Evaluation, Impact & Speed–Sustainability Compass

Turning technical health data into clear leadership signals.

🎯 Purpose

To transform engineering health data into **actionable insight** — guiding where to invest, what to stabilize, and when to accelerate.

This model defines how Heureka:

- Collects and automates Tech Health data.
- Reviews and decides based on clear trade-offs.
- Connects engineering metrics to business planning through **25 % allocation** and a **Dual Backlog**.

⚙️ 1. Data Collection & Automation

Start **simple and manual**, evolve toward automation:

Source	Metric	Tool / Integration
Google Sheets / Forms	Self-assessment, Pulse survey	Team input (quarterly)
Git + CI/CD	Deployment Frequency, Lead Time, CFR, MTTR	Haystack
SonarQube	Code quality, coverage, complexity	CI reports
Grafana / Prometheus	MTTR, SLO compliance	Error Budget dashboards
Backstage	ADRs, ownership, documentation	Tech Insights plugin

🎯 *Goal:* Automate the **highest-signal metrics first** — CFR (Change Failure Rate), MTTR (Mean Time to Recovery), and SLOs.

💡 *Example:* MTTR measured manually in Q1 via spreadsheet → automated from Grafana logs by Q3.

📊 2. Visualization & Scorecards

Each team maintains a **Tech Health Scorecard**, updated quarterly:

Header:

- Team name
- Current Compass position (Speed–Sustainability)
- Average maturity level (1–4 scale)
- Trend vs last quarter

Body:

- Radar chart (five Tech Health areas)
- DORA metrics: DF, LTC, CFR, MTTR
- 25 % allocation compliance (planned vs actual)
- Top 3 quick wins + 1 major risk
- Key ADR links

🧩 Example: Team Scorecard – *"Search Experience"*

Section	Metric / Item	Current (Q2)	Previous (Q1)	Δ Trend	Comment / Next Step
🧭 **Compass Position**	Speed: 72 / Sustainability: 55	⚡ → Slightly speed-heavy	68 / 62	⬆ +4 Speed, ⬇ -7 Sustainability	CI pipeline flaky; prioritize test stability.
📊 **Average Maturity (Tech Health)**	2.9 / 4	2.7	⬆ +0.2	Steady improvement.	

⚡ **DORA Metrics**	DF = 5 deploys/week LTC = 1.5 days CFR = 9% MTTR = 55 min	DF ↑, CFR ↓	✅ On track	Focus on lowering MTTR < 45 min.	
🧱 **Tech Debt**	Architecture: 3.0 Infra: 2.5 Process: 3.0 Knowledge: 2.0	+0.2 avg	➕ Slight progress	Add refactoring ADR next quarter.	
🧪 **Testing & Automation**	CI reliability 94% Coverage 75% Flaky < 5%	Stable	✅	Add nightly smoke test for new feed service.	
🔍 **Observability & Stability**	95% alert accuracy MTTR < 1h On-call rotation active	+	✅	On-track, good OpsGenie hygiene.	
⚙️ **25% Allocation**	Planned: 25% Actual: 27%	Within target	⚖️ Balanced	Slightly above target, justified by CFR trend.	

📘 **Key ADRs (Links)**	ADR-012 “CI Self-Healing Pipeline” ADR-017 “Search Refactoring”	–	–	Approved in Q2	
🔥 **Top Risks / Quick Wins**	**Risks:** flaky builds, unclear data ownership. **Wins:** CI fix, SLO < 1% error budget.	–	–	Follow-up at next Tech Health Review.	

📘 *Example:* Displayed via Confluence table + embedded Grafana panel.

Org-level dashboards aggregate team data:

Metric	Target	Aggregation	Visualization
Deployment Frequency	↑ 20 % QoQ	Median	Bar chart
MTTR	↓ 25 % YoY	Median	Line chart
CFR	< 10 %	Weighted average	Heatmap
Tech Debt	≥ 3.0 avg	Axis average	Radar chart
Pulse Health	≥ 7.0	Mean	Trend line

🧠 3. Review & Governance Rhythm

Tech Health Review (Quarterly)

Participants: CTO, CPO, EMs, Staff Engineers.

Agenda:
1️⃣ Review team trends and outliers.
2️⃣ Identify **3–5 Tech Goals** (e.g. “CI reliability”, “Domain refactoring”).
3️⃣ Verify 25 % allocation compliance.
4️⃣ Record trade-offs as ADRs.

📘 **ADR Example:**
ADR-001: 25 % Stability Allocation Policy

- **Context:** SEV-1 incidents rising in Matching Service.
- **Decision:** Reserve 25 % of team capacity in Q2 for platform stabilization.
- **Consequence:** Slightly slower BR delivery; MTTR reduced to < 1 h.

🧩 4. Dual Backlog Integration

Two visible and equal workstreams:

Attribute	Product Backlog	Engineering Backlog
Ownership	Product Owner	Tech Lead / EM
Governance	CPO	CTO
Focus	Business features	Stability, debt, strategy
Metrics	Business KPIs	Tech Health, DORA
Capacity	~ 75 %	~ 25 %
Changes	Only via ADR	Only via ADR

💡 *Dynamic Rule:*
If SLO burn > 70 %, stability allocation auto-increases (e.g. 25 % → 40 %).

📘 *Example:*
CFR > 20 % or MTTR > 2 h → Stability = 40 %.
CFR < 5 %, DF ↑ → Shift 10 % back to BRs.

🧭 5. Speed–Sustainability Compass

A visual leadership tool showing whether the organization leans toward **speed** or **sustainability** — and by how much.

