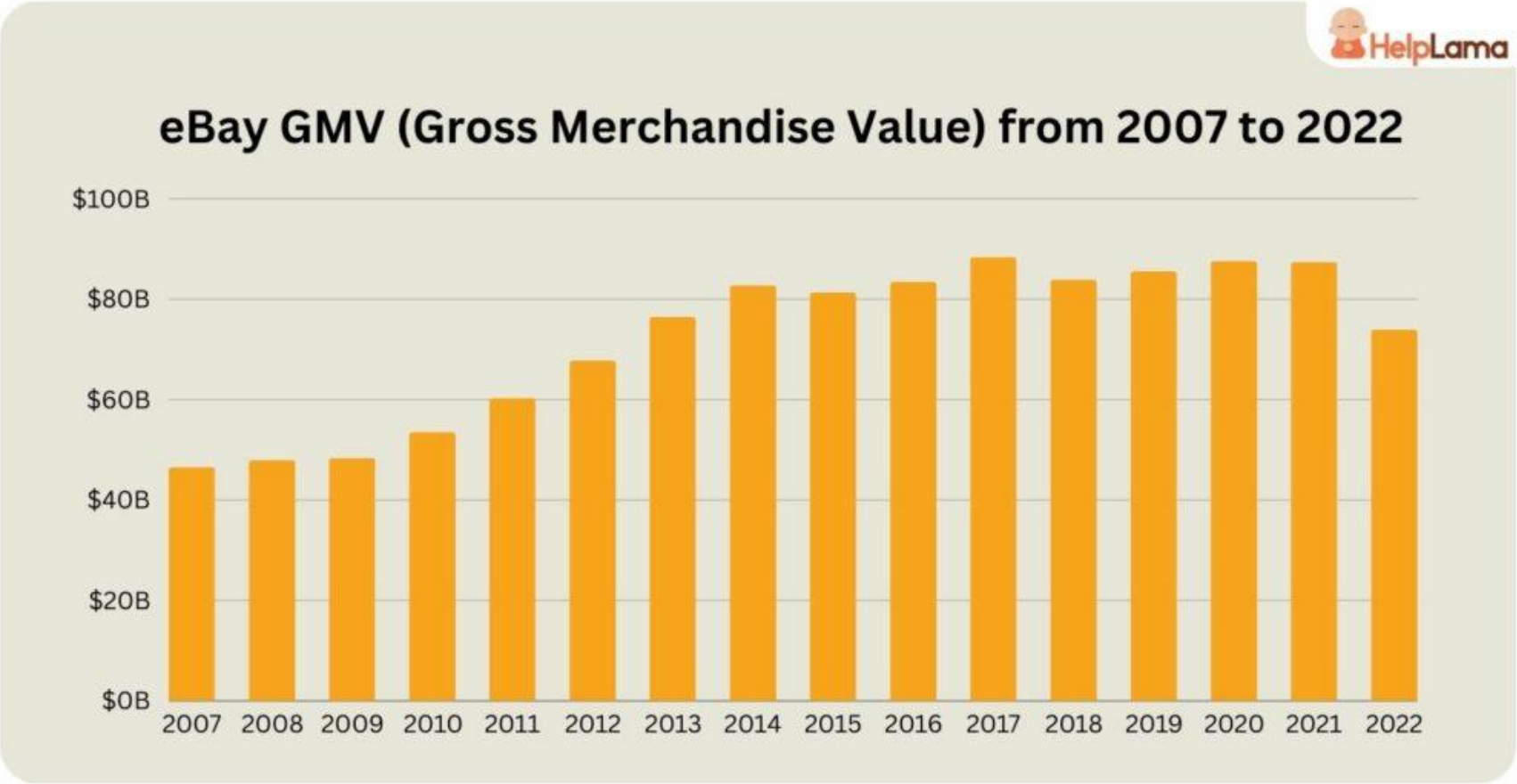
We Doubled Engineering Productivity at eBay …

