



The Developer Productivity Engineering *MATURITY MODEL*

--==

Contents

1. Introduction: What is Developer Productivity Engineering?	1
2. Overview: What is the Developer Productivity Engineering Maturity Model	1
3. Objectives: Why do we need a DPE Maturity Model?	2
4. Process: What was the process for developing the model?	2
5. How to use the model	3
6. Roadmap: Future DPE Maturity Model development	4
7. Executive Overview: DPE Maturity Model at a glance	5
8. Drill down: DPE Maturity Model by level	7

1. Introduction: What is Developer Productivity Engineering?

Developer Productivity Engineering (DPE) is a software development practice used by leading software development organizations to maximize developer productivity and happiness.

As its name implies, DPE focuses on improving developer productivity through an engineering approach rather than a management approach. This approach relies on automation, actionable data, and acceleration technologies to deliver measurable outcomes like faster feedback cycles and reduced mean-time-to-resolution (MTTR) for build and test failures. As a result, DPE has quickly become a proven practice that delivers a hard ROI with little resistance to product acceptance and usage.

Organizations successfully apply the practice of DPE to achieve their strategic business objectives such as reducing time to market, increasing product and service quality, minimizing operational costs, and recruiting and retaining talent by investing in developer happiness and providing a highly satisfying developer experience. DPE accomplishes this with processes and tools that gracefully scale to accommodate ever-growing codebases.

Gradle is pioneering the practice of DPE and Gradle Enterprise serves as a key enabling technology and solution platform.

2. Overview: What is the Developer Productivity Engineering Maturity Model

As organizations adopt and deploy DPE, they will go through several stages of maturity. Because DPE is a relatively new practice, the maturity model itself needs to mature. Thus, the maturity model proposed here (i.e., Version 1) is an attempt at a formal description of how this practice will evolve and mature in a broad range of IT settings. Having said that, it is most relevant to organizations with a thousand or more developers and should still be useful for teams with at least 500 developers.

Typical of many maturity models across business and IT disciplines, the DPE Maturity Model consists of 5 maturity levels. They are:

- **Level 1: Idling** – DPE as a practice, concept, and set of tools does not exist
- **Level 2: Dabbling** – DPE activity is opportunistic and informal
- **Level 3: Experimenting** – DPE activity is purposeful; leadership is paying attention
- **Level 4: Practicing** – DPE resources are dedicated; results are being reported
- **Level 5: Scaling** – DPE is being institutionalized and rolled out across the company

There are five dimensions of maturity at each level. They are:

- **People** – The users and practitioners of DPE tools.
- **Technology** – Technologies developed to improve developer productivity by focusing on the developer toolchain (e.g. feedback-cycle acceleration, observability, and analytic technologies).
- **Reporting & business impact** – DPE metrics and outcomes that are collected and shared within and across teams and management stakeholders, such as feedback-cycle times, and cost avoidance.
- **Leadership** – The IT management chain (as opposed to users/practitioners).
- **Processes** – The informal and formal DPE-related processes and workflows.

3. Objectives: Why do we need a DPE Maturity Model?

The model intends to help DPE practitioners, deployment leaders and executives:

- Benchmark where their companies stand in relation to broader adoption patterns experienced by comparable organizations within and across industries.
- Assess the areas of strength and performance gaps in their deployments.
- Pinpoint specific steps they might take to close gaps and graduate to their next journey stage.
- Communicate progress to developer productivity stakeholders in the practitioner, management, and executive communities to garner support for existing efforts or make the case to expand the initiative's footprint.

