Engineering Effectiveness: Achieving your goals with less

Ryan Murray
Executive Technical Director, Client Solutions
Thoughtworks

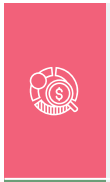
“[In the majority of organizations], software developers only spend 30-40% of their time on feature development”

- State of DevOps Report (2018,2019)

“The Developer Coefficient” - Stripe 2018

18 million	Estimated developers in the world
$17,000	Global GDP per capita
$51,000	GDP per developer
$918 billion	Aggregate GDP of developers globally
31.6%	Efficiency loss of developers (from survey)
~$300 billion	**Global GDP loss from developer inefficiency annually**

Why focus on Engineering Effectiveness?



Cost of engineering is skyrocketing / Need to “do more with less”

Talent acquisition & retention

Competitivity & Time to Value

Predictability

Industry Trend

Over the last 3+ years, industry leading organizations have taken a **systematic approach to engineering productivity**

Internal engineering effectiveness working groups @ Amazon, Google, Spotify, Etsy, and so on)

“You can’t (effectively) measure engineering productivity directly, but you can measure & eliminate waste.”

(paraphrased)

- Google, Nicole Forsgren, and many others

“The system that people work in and the interaction with people may account for 90 or 95 percent of performance.”

- W. Edwards Deming

What drains productivity and blocks flow?

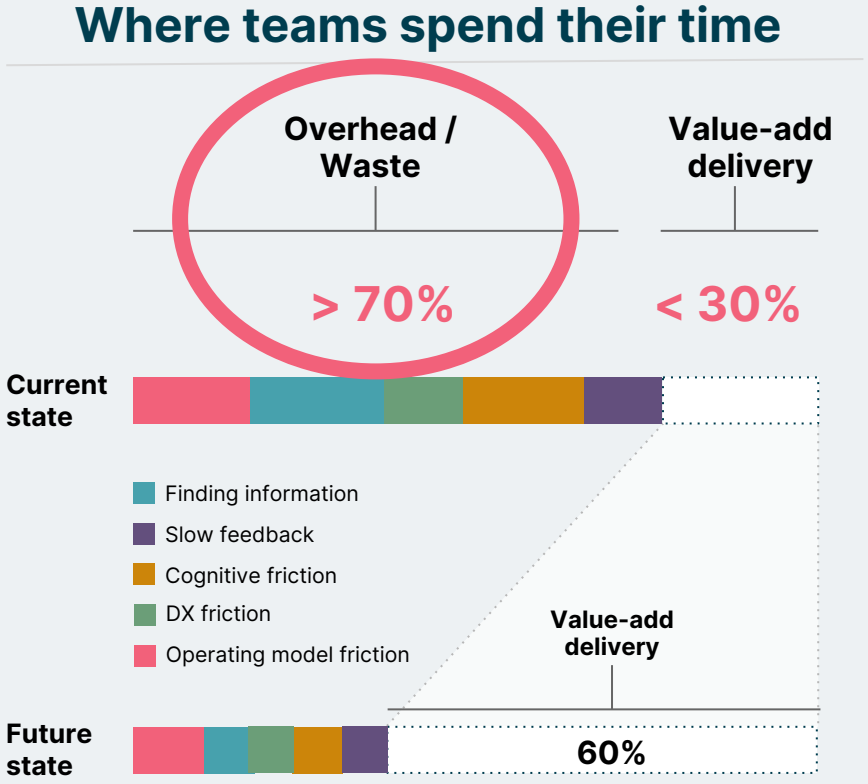
Developer Experience friction

Friction in finding essential information

Cognitive overload / task switching

Slow feedback loops (quality,functionality)

Operating model friction



Beyond the DORA 4 Key Metrics

change lead time | deployment frequency | mean time to restore (MTTR) | change fail %

> By focusing on the factors that predict high delivery performance —
>
> - **goal-oriented generative culture**
> - **a modular architecture**
> - **engineering practices that enable continuous delivery**
> - **effective leadership**
>
> — we can scale deployments per developer per day linearly or better with the number of developers. This **allows our business to move faster as we add more people**, not slow down, as is more typically the case.

