International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I3P102
Volume 73 Issue 3, 16-24, March 2025

Original Article

Architecting Change: A Leadership Guide to Technology Transformations

Maalika Tadinada

Director of Engineering, Electronic Arts, Massachusetts, USA.

Received: 09 January 2025 Revised: 19 February 2025 Accepted: 03 March 2025 Published: 15 March 2025

Abstract *- Technology transformations are crucial for modern enterprises to remain competitive, scalable, and innovative. Organizations prioritizing technology and operational excellence achieve higher efficiency, reduced costs, and sustained growth. This study presents an approach to executing technology transformations and addressing common organizational challenges such as operational inefficiencies, scalability limitations, and suboptimal team composition. By analyzing leadership principles and engineering and operational best practices, the research outlines a step-by-step methodology for aligning technology investments with business outcomes. Key insights include strategies for restructuring teams, optimizing development processes, and fostering a culture of trust.*

Keywords *- Digital transformation, Technical leadership, Operational excellence, Team building.*

1. Introduction

In today's rapidly evolving business landscape, technology transformations are critical for organizations aiming to maintain competitiveness and drive growth. Embracing digital transformation enables companies to enhance operational efficiency, improve customer experiences, and foster innovation. By investing in advancing technologies, organizations can streamline processes, reduce costs, and respond more agilely to market changes.

Research indicates that organizations prioritizing technological excellence significantly outperform their peers. Operational excellence, driven by technology adoption, can transform a challenged organization into a competitive entity or elevate a strong performer to a benchmark setter.

For instance, a mining company increased its output by 25% in the first year of implementing advanced technologies, with continued improvements in subsequent years, all while keeping capital and labor costs steady. [1] In a 2022 McKinsey survey, when CEOs were asked whether they think technology can improve business growth and productivity sufficiently to lift profits and shareholder value by 30 to 50 percent, a great many said yes.[2]

Technology transformations, however, are laced with fears and misconceptions. They either take longer than scheduled, fail to provide tangible value, or are deferred for fear of the investment being too significant to address.[3]

For leaders of technology teams responsible for mission-critical applications and systems, implementing technology transformations isn't just a matter of modernizing the technology stack but, more importantly, about implementing the right organizational structure and establishing a culture of excellence rooted in trust. This paper will guide leaders on a structured, sustainable approach to technology transformation that works in tandem with business outcomes.

2. Why Prioritize Tech Transformations

Technology transformations are essential for organizations to avoid technology bottlenecks that affect the company's growth and profitability. Without a culture of technology and operational excellence, technologies typically run into challenges like scaling limitations to support the company's growth trajectory, high operational costs and, inefficiencies that hinder profitability and innovation deficit where the team is largely reactive and lacking a proactive focus on innovation and value creation.

Technology leaders that focus their organizations on the structural elements necessary for sustainable technology growth avoid the organization from [4]

- becoming a 'feature factory' that focuses only on the short-term needs of the customer
- having high operational overhead to 'keep the lights on.'
- disjointed re-architecture attempts have led to duplicated work across teams, further depleting resources available for innovation.

- inconsistent structure and multiple management layers, with middle managers often acting as "status providers" rather than critical decision-makers.
- having blurred lines of ownership across different functions, creating a lack of functional depth leading to situations like:
  - Product Management: Shifted to tactical feature management, responding to short-term operational demands rather than developing long-term strategy and well-defined requirements.
  - Product Analytics: Primarily used for tactical analysis, lacking a structured experimentation and outcome analysis approach.
  - Data Science & Applied Research: Inconsistent paths to productization for research outcomes.
  - Engineering: Disproportionate time spent on re-platforming, on-call support, and justifying work, hindering the development of new capabilities.
  - User Experience: Focused on tactical UI design and wireframing rather than strategic engagement with end-to-end customer journeys and cohesive user experiences.

3. Executing Technology Transformations

This paper will cover five essential steps any technology leader must undertake to execute a technology transformation in their organization. Starting with taking stock to establish a clear vision and implementing critical organizational changes. With a focus on engineering and operational excellence, leaders can unlock innovation and enable a strategic mindset, no matter the size of the organization or the technology domain.