but Couldn't Change the Culture

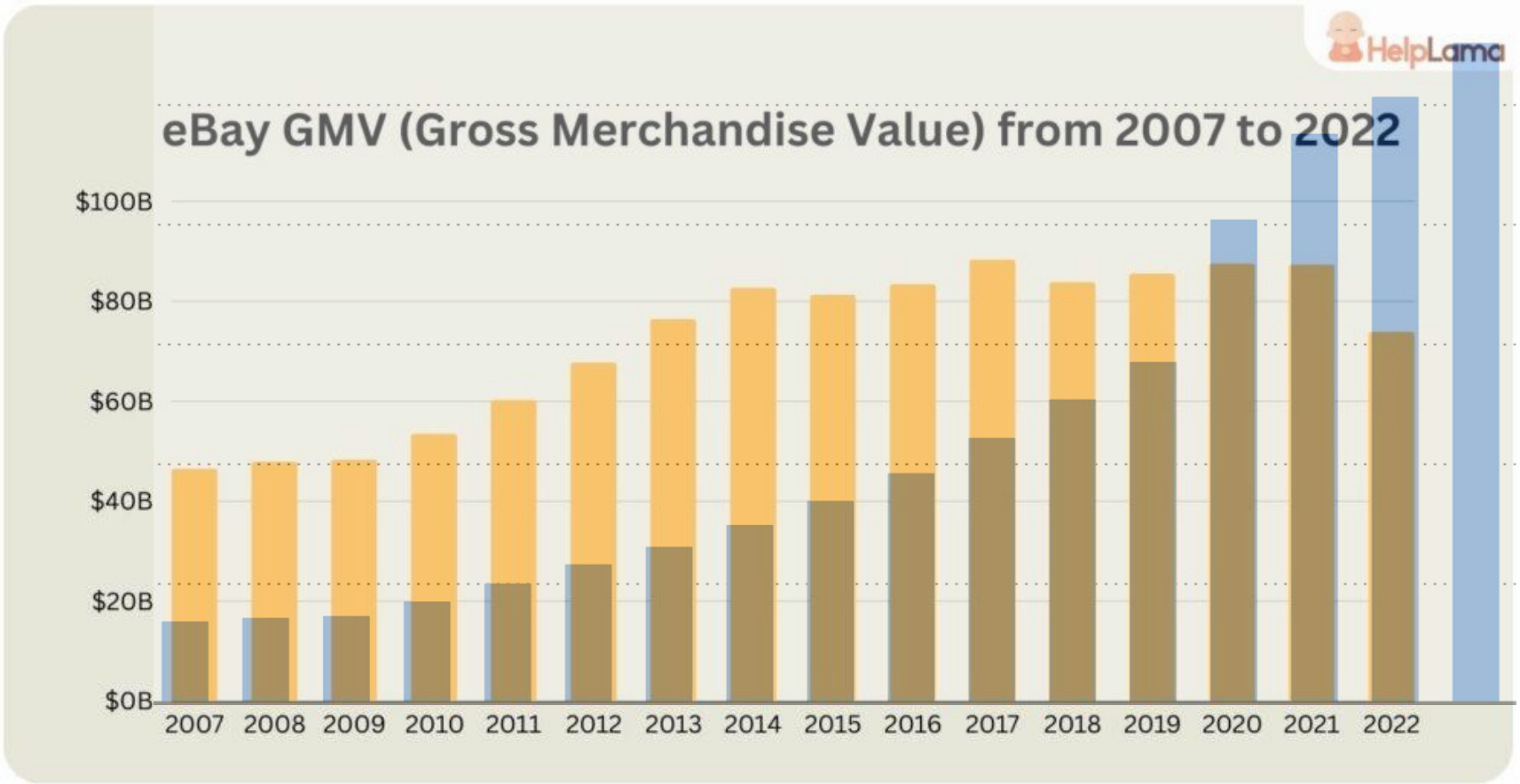
LeadDev London 2026

Randy Shoup

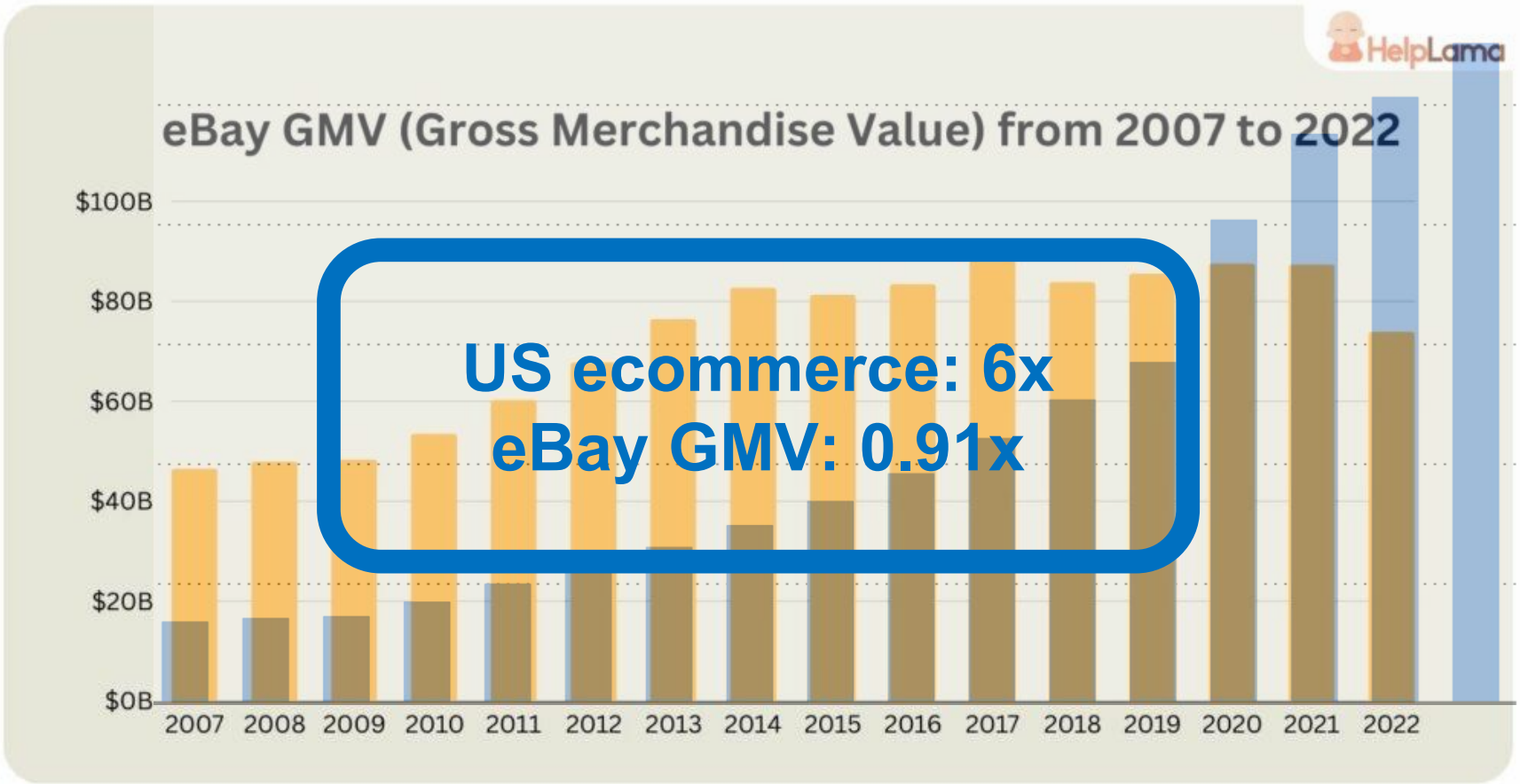
2007+: “A Household Name with a Flat Business”



2007+: “A Household Name with a Flat Business”



2007+: “A Household Name with a Flat Business”



“I need you to come back as Chief Architect -- to shake things up and to bring eBay into the modern world.”

-- eBay CTO

eBay in 2020: Slower Than Market Competitors

- 3000 engineers (~400 teams) in Core Product organization
- 2000 engineers in Core Technology
- Multi-quarter Initiatives across 50+ teams
- 4500 applications and services
- Deployment Frequency: **1-2 / month**
- Lead Time for Change: **10 days**

Velocity Initiative 2020–2025

We Doubled Engineering Productivity

2x Features and Bug Fixes

- Deployment Frequency improved 10x
- Lead Time improved 5x
- Change Failure Rate improved 3x
- Time to Recover improved 3x