Accelerate (2017)

- goal-oriented generative culture
- a modular architecture
- engineering practices that enable continuous delivery
- effective leadership

The Five “Impact Zones” of Engineering Effectiveness

Flow of the System & Strategic Alignment

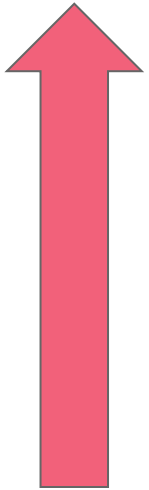
Playing with Legos - Architecture & Composability

Lifting All Boats - Platform Engineering & Accelerators

Flow of the Team - Culture, Autonomy, Goal Orientation

Flow of the Individual - Individual Skills, Tools

Impact



Engineering effectiveness transformation

Map the software delivery value stream and prioritize for maximum impact

Product and engineering value stream and impact analysis

Platform engineering capabilities
CI/CD/Deployment orchestration
Platform-wide observability & trace
Shift-left security, compliance & audit
Cloud cost management & optimization

DevEx & productivity accelerators
Microservices architecture -
"starter kits", "paved paths", API Authoring
toolchain Architecture pattern reference
implementations Design systems

Enterprise testing strategy
Platform test strategy (beyond the "test pyramid")
Service virtualization / contract testing
Ephemeral environments & testing in production
Test data management, generation, PII obfuscation

Domain-driven design & cloud architecture
Strategic DDD practice (incl build vs buy)
Platform API strategy (incl. events & data)
Architecture & design for cloud economics
Architecture governance - efficiency and
interoperability

Scaling knowledge & leadership
Searchable service & data catalogs (e.g. Backstage)
Documentation standards,
Inner-sourcing practices, CoEs & communities
Engineering workflow, behavioral, & code metrics

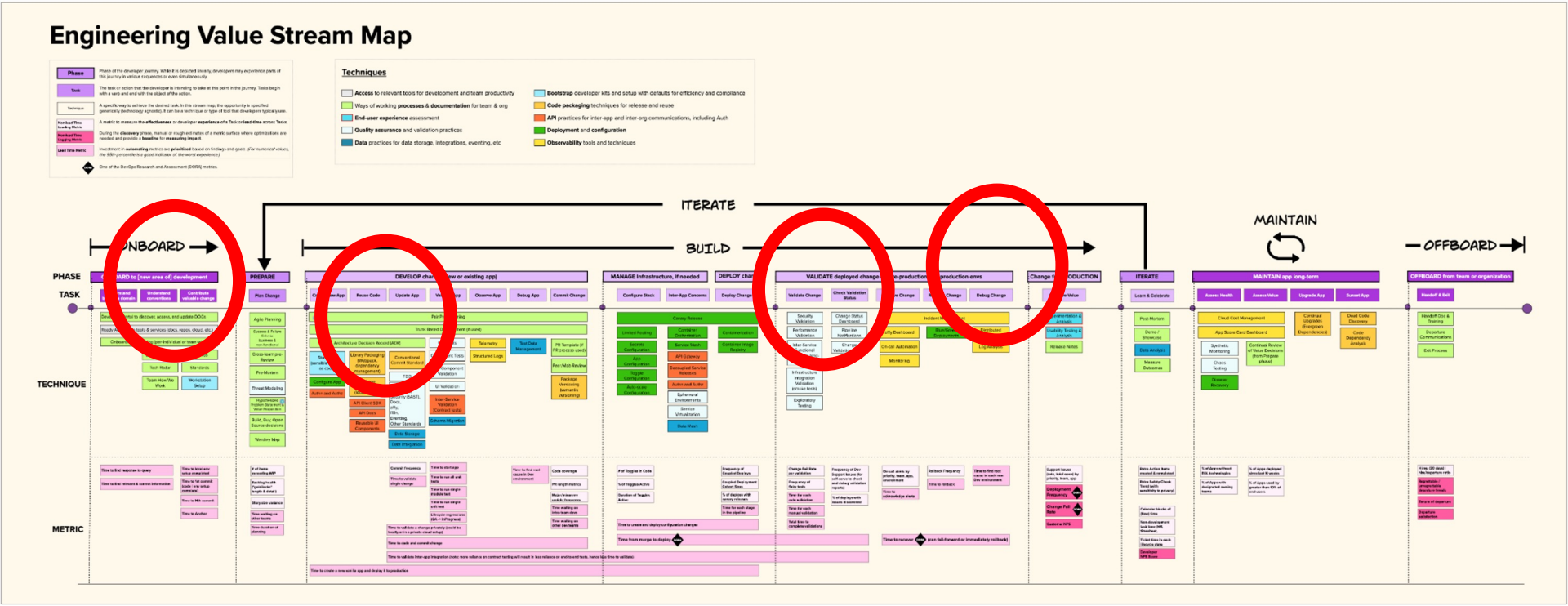
Operating model & governance
Product engineering operating model & roles (TPM)
Scaling alignment & strategy (OKRs, etc)
Team topologies & autonomy alignment
Work allocation, value delivery tracking