4. Step1: Take Stock

The first step for a leader is to take stock of what they are about to work with and work on. Four key areas are to assess: People, Processes, Products & Services, and Financials.

4.1. People

Leaders serve their people and are responsible for enabling them to serve the company's customers and shareholders. Their team is their first and foremost customer, partner, and stakeholder. They must invest time meeting with team members, cross-functional peers (sales, marketing, operations, finance, HR, legal), and key individuals reporting to their peers.

4.1.1. Prep

Create a near-comprehensive list of people to meet. Enlist a small, trusted group of senior leaders to help build the initial list and then progressively expand it. The goal is to consult with individuals across the following:

- Different levels of seniority
- Management stages (individual contributor, frontline manager, group manager, executive)
- Identified high performers and, if possible, those losing steam or recent flight risks.

4.1.2. Action: Meet

These meetings serve two purposes. First, they help a leader gather diverse perspectives on culture, processes, and organizational dynamics, which ultimately help inform the leader of what's working and what's not. Second, they demonstrate the intent of the leader and their methodical approach, even though it is too early to share informed opinions on specific concerns.

Leaders must develop a pre-planned structure for these conversations. The agenda must be clear to each participant and followed up with key takeaways and expectations for the next steps.

4.1.3. Synthesize

After meeting with most team members, leaders should synthesize notes to identify macro and micro patterns. Takeaways typically fall into themes like:

- Organizational Structure: Role clarity, expectations, aligned goals.
- Talent: Hiring bar and process, growth opportunities (structured learning, lateral transfers, career models and promotions, rewards and recognition).
- Culture: Clarity of purpose, defined guiding principles/values, processes for strategy/goal setting, planning, execution, and outcome measurement

4.2. Process

Every company has rituals and processes for important activities (e.g., budget setting and tracking, talent hiring, performance calibration and promotion, company/org-wide planning, operational progress, design and architectural review). These practices may be consistently defined across the company or localized to specific teams, with varying levels of consistency. Processes and rituals are essential for achieving mastery. They enable teams to create repeatable habits that ensure capability excellence.

4.2.1. Prep

- Talent: Acquisition, performance evaluation, promotions, rewards/recognition.
- Investment: Strategy development, planning, resourcing, prioritization.
- Delivery: Progress tracking, design review, execution check-ins.

4.2.2. Action: Assess

- Inventory existing processes/rituals.
- Attend meetings, observe, and learn how things are progressing.
- Document inferences on what's working and what's not.

Table 1. Examples of good & bad processes

Good Process	Bad Process
Clear purpose on what capability or decision it aims to affect. A clear definition of what measures will be used to track and discuss capability progress is needed. Transparent framework on how decisions will be made based on the tracker. Leaders helped define its active own governance, progressively iterating as needed.	Clear purpose on what capability or decision it aims to affect. A clear definition of what measures will be used to track and discuss capability progress is needed. Transparent framework on how decisions will be made based on the tracker. Leaders helped define it and actively own governance, progressively iterating as needed.

4.2.3. Synthesize

- Document the effectiveness of current processes. And gaps.
- Analyze processes based on established criteria (clarity, metrics, transparency, governance)

4.3. Products and Services Rendered

Understanding the value created for customers (live, launched, and operational products and services) is another crucial area for assessment.

4.3.1. Prep

- Examine published marketing and sales materials (latest and older iterations) to understand current and promised value. Management stages (individual contributor, frontline manager, group manager, executive)
- Use current offerings as an end-consumer, going beyond a cursory review to explore services, scenarios, and use cases.

4.3.2. Action: Conduct Technical Walkthroughs

- Architecture Review – review for durability (domain and service boundaries, scalability, independent evolution).
- Development plans – review for development practices and velocity (sustainable quality practices, automation, robust testing, documentation, reviews).
- Site/System Reliability – review for service quality (availability, reliability, latency, proximity to "three nines," and the opportunity cost of not achieving that maturity).
- Service Management – Review for dev-ops app-ops practices (incident management, outage processes, recurring incident/problem management).
- Service costs – review for maintenance costs (percentage of tech spend on live-site issues and production-related work).