4. Process: What was the process for developing the model?

The evidence for the descriptions of the various maturity characterizations presented in the model is largely anecdotal and not based on a formal study. Instead they are based on Gradle Inc.'s vast and deep experience working directly with hundreds of companies. There are few data points at the

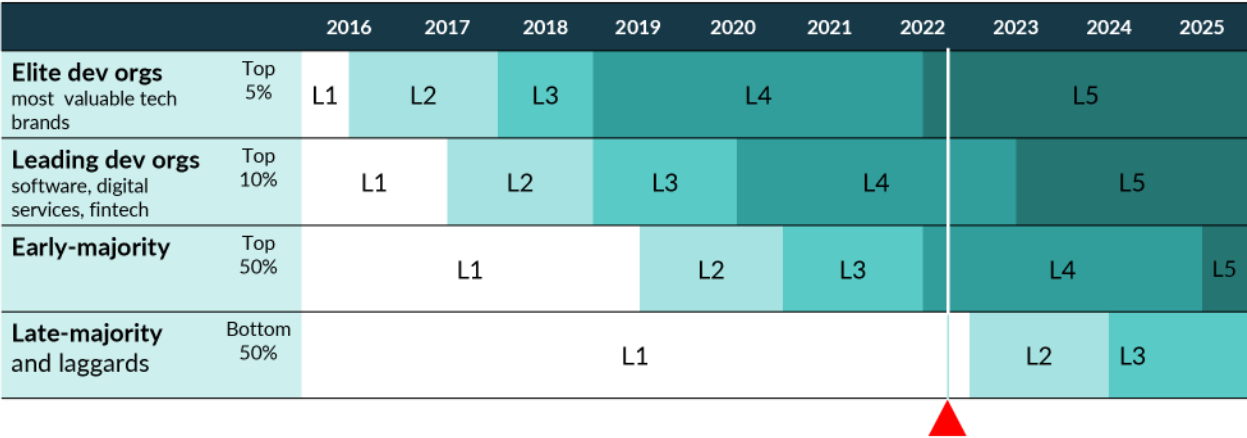
higher maturity levels due to the practice's newness. As a result, the descriptions and characterizations presented here include extrapolations based on the ample historical patterns associated with adopting enterprise-wide, IT-led strategic initiatives.

The next version of the maturity model (Version 2) will be based on a more qualitative (statistical) analysis of DPE adoption.

5. How to use the model

Team leaders can use a qualitative self-assessment to determine one’s maturity level. The best way to do that is to assess your maturity level along each of the five dimensions of the model by deciding which description best reflects your scenario. Then using the eye test, see where the individual assessments best cluster to determine your overall level of maturity. Another approach is adding your maturity level scores and divide by five. Five is a useful number because it may better characterize your maturity level (e.g., a score of 2.4 indicates your maturity level is best described at Level 2, but the organization is well on its way to achieving Level 3).

The model can also benchmark your maturity level against industry patterns. It can answer questions like how does my maturity level compare to elite development teams or against the industry average. The graphic below proposes some benchmarks that can be used for this purpose. But, note again that the data is anecdotal and not based on a qualitative/statistical study.



Where does your level of DPE maturity compare to the elite, leaders, and laggards?

6. Roadmap: Future DPE Maturity Model development

As stated above this is a Version 1 tool and more of a hypothesis than a statistical model. To address the resulting shortcomings, a planned Version 2 will be supported by a formal user/practitioner survey and analysis. The aim is a model that incorporates:

- A more sophisticated assessment process that integrates organizational data on DPE adoption patterns
- More granular industry benchmarks that can answer questions like how does my maturity level compare against companies in my vertical industry sector, development teams of a similar size, or companies operating in my geography
- A gap analysis comparison between DPE maturity level and CMMI maturity level to understand if DPE maturity is a relative strength or weakness in your overall software process
- Data on how long it typically takes to traverse the maturity model levels (e.g., how long can I expect to spend at a particular maturity level?)

7. Executive Overview: DPE Maturity Model at a glance

The matrix below brings together the five maturity levels with the five maturity dimensions. The descriptions in the cells attempt to capture concisely the defining characteristic of organizations at each maturity level along a given dimension. More elaborate descriptions of each entry are provided next (See section 8. Drill-down: DPE Maturity Model level details).