DORA / Accelerate Metrics: 2020

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

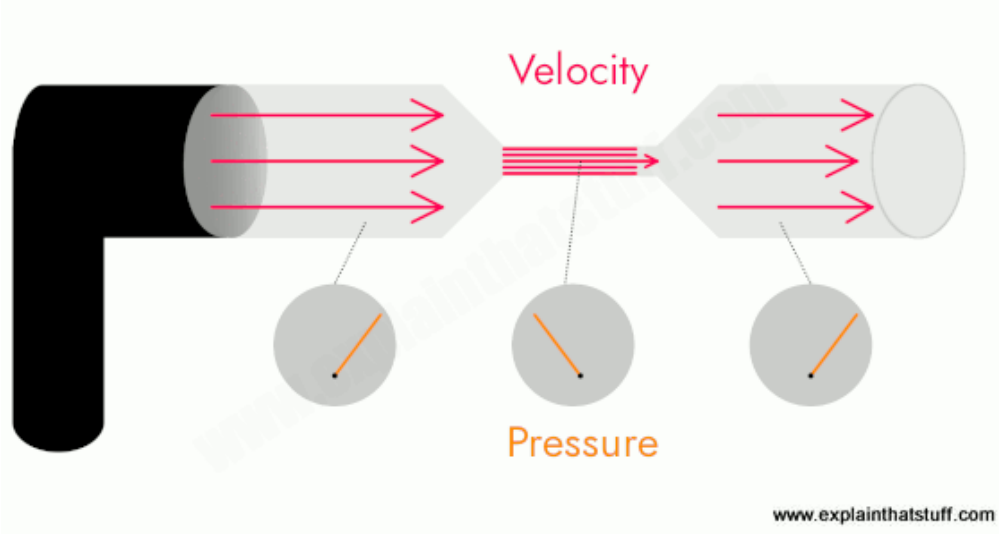
DORA / Accelerate Metrics: 2022-2025

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

We Doubled Engineering Productivity

DevOps Playbook

- Focused on identifying and removing bottlenecks
- Reduced build, test, startup, PR validation times
- Invested heavily in Staging environment
- Automated upgrades, testing, deployment, Site Speed
- Streamlined team processes, code reviews, ”Partner Signoffs”



How We Worked

Collaborate

- Cross-functional leadership
- Embedding model
- Platform and Product Engineering teams working together

Communicate

- Daily leadership standups
- Weekly Team-of-Teams meeting
- Weekly Deep Dives with teams
- Monthly Operating Review

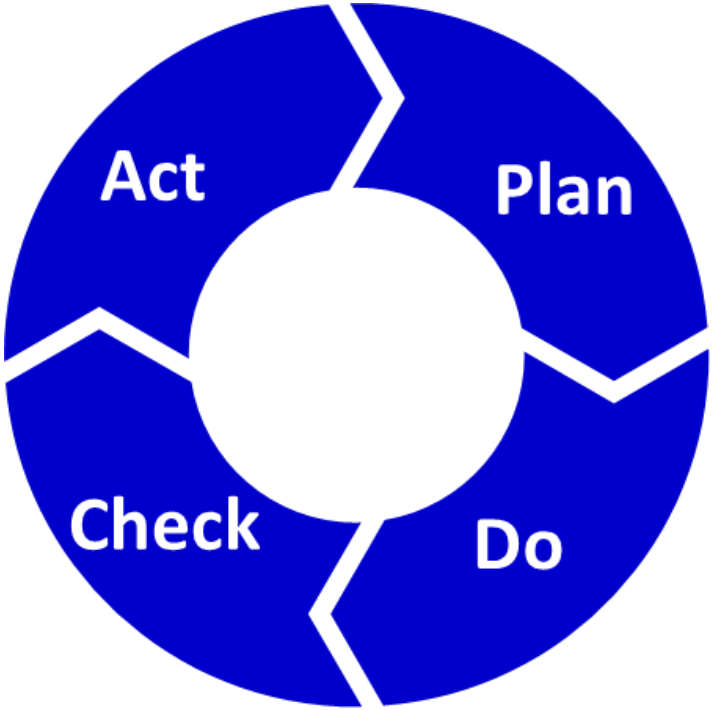
How We Worked

Measure

- DORA as outcome metrics
- “Developer Friction” as input metrics
- Instrumented entire delivery pipeline with timing and success metrics
- Dashboard for every app, team, org

Iterate

- Identified and removed impediments to flow
- Tight PDCA (Plan-Do-Check-Act) cycles



“If I told you that you had to deploy your application every day, please tell me all the reasons why you can’t.”

“OK, here are all the reasons …”

“Thanks! You just gave Platform Engineering our backlog."

Culture and Behavior

Excitement and Fun

- Regular weekly progress on metrics
- Teams inspiring each other to improve

Community and Sharing

- Teams automated their own workflows (PR reminders, Performance testing, Accessibility testing)
- Regular team demos
- Shared tools and learning broadly

Culture and Behavior

Partnership

- Made it psychologically safe to highlight impediments outside team
- Partnered with Security, Compliance, Accessibility, Localization, etc.

Executive Support

- CEO constantly highlighted in All Hands
- Board of Directors and Quarterly Earnings
- “This is the most important initiative at the company. Go faster!”