4.3.3. Synthesize

These technical walkthroughs will enable a leader to develop a perspective on:

- Current value commitments.
- Alignment of current versions with commitments.
- Product Robustness and potential weaknesses.
- Efficiency of ways of working.

4.4. Financials

The final step is understanding the company's financials and how technology impacts it. This is often an area where technology leaders are least comfortable. To develop 360° understanding and build close partnerships with finance, business teams (operations & commercial) and product leaders.

4.4.1. Walkthrough

- Financials: current revenue, current net operating cost (profitability), and future growth commitments.
- Business teams: financials of different services the company offers (and how technology powers or impedes their success).
- Product leads: product portfolio inventory and how each enables revenue, profit, and scale.

4.4.2. Synthesize

- Develop an independent view of the criticality of their technology portfolio.
  - Performance of different products and services (revenue, profit).
  - Competitive moats (high repeat engagement) vs. adjacent services.
- Reconnect with cross-functional leaders on their view and iterate to build an aligned summary view of company financials (present and future) and technology investments.
- Educate and inform the team on company strategy and contributing macro-factors.

5. Step2: Establish Clear Vision, Mission & Strategy

A successful technology transformation begins with a clear, well-articulated vision and strategy that aligns the engineering organization with the broader business goals. This foundational step ensures focus, alignment, and measurable outcomes. Below are three key topics that leaders must address to define and communicate this vision effectively.

5.1. Define the Purpose of the Technology Organization

The leader must articulate the core purpose of the technology organization within the broader company context. This statement should inspire the team, clarify priorities, and establish a guiding decision-making framework.

- *Support the Business Mission*: Highlight how the technology organization serves as the engine to enable the company's overarching goals, whether delivering exceptional products, enhancing customer experience, or driving innovation. Example: "Our technology organization exists to deliver seamless, scalable solutions that empower the company to meet customer needs faster, more reliably, and more innovatively than ever before."
- *Commit to Innovation and Operational Excellence*: Position the team as a driver of cutting-edge solutions and a steward of operational reliability, ensuring forward-looking innovation and day-to-day stability.
- *Empower Engineers as Strategic Partners*: Shift the narrative of the tech organization from being a "support function" to a strategic enabler. Example: "We are not just building software; we are crafting the future of our business through technology."

5.2. Aligning Technology with Business Outcomes

The vision must directly connect the engineering transformation to tangible business goals to justify the investment and gain buy-in across the organization.

- *Driving Revenue Growth*: Outline how technology will create new opportunities, whether through better products, enhanced customer acquisition tools, or improved go-to-market strategies. Example: A revamped architecture enabling faster feature delivery for SaaS products can improve customer retention and drive upsell opportunities.
- *Boosting Profitability*: Emphasize cost optimization through automation, infrastructure efficiency, and operational excellence. Example: Transitioning to cloud-native systems may reduce infrastructure costs by 30% while improving system reliability.
- *Scaling for Tomorrow*: Define how the organization will be prepared for future growth in terms of headcount and technical capacity. Example: Adopt scalable microservices to support rapid feature development as the business grows into new markets or scales its user base.
- *KPIs and Metrics*: Include measurable goals tied to revenue, profit, and scalability. Example: Set targets like "Reduce time-to-market for new features by 50%" or "Achieve 99.99% system uptime within 12 months."

5.3. Establishing the Strategic Thesis for Transformation

The technology transformation strategy must be built on a key thesis—a central, unifying idea that guides all decision-making. This thesis should be bold yet practical, balancing ambition with feasibility.

- *Customer-Centric Design*: "All technology decisions will prioritize customer experience, ensuring our systems and platforms deliver maximum value to the end user." Example: Building APIs and user interfaces that simplify customer workflows, ensuring frictionless adoption.
- *Agility Through Modernization*: "Our transformation will focus on modernizing legacy systems to create a flexible, scalable architecture that adapts to rapidly changing business needs." Example: Invest in modular, microservices-based architecture and leverage cloud-native solutions.
- *Data as a Competitive Advantage*: "We will become a data-first organization, leveraging advanced analytics and AI to gain actionable insights that drive smarter decisions across the business." Example: Build data pipelines to enable real-time decision-making in areas like pricing or supply chain management.
- *Engineering Excellence as a Core Competency*: "The strategy hinges on fostering a culture of engineering excellence, where the team operates at the intersection of speed, quality, and innovation." Example: Standardize engineering practices such as CI/CD pipelines and automated testing, ensuring rapid yet reliable software delivery

By defining the purpose of the technology organization, aligning it with business outcomes, and rooting the transformation in a key strategic thesis, leaders can inspire teams, gain executive buy-in, and set a clear path for success.

