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

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

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

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

<table>
<thead>
<tr>
<th>Level</th>
<th>Satisfaction</th>
<th>Performance</th>
<th>Activity</th>
<th>Collaboration</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Individual</strong></td>
<td>One person</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Developer satisfaction</td>
<td>Code Review Velocity</td>
<td>Number of code reviews completed</td>
<td>Code review score (quality of thoughtfulness)</td>
<td>Code review timing</td>
</tr>
<tr>
<td></td>
<td>Retention*</td>
<td></td>
<td>Coding time</td>
<td>PR merge times</td>
<td>Productivity perception</td>
</tr>
<tr>
<td></td>
<td>Satisfaction with code reviews assigned</td>
<td></td>
<td># commits</td>
<td>Quality of meetings*</td>
<td>Lack of interruptions</td>
</tr>
<tr>
<td></td>
<td>Perception of code review</td>
<td></td>
<td>Lines of code*</td>
<td>Knowledge sharing, discoverability (quality of documentation)</td>
<td></td>
</tr>
<tr>
<td><strong>Team</strong></td>
<td>People who work together</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Developer satisfaction</td>
<td>Code Review Velocity</td>
<td># story points completed*</td>
<td>PR merge times</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retention*</td>
<td></td>
<td></td>
<td>Quality of meetings*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code Review Velocity</td>
<td></td>
<td></td>
<td>Knowledge sharing, discoverability (quality of documentation)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story points shipped*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>(End-to-End work like a development)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction with engineering system (e.g., CI/CD pipeline)</td>
<td>Code Review Velocity</td>
<td>Frequency of deployments</td>
<td>Knowledge sharing, discoverability (quality of documentation)</td>
<td>Code review timing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EPIs are custom measurements for each team or organization. EPIs may measure any team or organization-important software delivery performance.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
<th>Measurement Criteria</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect Density</td>
<td>The number of defects or bugs identified in the code per unit of code.</td>
<td>Number of defects per 1,000 lines of code (KLOC).</td>
<td>5 defects per KLOC.</td>
</tr>
<tr>
<td>Code Churn</td>
<td>The frequency of code changes or churn in the codebase.</td>
<td>Number of code commits or changes per week.</td>
<td>50 code changes per week.</td>
</tr>
<tr>
<td>Test Coverage</td>
<td>The percentage of code covered by automated tests.</td>
<td>Percentage of code lines covered by tests.</td>
<td>80% test coverage.</td>
</tr>
</tbody>
</table>
Gen AI Tools: Transforming Developer Productivity

- Analysis
  - Requirements writing and analysis
  - User story generation

- Design
  - Architecture writing assistance
  - Sequence, flow diagram generation
  - Data Model authoring
  - UX design assistance

- Development
  - Code generation
  - Debugging
  - Explain code
  - Improve consistency
  - Code translation

- Testing
  - Test cases writing
  - Testing code generation

- Deployment
  - Continuous integration/Continuous deployment generation
  - Infrastructure as Code script writing support
  - Automation script writing assistance

- Maintenance
  - Performance monitoring and remedy suggestion
  - AI-assisted support
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!

buragapu@adobe.com

Twitter: @buragapu

www.adobe.com/careers

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