сервіс знову не вкладеться в ціль, і прямо каже, що ця політика не є покаранням.

SLO збігається з SLA. Коли внутрішня ціль дорівнює договірній, першою звісткою про проблему для команди стає порушення. Книга радить внутрішню ціль, суворішу за обіцяну назовні, як запас на хронічні негаразди. Протилежна помилка теж шкодить: сервіс, що працює значно краще за свою ціль, привчає користувачів на це розраховувати, тож книга пропонує час від часу планові простої, обмеження пропускної здатності (throttling) або систему, яка не стає швидшою за малого навантаження.

Вікно довше, ніж дані. SLO на 28 днів, обчислена на сервері Prometheus, який зберігає 15 днів даних (типове значення, коли термін зберігання не задано), тихо вимірює 15 днів і видає їх за чотири тижні. Задайте термін

зберігання, довший за вікно, перш ніж довіряти цифрі:

```
prometheus --storage.tsdb.retention.time=35d
```

Що почитати далі

- Service Level Objectives — розділ 4 книги Google про Site Reliability Engineering: індикатори, цілі й угоди та як обрати ціль.
- Implementing SLOs — розділ SRE Workbook про SLI як відношення, бюджети помилок, вікна і про те, як записати ціль.
- Example Error Budget Policy — повна політика для вигаданого ігрового сервісу: коли зупиняються релізи і хто розв’язує незгоду.
- Embracing Risk — чому Google вимірює доступність успішними запитами, а не часом безвідмовної роботи, і скільки коштує кожен крок надійності.
- Availability Table — скільки простою дозволяє кожен рівень доступності на рік, квартал, місяць, тиждень, день і годину.
- Storage — як Prometheus зберігає дані на диску і які прапорці терміну зберігання визначають, наскільки далеко назад бачить запит для SLO.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке ціль рівня обслуговування?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/service-level-objective/