A well-structured vision ensures that the technology organization evolves not just as a cost centre but as a powerful driver for growth, efficiency, and long-term competitiveness.

6. Step3: Implement Critical Organizational Changes

Technology transformation requires ensuring the team is structured for success, encompassing structural integrity and robust, transparent processes.

6.1. Structural Integrity

Structural Integrity has three key aspects: a) strong functional depth, b) balanced teams (functionally and by seniority), and c) a relatively flat organizational hierarchy.

6.1.1. Functional Depth

Most talent is hired for functional expertise. Leaders must assess current functional coverage, identifying strengths and weaknesses.

Define Clear Functional Requirements

- Evaluate which functions are well-understood and effectively deployed and which need refinement or are missing.
- Example: Successful Data Science/Applied Research (DS/AR) investment requires complementary functions like Machine Learning Engineering and Product Managers with relevant expertise.
- Prematurely introduced functions may be misused/underused if deployment is not well-planned.
- How:
    - Request an HR inventory of functions and career ladders.
    - Review role definitions and expectations.
    - Corroborate with team members.
    - Address gaps by tasking functional leaders to review, update, and communicate expectations.
    - Evaluate recourse for prematurely introduced functions (repurposing, hiring pause).

Build Functional Depth as Needed

- Complex problem-solving requires functional depth and experience.
- Not all functions require extensive depth. Balance depth with the complexity of the problems faced.
- Example: A small mobile app company doesn't need a 20+ year architect.
- Implement a deliberate approach to hiring for depth based on problem maturity and ambiguity.

6.1.2. Balanced Teams

High-integrity teams require functional balance and a seniority mix.

Functionally Balanced Teams:

- Technology problem-solving often involves understanding consumer problems, evaluating competitors, customer journey mapping, functional and technical design, consumer targeting, and post-launch analysis.
- Each task requires specific functional expertise (e.g., product manager, designer, engineer, analyst).
- Team success depends on the right functional expertise at each stage.
- "2-pizza team" (team size such that 2 pizzas would suffice to feed them for lunch) with an "n-in-a-box" formation is a useful model:[5]
    - "2-pizza team": ~7-9 engineers (individual contributors and 1 manager).
    - "n-in-a-box": Teams of SWE/EM, PM, XD, and PA in configurations like "2 in a box," "3 in a box," or "4 in a box" depending on the problem domain.

Seniority Mix

- Balance experience and fresh perspectives. A good mix could be:
    - 25% Entry-level (<3 years experience)
    - 45% Mid-level (3-7 years experience)
    - 20% Senior-level (7-12 years experience)
    - 10% Staff/Principal-level (12+ years experience)

This supports learning and mentorship opportunities.

Table 2. Ratio of 'pizza teams' to other functions

	SWE + EM	Product	Design	Product Analyst
2 in a box	X	X		
3 in a box	X	X	X	
3 in a box	X	X		X
4 in a box	X	X	X	X

6.1.3. Flat Organizational Structure

Mature organizations prioritize "doers over facilitators" and "leader managers."

- Principles:
    - High span of control between individual contributors and managers (management/leadership <12-15% of the organization). Aim for 3-4 levels between contributors and senior executives.
    - Management as a leadership function: Each management level has specific tasks, measurable outcomes, and KPIs.
- Transparency:
    - Publish explicit targets for each leader and team to ensure clarity and collective accountability.