	Level 1: Idling	Level 2: Dabbling	Level 3: Experimenting	Level 4: Practicing	Level 5: Scaling
Current Status of Industry Adoption	Laggard (Bottom 10%)	Late Majority Bottom 50%	Early Majority Top 50%	Early Adopter (Top 10%)	Visionary (0-5%)
PEOPLE	None There is no acknowledged responsibility for DPE even on an informal basis.	Informal Team leaders take informal responsibility for their teams.	Formalizing Team leaders are formally responsible; no dedicated DPE individuals or team, except build engineers who see DPE as an extension of their traditional role.	Dedicated A dedicated DPE team/practice supports multiple teams.	EPMO A formal dedicated Enterprise DPE PMO coordinates the DPE initiative enterprise-wide.
TECHNOLOGY	Accidental DPE technology is not sought out and implemented for the purpose of improving developer productivity.	Out-of-the-box Acceleration technologies are used out of the box to speed up feedback cycles. Still there is no effort to monitor, maintain or improve gains or use data to troubleshoot problems.	Reactive Basic data observability and analytics to reactively manage, build and test performance, and troubleshoot failures.	Proactive Data and trend observability are used regularly to proactively manage performance and failure rate trends and take preventative measures.	Biz Insights Teams are committed to being early adopters of new DPE technology innovations such as AI/ML based analytics to gain maximum observability and automatic anomaly detection.

<p>REPORTING & BUSINESS IMPACT</p>	<p>None There is no purposeful business impact.</p>	<p>Tactical Tactical and short-term developer productivity gains can be measured and anecdotally improve the developer experience in a noticeable if not quantifiable manner.</p>	<p>Measuring Feedback cycle times can be measurably improved and the software developer experience is viewed as a recruiting and retention advantage, but reporting is still relatively tactical (e.g., no comparisons to historical baselines).</p>	<p>Strategic Productivity goals and gains are tracked in a strategic manner (e.g., with a richer set of metrics and comparison data). Improvements can be projected based on historical and other data.</p>	<p>Transformational Metrics are reported at the highest levels and across orgs. Executive dashboards enable monitoring with a focus on direct impact to high-level business metrics such as revenue/TTM, quality, brand impact.). Executives may comment on or report results publicly</p>
<p>LEADERSHIP</p>	<p>Unimportant Developer productivity and the developer experience is not a business consideration and is not considered by management to be a burning platform.</p>	<p>Unprioritized Developer productivity and the developer experience is not a priority and depends on the heroic efforts of a few developers or build engineers motivated by low-hanging fruit gains.</p>	<p>Engaged Developer productivity and the developer experience (creative flow/context switching impact) is an emerging concern and on management's radar, motivated by repeatable success.</p>	<p>Advocating Developer productivity is an IT management priority and led by dedicated professionals with senior leadership support.</p>	<p>Crusading DPE has executive sponsorship and buy-in at all levels of software development team leadership enterprise wide. Executives are out-spoken public champions of DPE.</p>
<p>PROCESSES</p>	<p>None There are no DPE processes in place.</p>	<p>Ad Hoc Processes are <i>ad hoc</i> and generally not repeatable.</p>	<p>Collaborative Processes for collaboration gain traction and start to be widely socialized and used successfully.</p>	<p>Reliable Dedicated teams formalize trusted processes for monitoring and optimizing performance and troubleshooting.</p>	<p>Institutionalized All processes are highly reliable and institutionalized and followed when "nobody is looking".</p>

8. Drill down: DPE Maturity Model by level

Level 1: Idling – DPE as a practice, concept, and set of tools does not exist	
PEOPLE	<p>None</p> <ul style="list-style-type: none"> • There is no acknowledged responsibility for DPE even on an informal basis. • Developers are often accustomed to the pain of slow feedback cycles and inefficient troubleshooting and accept it as an expected hazard of being a developer. • Decisions that directly impact developer experience are made arbitrarily based on anecdotal data.
TECHNOLOGY	<p>Accidental</p> <ul style="list-style-type: none"> • Tools may be deployed that effect improvements to developer productivity, but DPE technology is not sought out and implemented for that express purpose. • Developer productivity management tools (e.g. DPM tools that collect activity-based metrics like time spent performing tasks) or output-based metrics like SLOC written may be used .
REPORTING & BUSINESS IMPACT	<p>None</p> <ul style="list-style-type: none"> • DPE metrics are not tracked, and dark costs are present but uncalculated. • Rudimentary activity-based or output-based DPM metrics may be tracked. • A lack of meaningful DPE metrics creates confusion and wasted cycles. • Developer experience is being perceived through a narrow lens.
LEADERSHIP	<p>Unimportant</p> <ul style="list-style-type: none"> • Developer productivity and the developer experience is not an acknowledged issue (there is no burning platform and/or no data to understand the volume of waste). • Leadership may only be aware of a DPM-based approach. • Developer experience is being perceived through a narrow lens.
PROCESSES	<p>None</p> <ul style="list-style-type: none"> • There are no formal or informal DPE processes or workflows. • DevOps practices may be implemented, but these practices are not sufficiently addressing DPE’s critical pain points.

