### Unlocking High-Velocity Development Strategies, Tactics, and Metrics

### Ravikumar Buragapu

September 2023



## Agenda

- 7 Key Factors for Building a High-Velocity Development Teams
- Internal Developer Platforms (IDPs)
- Service Maturity Insights (SMI)
- Gen AI for DevPro Transformations
- Key Insights





### 7 Key Factors for Building a High-Velocity Development Teams



Integrated workflows and platforms



World-class tools



Robust product-management function



Developer Velocity and Experience Metrics



Culture of unwavering psychological safety



Talent management focus on developer experience



Rapid adoption



## Internal Developer Platforms (IDPs)

#### A modern way to enable engineering teams

IDPs automate software development tasks and manage infrastructure, striking a balance between standardization and developer flexibility. Key benefits include alignment with standards, compliance support, developer focus, autonomy, and efficiency.



Primary goal: To optimize software development and manage infrastructure for speed.



IDPs strategically boost developer productivity and streamline software development.



An IDP is a complete ecosystem of tools, services, and processes.



Adopting IDPs drives faster, standardized, and efficient software practices.



### SMI (Service Maturity Insights) Metrics that matter most

- Select metrics aligned with your organization's goals
- Opt for actionable metrics
- Opt for measurable metrics
- Prioritize sustainable metrics
- Emphasize real-time tracking
- Explore Service Maturity Scorecards and campaigns



#### Service Maturity Insights (SMI) Framework





#### DORA A framework to measure four key software delivery performance factors, DORA analytics track progress, pinpoint improvement opportunities, and benchmark against other teams effectively

DORA Metrics							
Metric	Description	Measurement Criteria	Example Value				
Deployment Frequency	The frequency of deployments to production.	Number of deployments per unit of time (e.g., per day).	3 deployments/day				
Lead Time for Changes	The time it takes to go from code commit to deployment.	Hours or days required for a change to be deployed.	4 hours				
Change Failure Rate	The percentage of changes that result in a failure or require remediation.	Percentage of failed changes out of total changes.	2%				
Mean Time to Recover	The average time it takes to recover from a failed change or incident.	Time in minutes or hours.	30 minutes				

## SPACE

## A modern framework emphasizing software delivery efficiency, SPACE metrics highlight bottlenecks and improvement opportunities.

A research-based approach to measuring, understanding, and improving developer productivity

Level	Satisfaction	Performance	Activity	Collaboration	Efficiency
<b>Individual</b> One person	<ul> <li>Developer satisfaction</li> <li>Retention*</li> <li>Satisfaction with code reviews assigned</li> <li>Perception of code review</li> </ul>	<ul> <li>Code Review Velocity</li> </ul>	<ul> <li>Number of code reviews completed</li> <li>Coding time</li> <li># commits</li> <li>Lines of code*</li> </ul>	<ul> <li>Code review score (quality of thoughtfulness)</li> <li>PR merge times</li> <li>Quality of meetings*</li> <li>Knowledge sharing, discoverability (quality of documentation)</li> </ul>	<ul> <li>Code review timing</li> <li>Productivity perception</li> <li>Lack of interruptions</li> </ul>
<b>Team</b> People who work together	<ul><li>Developer satisfaction</li><li>Retention*</li></ul>	<ul> <li>Code Review Velocity</li> <li>Story points shipped*</li> </ul>	<ul> <li># story points completed*</li> </ul>	<ul> <li>PR merge times</li> <li>Quality of meetings*</li> <li>Knowledge sharing, discoverability (quality of documentation)</li> </ul>	<ul> <li>Code review timing</li> <li>Handoffs</li> </ul>
<b>System</b> (End-to-End work like a development	<ul> <li>Satisfaction with engineering system (e.g., CI/CD pipeline)</li> </ul>	<ul> <li>Code Review Velocity</li> <li>Code review (acceptance rate)</li> <li>Customer</li> </ul>	<ul> <li>Frequency of deployments</li> </ul>	<ul> <li>Knowledge sharing, discoverability (quality of documentation)</li> </ul>	<ul> <li>Code review timing</li> <li>Velocity/flow through the system</li> </ul>

EPIs are custom measurements for each team or organization. EPIs may measure any team or organization-important software delivery performance.

Engineering Performance Indicators (EPI)						
Metric	Description	Measurement Criteria	Example Value			
Code Quality	The overall quality of the codebase, including readability and maintainability.	Code review scores, adherence to coding standards, and code complexity.	Code review score of 9/10.			
Defect Density	The number of defects or bugs identified in the code per unit of code.	Number of defects per 1,000 lines of code (KLOC).	5 defects per KLOC.			
Code Churn	The frequency of code changes or churn in the codebase.	Number of code commits or changes per week.	50 code changes per week.			
Test Coverage	The percentage of code covered by automated tests.	Percentage of code lines covered by tests.	80% test coverage.			

#### Gen Al Tools: Transforming Developer Productivity





### Key Insights



Build unified platforms for development



Benchmark Your Performance



Look at the Leading Indicators



Institute Teamwork Norms



Harnessing the Power of Generative AI



Focus on Enhancing Developer Experience

# Let's Connect!



<u>buragapu@adobe.com</u>



Twitter: @buragapu

www.adobe.com/careers



Linkedin: <u>https://www.linkedin.com/in/ravikumar-buragapu-a966292/</u>