Illustration of Span

Table 3. Different levels in an organization vs span of control

Team Level	Ratio
Individual Contributors (ICs)	8 to 10 ICs = 1 manager
Frontline manager (FLM)	4 to 5 managers = 1 group manager (i.e. 30 to 50 ICs)
Group Manager (GRPM)	3 to 4 manager = 1 director (i.e. 90 to 200 ICs)
Director (DIR)	3 to 4 directors = VP (i.e. 270 to 500)
VP	3 to 4 directors = SVP/CTO (i.e. 800 to 2000)

Table 4. Ideal % of management layers vs ICs

% IC	% Manager	% Exec	% Leadership (Manager + Exec)
8-87%	13-14%	1 – 1.5%	13-14%

Illustration of Management Responsibilities

The structured design aligns roles, responsibilities, and outcomes, fostering efficiency and accountability. Balancing hierarchy with agility and emphasizing leadership-driven management creates a cohesive, high-performing team.[6]

Table 5. Examples of management layers vs ICs

	Management Roles		Executive Roles			
IC	FLM	GRPM	DIR	VP	CTO	CTO
220	24	6	2	0	1	254
420	45	15	5	2	1	248
650	70	22	7	2	1	752
870	94	27	8	3	8	1003

Table 6. Examples of management responsibility and frequency of tasks

	Task	Frequency	VP	Director	Group	Lead	IC
Business Level Definition	Mission (STO Purpose)	1x / 3 Years	A + R	C			
	Identify / Align Key Leaders	1x / 3 Years	A + R	R			
	Vision *STO-wide*	1x / 3 Years	A + R	C	C		
	Financial & Operational Targets *$$$, Process + System Ops*	1x / Year	A	R	C		
	Product & System Targets	1x / Year		A	R	C	
Tech Org Definition	Technology Strategy *Org Tech-wide engineering strategy*	1x / Year	A	R	R + C		
	Org definition *Org design (pods, people, outcomes)*	2x / Year	A	R	C		
	Team definition	2x / Year		A	R	C	
	Product Strategy *for each product*	1x / Year		A	R	R	C
Plan	Long-Term Plan *12+ months*	2x / Year		A	R	R	C
	Quarterly Planning *(ritual only) Need/Have Level*	1x / Quarter			A	R	R
Design	Product + UI/UX Spec	Ongoing				A	R
	Tech Design (high-level)	Ongoing				A	R
	Experiment Plan	Ongoing				A	R
	UAT & Release Plan	Ongoing				A	R
Build & Release	Design Reviews (Spec, UI/UX, Tech)	2x / Month			A	A	R
	Architecture Review *on group of systems*	1x / Quarter		A	R	A + R	R
	Execution Velocity Tracking	2x / Month			A	R	
	Release Quality Tracker	2x / Month			A	R	
	Relese Notes & Demo	2x / Month			A	R	
	Experimentation Readout	2x / Month				A	R
	Task	Frequency	VP	Director	Group	Lead	IC
Operate & Track	Product Metrics Review	4x / Month			A	R	
	Incident / Outage Tracker	2x / Month			A	R	
	Quality of Service Review	1x / Month			A	R	
	Budget Tracking	1x / Month	A + R	R			
	Big Initiatives Progress Review	1x / Month		A + R	R		
	Experimentation Tracker	1x / Month				A	R
Talent	Durable teams	1x / Year	A	R			
	Goal Definition (for Level - 1)	1x / Year	A + R	R	R	R	
	Career Dev Plan (for Level - 1)	1x / Year	C	C	C	C	R
	Performance Check-ins (for L - 1)	2x / Year	A + R	R	R	R	

6.2. Robust Processes

A key indicator of team success is the strength and transparency of its processes, especially in investment allocation and cross-team collaboration. Inconsistent practices across growing organizations can hinder interoperability and slow progress. Technology leaders must establish consistent expectations that empower teams while enabling effective collaboration on large initiatives.

Robust processes are the foundation of operational efficiency and scalability. They provide the clarity, consistency, and structure needed to manage complexity and ensure quality outcomes. Here's a blueprint for implementing robust processes:

6.2.1. Planning (Transparency & Predictability)

Effective planning enables better prioritization and alignment.