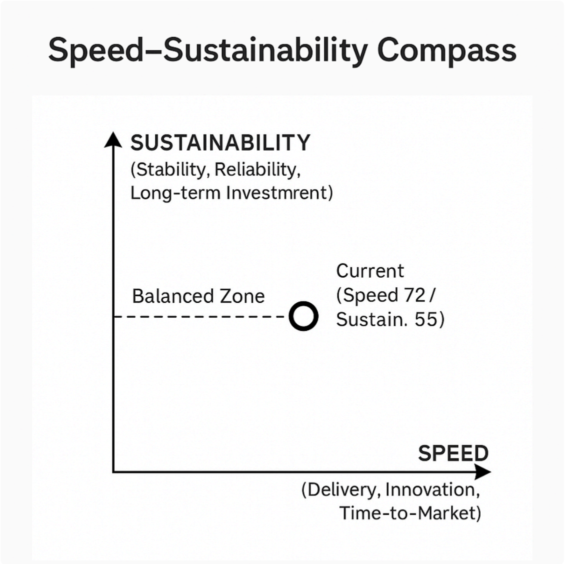
Axis	Meaning	Metrics	Key Question
⚡ **Speed**	Ability to deliver value quickly.	DF, LTC	"How fast can we safely deliver?"
🛡️ **Sustainability**	Ability to recover, stay reliable, and invest in long-term health.	CFR, MTTR, SLOs, Tech Debt trend, % Engineering allocation	"How predictable and resilient are we?"

Interpretation Guide:

Position	Meaning	Action
⚡ Right (Speed-heavy)	Fast but fragile (rising CFR / MTTR).	Increase stability allocation to 40 %; focus on alerting & CI reliability.
🛡️ Left (Sustainability-heavy)	Over-invested in maintenance; innovation slowdown.	Shift 10 % capacity back to BRs.
🎯 Centered	Balanced execution & resilience.	Maintain; share practices across org.

💡 *Google SRE principle:*
"When reliability exceeds objectives, increase innovation.
When it falls short, reinvest in stability."

⚙️ Compass: Q2 Snapshot – *Search Experience Team*



📊 Compass Trend Over Time

Quarter	Speed	Sustainability	Interpretation	Action
Q1	68	62	Balanced	Maintain 25% allocation
Q2	72	55	Speed-heavy	Reinforce stability focus
Q3 (target)	70	65	Optimal zone	Achieve steady delivery rhythm

---

📈 6. ROI & Business Impact

Every stability initiative should demonstrate business outcomes:

Technical Metric	Target	Business Effect

CFR ↓ to < 10 %	Fewer failed releases	Saved developer time, higher trust
MTTR ↓ 25 %	Faster recovery	Lower downtime cost
DF ↑ 20 %	Faster releases	Shorter time-to-market
Architecture Debt ↓ 1 lvl	Clearer ownership	Fewer cross-team blockers
Pulse Survey ↑ 1 pt	Higher satisfaction	Improved retention & morale

---

🧱 Service Criticality — Context Behind the Numbers

> Not every red metric means danger. What matters is **how critical the service is** to customers and revenue.

🔺 Tier	Meaning	Focus	Example
🔴 **Tier 1 – Critical**	Directly affects users or revenue.	Highest stability & fastest recovery (SLO 99.9 %).	Checkout, Search, Matching API
🟠 **Tier 2 – Important**	Impacts UX but not core flow.	Balance between delivery speed and reliability (SLO 99.5 %).	Recommendations, Notifications
🟢 **Tier 3 – Supporting**	Internal or non-critical tools.	Flexibility, innovation speed (SLO 99 %).	Admin tools, Reporting, Sandbox envs

💡 Interpretation Tip — Understand System Criticality

When reviewing Tech Health or Compass results, **not all red numbers mean the same thing**. The key is to weigh metrics by how critical each service is to the ecosystem.

Ask:

- 🧩 “If this service fails for 10 minutes — who feels it?”
- ⚙️ “If it degrades for hours — what other services or teams slow down?”
- 🔗 “Does it trigger cascading failures or block dependent domains?”

Why it matters:
Some issues are loud but isolated — others silently amplify across the platform.
Stability work should focus not just on uptime, but **on preventing systemic ripple effects**.

Principle — Proportional Reliability

> Aim for reliability *in proportion* to business and technical criticality.
> Stabilize what drives revenue and platform integrity.
> Experiment where failure is recoverable and low-cost.

Example Matrix (for context):

Criticality	Typical Example	Impact Type	Reliability Target
🟥 **Tier 1 – Business Critical**	Checkout, Search, Price Feed	Customer-facing, cascading	SLO ≥ 99.9 %, MTTR < 1h
🟧 **Tier 2 – Platform Core**	Identity, API Gateway	Affects multiple domains	SLO ≥ 99.5 %, MTTR < 2h
🟨 **Tier 3 – Internal or Supportive**	Back-office tools, reporting	Limited scope	SLO ≥ 99 %, MTTR < 6h
🟩 **Tier 4 – Experimental / Low Impact**	Sandboxes, POCs	Safe to fail	No strict SLOs, MTTR best effort

---

🔁 **Evaluation Note:**
When interpreting Compass or DORA results, **adjust expectations by criticality** —

a Tier-1 service with MTTR 2h is an emergency,
while a Tier-4 service with MTTR 2h is acceptable learning cost.

💬 **In One Sentence**

A lightweight, data-driven operating model that makes **technical health visible, quantifiable, and actionable** — aligning engineering and business around facts, not opinions.