Generative AI

Organizational enablers optimized for engineering transformation

Communication, culture, strategy, OCM	Upskilling & enablement program	Talent management & recruitment

Focus on the high cost friction and waste



Measure what matters

(Not what is easy to measure)

Actionable leading indicators of high performance

These are example measures that drive focus quickly to root cause issues.

% strategic aligned capacity

% late stage defect discovery

Avg consecutive focus hours / day

Story flow efficiency (backflow)

Indicator	Low Effectiveness	High Effectiveness
Validate a local code change works	2 mins	5-15 seconds
% late stage defect discovery	>20%	<5%
% capacity aligned to strategic work	<30%	>60%
Find root cause for defect	4-7 days	4 hours
Rate of adoption of paved paths & standards	<30%	>75%
Work in progress (stories) by engineer	3+	1
Get answers to an internal Information / technical query	2 days	30 mins
Pipeline run / queue times & failures	>2 hours / >30%	20 min / 5%
Launch a new service in production	2-4 months	3 days
Consecutive focus hours per day	1-2 hours	4+ hours
Story flow efficiency	<50%	90%
Code review / PR merge times	>3 days	<1 day

Measure and govern the impact

Value theme

Time to Value
Accelerate value delivery

Predictability
More consistently meet your commitments

Talent Acquisition & Retention
Attract and retain the right talent

Reduce Cost
Increase throughput through efficiency and reduced waste

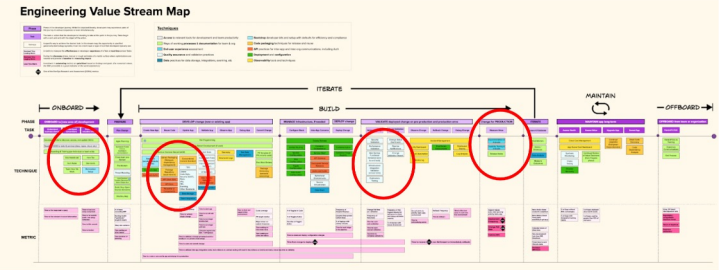
Measurable outcome (KR)

Initiative: Developer Onboarding

Measure: **Time to right information** Target: **decrease 50%**
Measure: **Commit rate** Target: **increase 3x**
Measure: **Time to "hello world" in prod** Target: **decrease 5x**

Initiative: Enterprise Testing Strategy

Measure: **Defect escape rate** Target: **decrease 2x**
Measure: **Test feedback cycle time** Target: **decrease 5x**
Measure: **Share test env queue time** Target: **decrease 3x**



Specific initiatives

- Improve documentation & findability (dev portal)
- Develop starter kits
- Developer environment and tooling standardization
- Invest in communities and org change management
- Shift-left test coverage normalization