- Capacity Transparency: Track team capacity, factoring in ongoing work, technical debt, and innovation. Use tools like sprint velocity tracking or workload capacity planning.
- Scope Clarity: Define the scope, timeline, and impact for every work item. Use frameworks like RICE (Reach, Impact, Confidence, Effort) for prioritization.
- Collaborative Roadmapping: Collaborate with stakeholders to create and communicate transparent roadmaps. Regularly update roadmaps.

6.2.2. Thoughtful Design

Robust design processes identify risk early.

- Pre-Implementation Reviews: Introduce structured design reviews where architects, senior engineers, and cross-functional stakeholders validate proposed solutions. Ensure design documentation addresses scalability, maintainability, security, and potential edge cases.
- Experimentation & Prototyping: Incorporate rapid prototyping to validate feasibility and uncover risks in complex projects before full-scale implementation.
- Design Guidelines & Playbooks: Develop standardized design guidelines to ensure consistency across teams while allowing flexibility for innovation.

6.2.3. Execution (Productivity & Quality)

Balance speed and quality.

- Agile Development: adopt agile or hybrid software development methodologies while ensuring the conduct of sprint retrospectives.
- Quality-First Culture: Integrate quality assurance into the development lifecycle with Test-Driven Development (TDD), code reviews, and peer reviews.
- Automation & Tooling: Invest in CI/CD pipelines to reduce manual errors and ensure faster, reliable deployments. Automate repetitive tasks, such as regression testing or code linting, to free up engineers for high-value work.

6.2.4. Effective Release Management

Minimize disruption and ensure smooth launches.

- Release Cadence: Define a predictable release schedule (e.g., weekly, bi-weekly) that aligns with the organization's operational tempo. Allow flexibility for urgent hotfixes without disrupting the main release pipeline.
- Risk Mitigation: Implement canary releases, feature toggles, or staged rollouts to minimize risk during production deployments. Conduct thorough pre-release testing in environments that closely mirror production.
- Post-Release Monitoring: Establish robust monitoring and alerting systems to detect and address issues quickly after deployment. Use postmortem reviews for significant incidents, focusing on lessons learned and preventive actions.

6.2.5. Measuring Outcomes and Accountability

Track impact and drive improvement.

- Outcome-Focused Metrics: Track KPIs such as time-to-market, customer satisfaction (e.g., Net Promoter Score), and operational metrics like uptime or MTTR (mean time to recovery). Align these metrics with

broader business goals to demonstrate engineering's impact.
- Team Accountability: Use OKRs (Objectives and Key Results) to align team efforts with organizational priorities. Encourage a culture where teams own outcomes, not just deliverables.
- Feedback Loops: Establish regular check-ins with stakeholders to assess progress and recalibrate priorities as needed.

6.2.6. Monitoring Production Health

Ensure operational excellence and customer satisfaction.
- Service-Level Objectives (SLOs): Define clear SLOs for system performance, reliability, and response times and monitor them rigorously.
- Incident Response Protocols: Implement well-documented incident management processes, including defined roles, escalation paths, and post-incident reviews.
- Cost Optimization: Regularly review cloud and infrastructure costs to ensure operational spending aligns with performance goals. Use cost-management tools to identify inefficiencies and optimize resource utilization.

Robust processes for planning, design, execution, release management, measurement, and monitoring are crucial for technology transformation.

They enable scaling, foster innovation, ensure quality, and deliver measurable business impact. Embedding these practices ensures consistent operation at the intersection of productivity, quality, and accountability.

7. Step4: Focus on Engineering & Operational Excellence

Engineering and operational excellence are foundational to a technology transformation. Achieving excellence in both requires clarity in expectations across all levels of leadership — from individual contributors ensuring high-quality execution to frontline managers establishing best practices, group managers setting organizational standards, and executives reinforcing a culture of continuous improvement.

Excellence is not just the responsibility of a single group—it is a collective mandate that spans all levels of the engineering organization.

Below, we define engineering and operational excellence along three dimensions — quality, speed, and cost —while emphasizing that the journey to maturity must be built on a foundation of trust and continuous learning, allowing teams the space to improve iteratively without the fear of punitive consequences for misses.

7.1. Engineering Excellence

Engineering excellence ensures that every new feature, capability, and product is delivered with high quality, at a predictable speed, and an optimized cost. The goal is to build a culture of craftsmanship where engineers take pride in building reliable, scalable, and maintainable software.