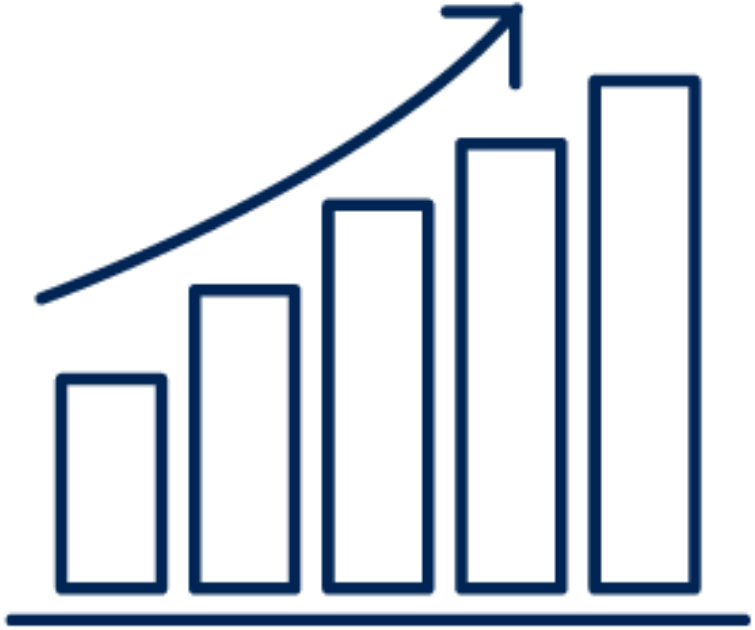
Scaling the Initiative

Quarterly Cohorts

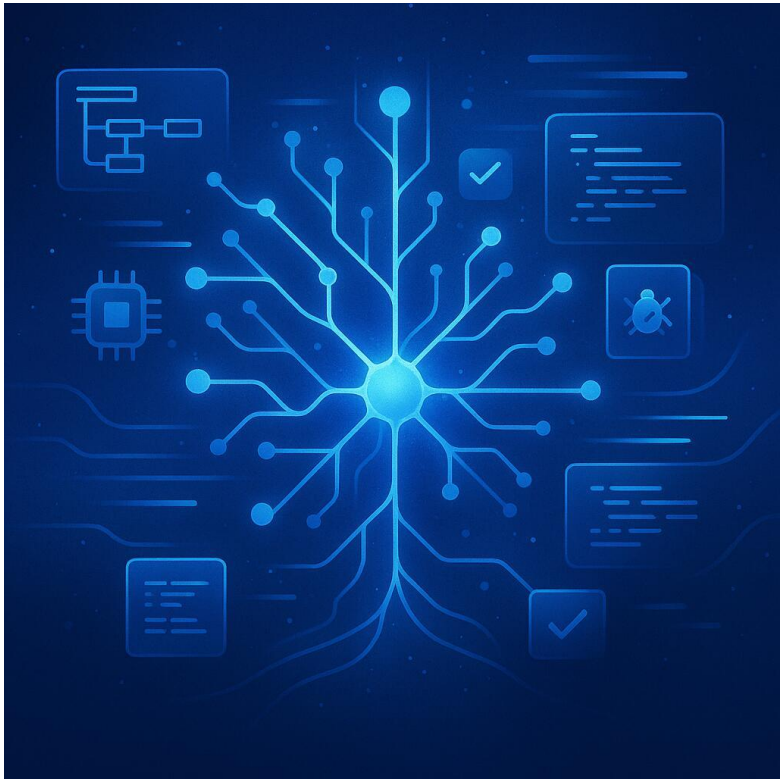
- Pilot cohorts in 2021 immediately oversubscribed
- Expanded quarter-by-quarter to include all 400 teams, 3000 engineers, 4500 applications

Automation

- Regular deployments for every app and service, even if not actively maintained
- Automated “patch pipeline” for security vulnerabilities, dependencies, legacy APIs, etc.



AI Across the SDLC



AI Across the SDLC

- Code generation, test generation, test data
- Legacy code migrations
- PR summarization and code review
- CI pipeline management (test failures, automatic RCAs, predictive optimization)
- Deployment monitoring and automated rollbacks
- Developer support and feedback analysis

Mobile Modernization

Modular Architecture

- Modularized apps into domains, aligned to the org structure
- Build domain mini-apps independently