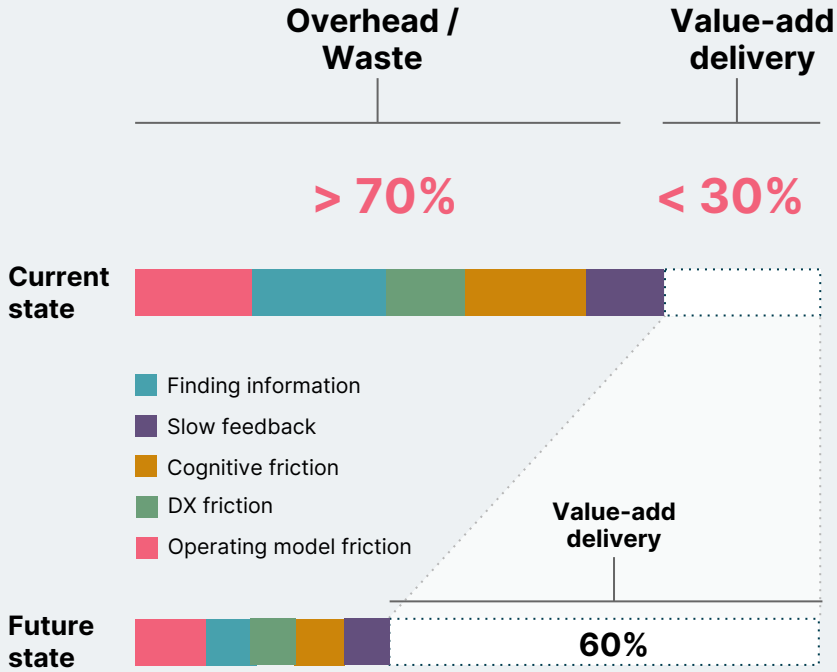
Generative AI

(Your mileage may vary)

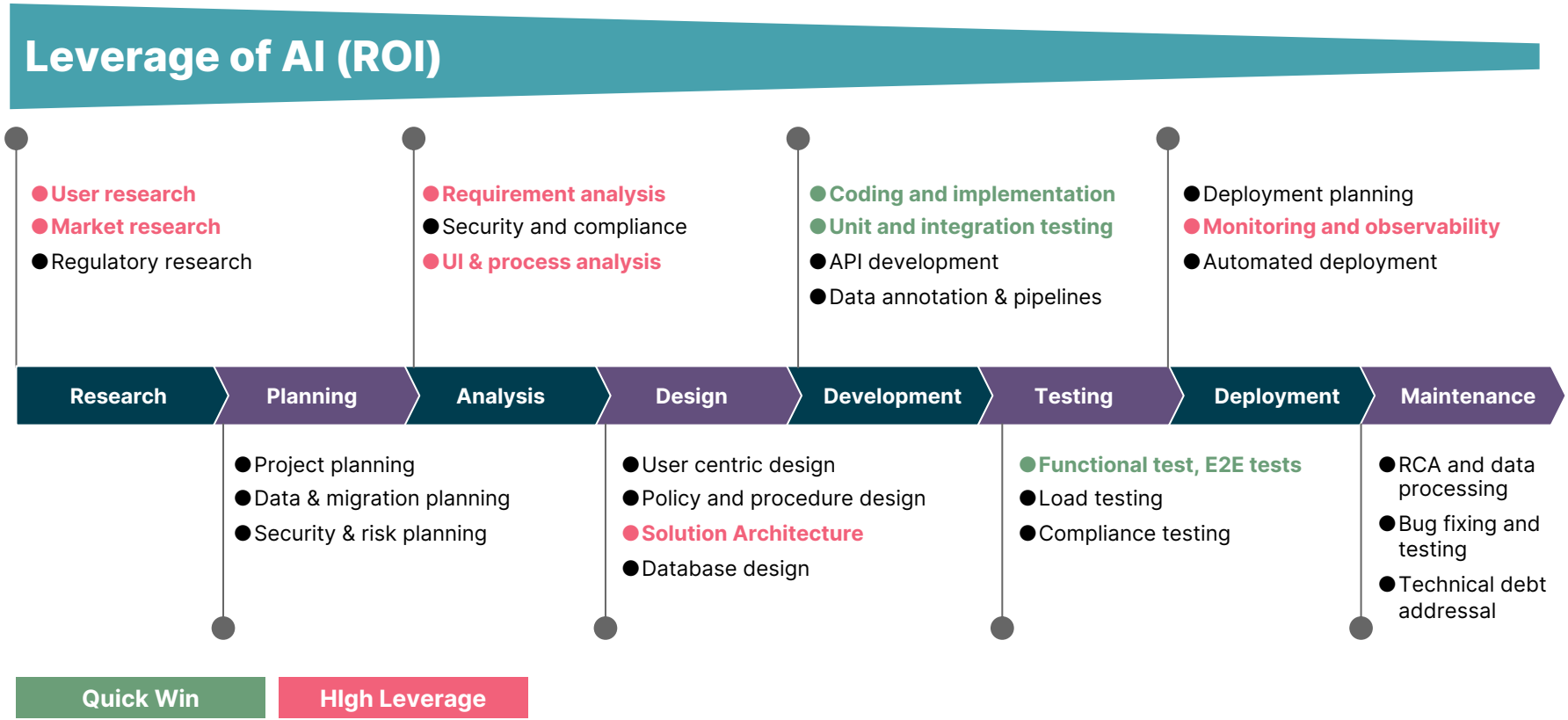
Gen AI addresses common impediments to flow