7.1.1. Quality

Achieving high-quality deliverables requires a structured approach to testing, automation, and clear definitions of completeness.
- Testing Standards & Automation: Set clear expectations around the unit, integration, and end-to-end testing as non-negotiable development components. Automate as much as possible—CI/CD pipelines should run comprehensive test suites before every release to catch regressions early.
- Robust Environment Control: Ensure sandboxed environments where engineers can safely test features before they reach production. Standardize staging environments that mirror production closely, reducing the risk of unexpected failures.
- Universal Definition of "Done": Align all teams on what constitutes a "finished" feature—e.g., it must pass functional testing, meet security requirements, and have proper documentation before deployment.

7.1.2. Speed

Enabling fast and predictable engineering throughput does not mean rushed development — it means moving fast without breaking things.
- Architecture & System Design Best Practices: Mandate thorough design reviews before coding begins, ensuring sound and future-proofed technical decisions. Implement scalable architecture patterns that reduce tech debt and enable teams to confidently ship features.
- Ownership of Engineering Throughput: Hold engineering leaders accountable for tracking and improving development velocity. Use flow metrics (cycle time, deployment frequency) to identify bottlenecks and continuously refine engineering processes.

7.1.3. Cost

Driving efficiency in development work must be measured and optimized continuously to ensure the best use of resources.
- Accurate Work Estimation: Invest in better estimation practices, using historical data and time-boxing to ensure realistic scoping.
- Effective Time Allocation for Engineers: Measure how much-focused development time each engineer has in a sprint (eliminating unnecessary meetings and distractions).
- Increasing Focus Factors & Efficiency: Track efficiency iteratively—identify roadblocks and

streamline processes without overburdening teams. Encourage pair programming and peer reviews to reduce rework and improve quality while maintaining efficiency.

7.2. Operational Excellence

Operational excellence ensures that production services run with high reliability, fast incident resolution, and optimized cost. It is the engineering organization's responsibility to own the health of the systems they build.

7.2.1. Quality

Tracking and Improving Quality of Service.

- Transparent Reliability Tracking: Every leader, from frontline managers to executives, must have visibility into the organization's reliability metrics.
- Differentiated Targets for Critical vs. Non-Critical Services: Set higher reliability targets (e.g., 99.99% uptime) for mission-critical services and appropriate thresholds for less critical systems.
- Continuous Improvement Practices: Regularly review and refine SLOs (Service Level Objectives) to keep pace with business needs. Invest in observability tools to proactively detect and resolve issues before they impact customers.

7.2.2. Speed

Managing service reliability with operational rigour means resolving issues quickly and preventing recurrence while maintaining a high-quality user experience.

- Incident Response Metrics & Standards: Establish clear SLAs for Mean-Time-To-Acknowledge (MTTA) – how quickly incidents are identified and responded to and Mean-Time-To-Resolve (MTTR) – how efficiently teams resolve issues.
- Operational Runbooks & Standardization: Ensure every service has a defined incident response process that is well-documented and rehearsed. Automate response workflows where possible to speed up resolution times.
- Transparent & Actionable Postmortems: High-severity incidents should trigger non-punitive postmortems to identify long-term improvements. The focus should be on preventative measures, not blaming individuals.

7.2.3. Cost

Optimizing engineering bandwidth for operations is critical to maintaining sustainable business growth.

- Measuring Engineering Bandwidth for Operations: Track what percentage of engineering capacity is dedicated to maintaining high-quality production services.
- Setting Cost Targets Without Compromising Quality: The goal is to keep operational costs below 10% of total engineering spend without sacrificing reliability. Encourage teams to reduce toil through automation rather than just adding more headcount.
- Balancing Quality vs. Cost: Identify strategic investments in infrastructure and automation that reduce long-term operational costs. Ensure cost optimizations align with maintaining (or improving) service reliability.

7.3. Trust

While quality, speed, and cost are critical to achieving excellence in engineering and technology operations, the foundation and guiding principle should always be "maturing through trust and learning". A transformation towards engineering and operational excellence is not about enforcing rigid controls but about creating an empowered culture where teams feel ownership and accountability.