Release Management

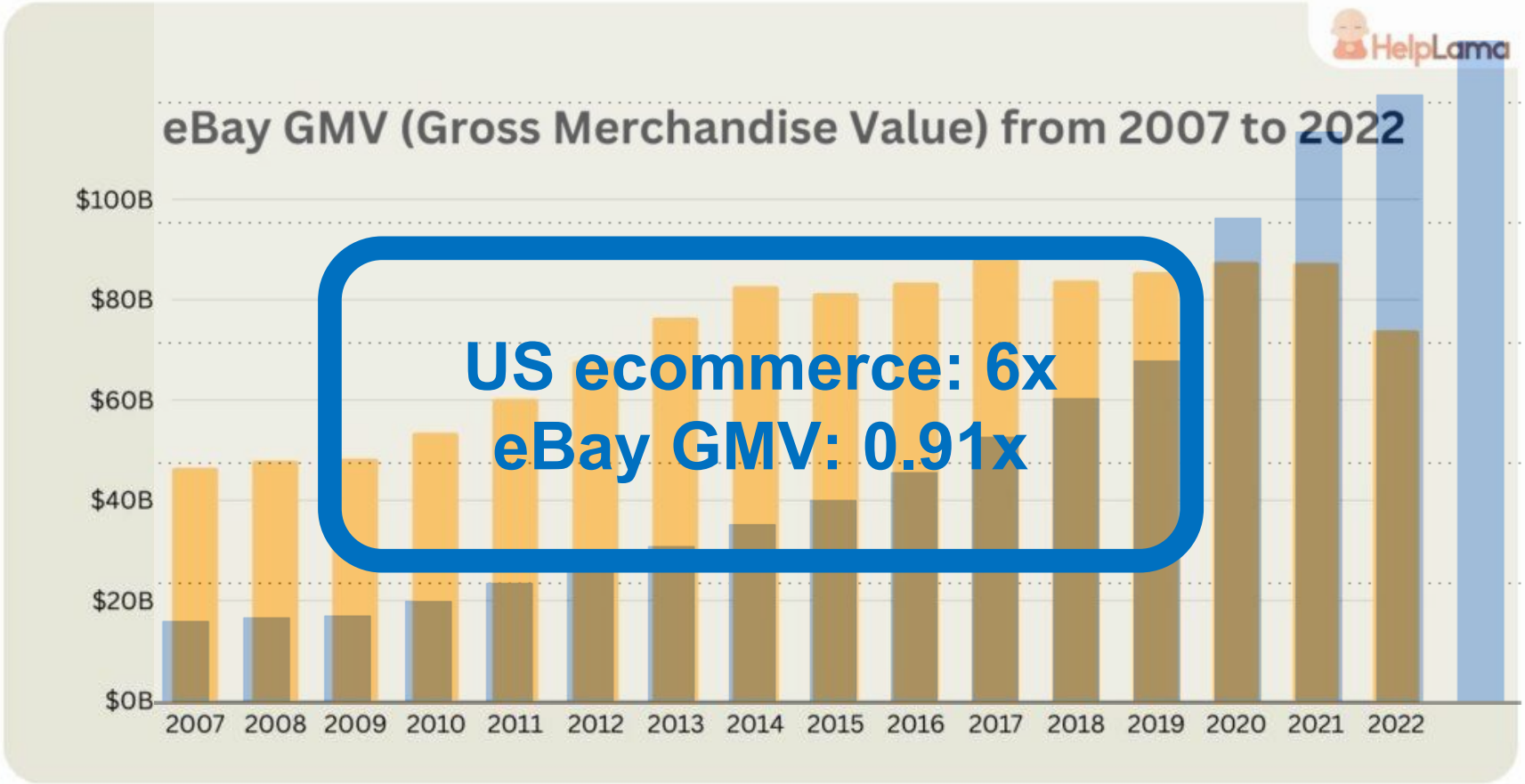
- Monthly → Weekly → Daily capability
- Ship Blockers and “Scuttling”
- Seeded Rollouts and Feature Flags
- Blameless Retrospectives

“You were kind enough to help me adapt and see the light, through air cover and rational small tests of the process” – *Mobile Release Manager*

“What would Randy do?”

-- *Velocity* team

2007+: “A Household Name with a Flat Business”



Why Did *Velocity* Not Save the Company?

1. Strategy and Planning
2. Execution and Delivery
3. Organizational Culture

1. Strategy and Planning

2. Execution and Delivery

3. Organizational Culture

Strategy and Planning

Innovators Dilemma

- Unwilling to disrupt historical business model
- Competitors disrupt or arbitrage

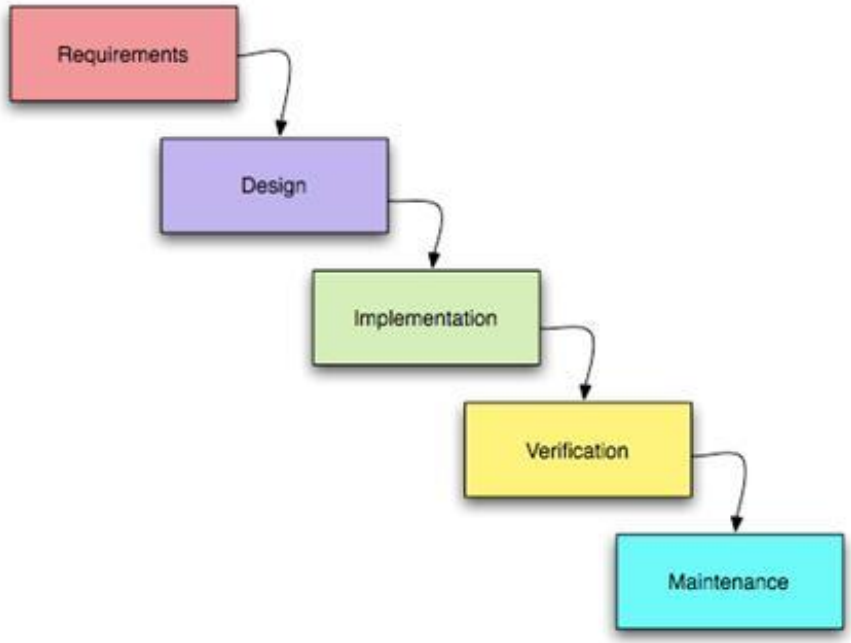
Learned Helplessness

- Flat business for ~15 years
- Highly risk-averse
- Every user-facing change met with near-revolt (“*Seller Straightjacket*”)

