Strategic Whitepaper



🔆 Heureka Tech Health Framework

Strategic Whitepaper — Why, What, and How

Purpose

Create a shared, lightweight operating system for technical leadership a single framework that connects engineering health to strategic decision-making. It's a mechanism for visibility, prioritization, and deliberate trade-offs between speed, stability, and strategic investment.

(X) Context

Heureka operates multiple legacy platforms and stacks.

Each measures velocity and quality differently.

Without a unified view, leadership can't quantify risk or plan technical investment confidently.

Industry experience (Google SRE, Spotify, Meta, Amazon) proves one thing:

Metrics alone don't fix culture — transparency and rhythm do.

💡 Why DORA Alone Isn't Enough

DORA metrics — Deployment Frequency (DF), Lead Time for Changes (LTC), Change Failure Rate (CFR), and Mean Time to Recovery (MTTR) are foundational but **incomplete** in heterogeneous environments.

To make them actionable, they must be combined with:

- SLOs & Error Budgets (Google SRE principle)
- Qualitative context from teams
- Architectural and process maturity signals
- 👉 The **Tech Health Framework** unites these signals into one model.

Five Areas of Tech Health

A. TECH DEBT (5 axes)

Axis	Description
Code Quality Debt	Complex, unstable, or duplicated code. (SonarQube)
Architecture / Domain Debt	Poorly defined boundaries, tight coupling.
Infrastructure Debt	Manual deploys, env drift, missing rollback automation.
Process / Delivery Debt	Ad-hoc fixes, missing ADRs, inconsistent flow.
Knowledge Debt	Tribal knowledge, missing docs, outdated context.

Example: Amazon and Allegro use "debt heatmaps" — visualizing debt categories across teams to drive quarterly priorities.

B. TESTING & AUTOMATION

Testing isn't about counting tests — it's about confidence in every change.

The goal is to release fast without fear and catch failures before users do.

Key dimensions:

- Meaningful test strategy focus testing effort where failure hurts most (critical business flows, integrations, edge conditions).
- Unit & contract testing protect core logic and ensure service boundaries stay reliable.
- **E2E automation** cover only essential customer journeys, keeping tests fast and stable.
- **Pipeline reliability** flaky tests < 2%, build failures are actionable, not random.
- Deployment automation "push-button" deploys with tested rollback and canary mechanisms.

⊚ Goal: Increase delivery confidence, not just coverage.

Teams should measure the *confidence-to-coverage ratio* — how much they trust their pipeline, not how many tests exist.

C. OBSERVABILITY & STABILITY

- · Monitoring coverage & alert hygiene.
- MTTR and incident trends (severity level).
- · Tracing coverage and actionable logs.
- Root Cause Analyses (RCA) culture and learning loops.
 - Atlassian and Google SRE use similar "Incident Readiness Reviews" before critical launches.

D. DELIVERY PERFORMANCE (DORA)

- Deployment Frequency
- Lead Time for Changes
- Change Failure Rate
- Mean Time to Recovery
- Rollback Frequency

Start manual → automate via **Haystack**.

Visualize in **Grafana** or **Backstage Tech Insights**.

E. GOVERNANCE & KNOWLEDGE

- Governance is about making decisions visible and reusable.
- ADRs capture both team-level and organization-wide decisions team ADRs can link to Jira Epics, while global ones live in a shared ADR index.
- Architecture docs (C4/Arc42) are updated regularly, and ownership of each service is clear and public.
 - Thoughtworks and Meta treat **decision traceability** as a core pillar of long-term agility.

Roadmap to Adoption

Phase	Timeline	Focus	Output
O. Pilot	0-2 months	Content Gang teams run a	Covers Phases 1 + 2 to validate

		small-scale pilot combining self- assessments, DORA metrics, and a Tech Health Review.	metrics, cadence, and value before wider rollout.
1. Start	1–3 months	Manual self- assessment & DORA pilot	100% teams onboard
2. Align	2–6 months	Tech Health Review + ADRs	Leadership rhythm
3. Automate	6–12 months	Haystack, SonarQube, Grafana	Scorecards & trends
4. Integrate	12-18 months	Backstage Tech Insights	Centralized dashboards

Cultural Guardrails

- Metrics are mirrors, not weapons.
- Measure trends, not teams.
- Context > Comparison.
- Start manual, scale with value.
- Every decision leaves a trail (ADR).

III Level of Adoption (Maturity Grid)

Level	Description
0. None	No data, no review.
1. Manual	Sheets & surveys, baseline DORA.
2. Rhythmic	Quarterly reviews, ADRs active.

3. Automated	Four Keys, Sonar, partial dashboards.
4. Integrated	Backstage scorecards & automated capacity steering.

⋠ In One Sentence

A pragmatic, engineering-led system for turning technical health into a business conversation — without adding bureaucracy.