Probable Impact	Examples of common areas of friction
MED	DevEX friction
HIGH	Finding information
MED	Cognitive overload / task switching
MED	Slow quality feedback loops
LOW	Operating model friction

Where teams spend their time



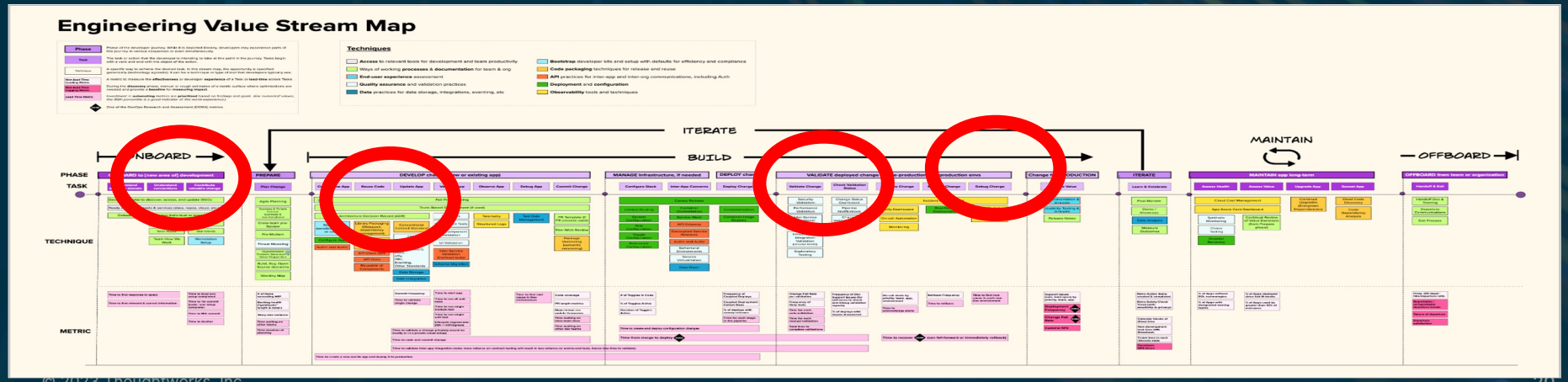
Focus on quick wins and high leverage SDLC areas



Focus on what matters

Accelerate ::

- goal-oriented generative culture
- a modular architecture
- engineering practices that enable continuous delivery
- effective leadership



Leadership

(at all levels)

is the critical factor

Three essential questions to execute on engineering effectiveness

1. What process and incentives need to change?

2. What is your strategy to get there?

3. Where is your data to prove it?

Engineering Effectiveness

5 / 3 / 1

Impact Zones

Essential Questions

Result