Strategy and Planning

Centralized Waterfall Planning

- Annual multi-month company-wide planning cycle
- Work can only happen if approved by executive team
- Work can only be approved if it is big enough to get on the initiative list
- Smaller projects can only survive by being tacked on to massive initiatives as “riders”



1. Strategy and Planning

2. Execution and Delivery

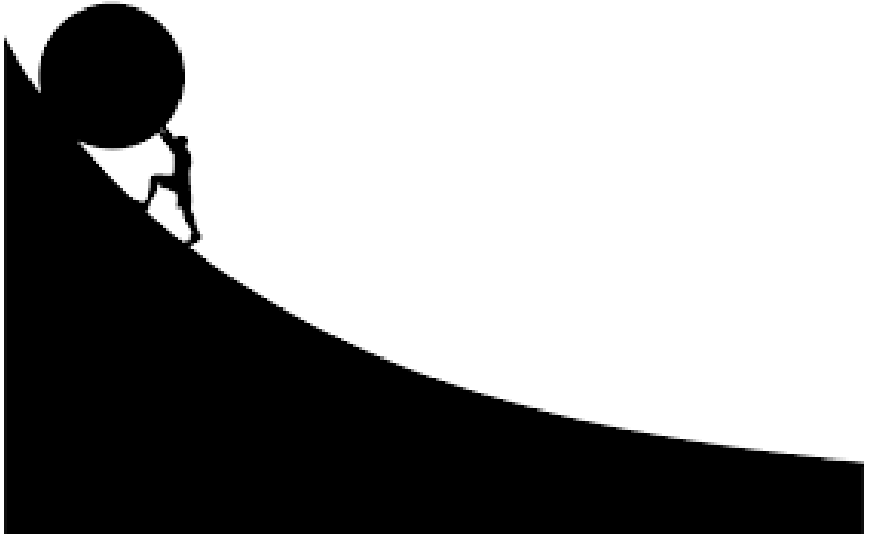
3. Organizational Culture

Execution and Delivery

Massive Coordinated Releases

- Cycle time measured in quarters or years
- Commonly involve **50 or more teams**
- E.g., **eBay Managed Payments**
  - 3 years, 2000 engineers
  - $1.5B in personnel costs alone
  - Resulted in **slower payments to sellers**

Execution and Delivery



“Feature Factory”

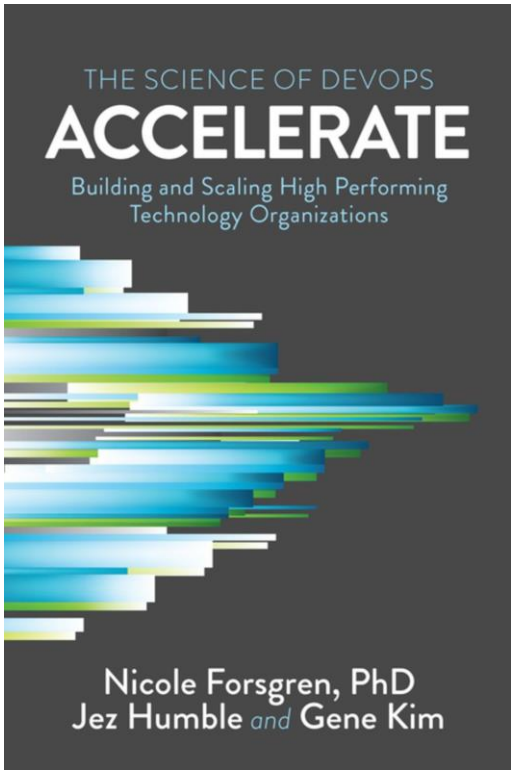
- Outputs, not Outcomes
- Rewards for milestones and effort, not metrics and customer value
- E.g., **Train Seats**
  - “We delivered 5000 train seats to the business this quarter”
  - “Our organization worked 10,000 person-weeks this quarter”
  - “Our organization spent $60M this quarter”