- Trust as a Foundation: Excellence is only sustainable when teams feel safe to experiment, learn, and improve. Avoid a culture of fear-based metrics—instead, focus on iterative improvements and celebrating progress.
- Continuous Learning and Growth: Engineers should feel empowered, not burdened, to improve processes and meet high standards. Failures should lead to learning and process refinement, not blame.
- Durable, Not Just Reactive: Operational and engineering excellence must be built for long-term sustainability. Proactive investments in tooling, automation, and cultural shifts will yield exponential improvements.

Achieving engineering and operational excellence requires collective responsibility across all levels of leadership. Excellence is not the job of just engineers or operations teams—it is an organizational commitment that involves setting high standards, driving efficiency, and continuously improving with trust at the core.

By embedding these principles into the organisation's fabric, engineering teams can move faster, with higher quality, and at optimal cost, creating a resilient, high-performing technology organization that enables the business to scale successfully.

8. Step5: Unlock Innovation & Enable Strategic Mindset

For an engineering organization to drive meaningful innovation, it must operate as a strategic partner rather than just an execution function. This requires empowering teams, implementing scalable processes, and investing in high-impact initiatives that align with business goals.

- *Empower Engineering Teams*: Innovation starts with agency and ownership at all levels. Engineers should proactively shape solutions and drive value beyond feature development.

- Shift from Execution to Strategy – Involve engineers in problem discovery, not just implementation. This means creating hard/direct ownership of business outcomes for engineering executives (along with product and business executives).
- Foster a Culture of Experimentation – Encourage hack weeks, incubation sprints, and risk-taking without fear of failure.
- Enable Leadership-Driven Innovation – Managers must advocate for, fund, and support engineering-led initiatives. Introduce healthy competition that incentivises engineering leaders and teams to pursue innovation that unlocks business value. Offer high-performing managers/teams more agency to allocate funds for innovations identified from the ground.
- *Invest in High-Impact Initiatives*: Not all innovation is equal — focus on initiatives that drive measurable business value.
    - Product Innovation – New revenue-generating features, AI-driven personalization, and automation.
    - Process Innovation – Streamlining development workflows and reducing engineering effort.
- Platform Innovation – Scalable architectures and reusable APIs for faster feature delivery.
- Data & AI Innovation – Using machine learning and analytics to unlock operational efficiencies.
- *Build a Culture of Sustainable Innovation*: Innovation thrives in a culture of trust, collaboration, and recognition.
    - Psychological Safety – Encourage risk-taking and learning from failures.
    - Cross-Team Collaboration – Foster open discussions between engineering, product, and business teams.
    - Incentivize Innovation – Recognize contributions through internal funding, visibility, and rewards.

9. Conclusion

Technology transformations are not merely an operational necessity but a strategic imperative. Organizations that invest in technological excellence and foster strong collaboration between business and technology leaders are better positioned to achieve operational excellence, drive innovation, and sustain long-term success.

References

[1] Today's Good to Great: Next-gen Operational Excellence, McKinsey, pp. 1-10, 2024. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/todays-good-to-great-next-generation-operational-excellence

[2] The Cornerstones of Large Scale Technology Transformation, McKinsey, 2018. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-cornerstones-of-large-scale-technology-transformation

[3] Overcoming to Core-Technology Transformation Stalemate, McKinsey, pp. 1-7, 2020. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/overcoming-the-core-technology-transformation-stalemate

[4] Ian Thomas, and David Rosewell, The Four Essential Pillars of Digital Transformation, pp. 1-6, 2016. [Online]. Available: https://www.fujitsu.com/nl/Images/The_Four_Essential_Pillars_of_Digital_Transformation.pdf

[5] Powering Innovation and Speed with Amazon's Two-Pizza Teams, Daniel Slater, Amazon, pp. 1-14, 2023. [Online]. Available: https://aws.amazon.com/executive-insights/content/amazon-two-pizza-team

[6] Two Pizza Team, Martin Fowler, 2023. [Online]. Available: https://martinfowler.com/bliki/TwoPizzaTeam.html