1. Strategy and Planning

2. Execution and Delivery

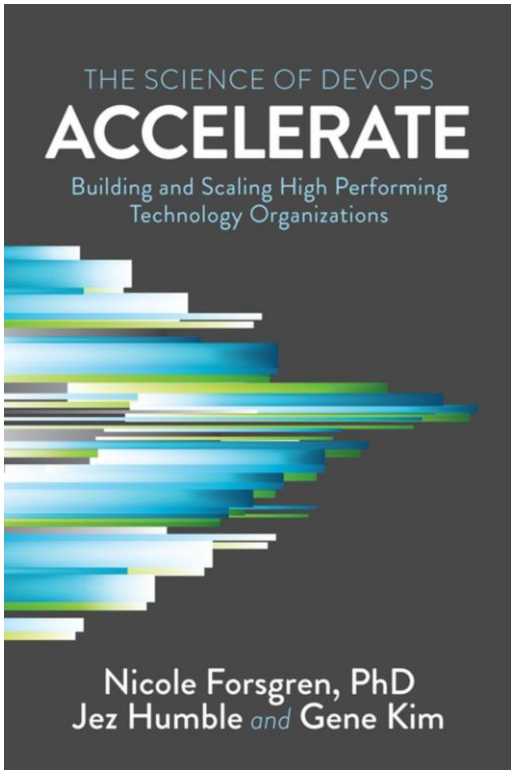
3. Organizational Culture

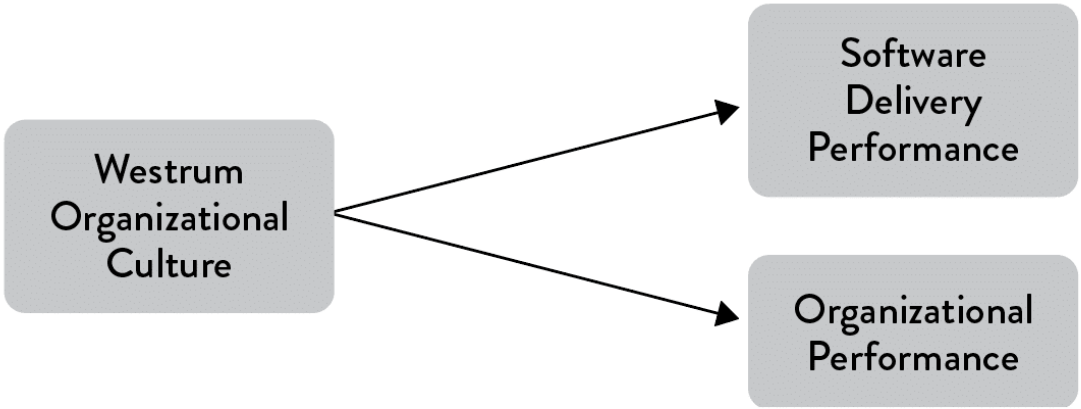
Culture as Foundation



Pathological (Power-Oriented)	Bureaucratic (Rule-Oriented)	Generative (Performance-Oriented)
Low cooperation	Modest cooperation	High cooperation
Messengers "shot"	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Culture as Foundation





Pathological Organization

“Culture of Fear”

- Highly political and risk-averse organizational culture
- Acknowledging failure seen as rude or threatening

Executive Empire Building

- Zero sum, scarcity mindset
- Maximize span of control at expense of other teams
- Maximize size of team within overall flat headcount

Pathological Organization

eBay Exceptionalism

- Insular culture of *Not Invented Here*
- Industry-standard approaches resisted and rejected by default
- Long employee tenures, minimal cross-pollination inside or outside

Top-Down, Waterfall

- No real-time autonomy for teams or individuals
- Experiments used to confirm initial biases
- Experiments used to make sure nothing broke

VP “X”

Culture of Terror

- Engineers in constant fear of making any mistake
- Threatened high performers with poor reviews if they left the team
- Exceptional engineers and leaders became internal refugees or left the company entirely

Empire Building

- More than 700 employees and contractors to build Buyer Experience

Faux Agile

- Multi-year projects, regularly delayed
- “Planning Sprints”, “Design Sprints”, “Development Sprints”, “QA Sprints” …
- Personally approved all deployments for more than a year

Karma in Action

- Maneuvered behind the scenes to fire Chief Architect in 2022
- Fired by CPO 6 months later

What I Learned

1. Top-Down, Bottom-Up, Middle-Out

- *Engage peer execs as allies from the start*

2. Route Around Resistance

- *Better to bypass than to appear to threaten*
- *Legacy execs are masters at survival*

3. See the Whole Board

- *Demonstrate results and credibility with Software Delivery, then tackle Planning*

4. Easier to Transform a Malleable Organization