Level 2: Dabbling – DPE activity is opportunistic and informal	
PEOPLE	<p>Informal</p> <ul style="list-style-type: none"> • Team leaders take informal responsibility for their teams. • Developers may be familiar with individual tools and may not be aware that they fall under the umbrella of DPE tools. Further, they may not even be familiar with the term DPE or the broader practice.
TECHNOLOGY	<p>Out-of-the-box</p> <ul style="list-style-type: none"> • A limited set of acceleration technologies are used out of the box in a single or small number of projects to speed up feedback cycles. Still, there is no effort to monitor, maintain or improve gains or use data to troubleshoot problems. • Some data gathering tools may be implemented to gain insights about build behavior, but that data is suboptimal and difficult to correlate with developer experience.
REPORTING & BUSINESS IMPACT	<p>Tactical</p> <ul style="list-style-type: none"> • Tactical and short-term developer productivity gains can be measured and anecdotally improve the developer experience in a noticeable, if not quantifiable, manner. • DPE tribal knowledge is born.

LEADERSHIP	<p>Unprioritized</p> <ul style="list-style-type: none"> Developer productivity and the developer experience are not a priority and depend on the heroic efforts of a few developers or build engineers motivated by low-hanging fruit gains. Leadership is, in general, still not paying attention.
PROCESSES	<p>Ad Hoc</p> <ul style="list-style-type: none"> Processes are <i>ad hoc</i>, generally not repeatable, and do not fully address DPE's pain points The lack of comprehensive data skews the interpretation of actual developer experience, making it difficult to devise and test effective processes.

Level 3: Experimenting – DPE activity is purposeful; leadership is paying attention	
PEOPLE	<p>Formalizing</p> <ul style="list-style-type: none"> Team leaders or developer platform engineers are formally responsible, and Developer Productivity Engineering is often a formal responsibility in project leader job descriptions. There are no dedicated DPE individuals or teams except build, platform, and/or DevOps engineers who see DPE as an extension of their traditional role.
TECHNOLOGY	<p>Reactive</p> <ul style="list-style-type: none"> A growing set of acceleration technologies is being optimized for multiple projects as performance problems are uncovered. Basic data observability and analytics are used to reactively manage build and test performance and troubleshoot failures. Awareness of using DPE to observe longer-term performance and failure trends and manage potential issues proactively is in its initial phase.
REPORTING & BUSINESS IMPACT	<p>Measuring</p> <ul style="list-style-type: none"> Feedback cycle times and failure rates can be measurably improved, leading to confidence that software delivery times and quality has improved significantly. The software developer experience is viewed as a developer recruiting and retention advantage, but reporting is still relatively tactical (e.g., no comparisons to historical baselines).
LEADERSHIP	<p>Engaged</p> <ul style="list-style-type: none"> Developer productivity and the developer experience (creative flow/ context switching impact) is an emerging concern on the mgmt chain's radar, motivated by replicable success. Leadership is beginning to inquire about metrics and impact.
PROCESSES	<p>Collaborative</p> <ul style="list-style-type: none"> Processes for collaboration gain traction and start to be widely socialized and used successfully (e.g., Build Scan link sharing). Operational responses to analytics data, such as flaky test analysis, are being considered and prototyped.

Level 4: Practicing – DPE resources are dedicated; results are being reported	
PEOPLE	<p>Dedicated</p> <ul style="list-style-type: none"> A dedicated DPE team/practice may be in place that supports multiple engineering teams

	<ul style="list-style-type: none"> Developer platform engineers – responsible for ensuring developer productivity and a positive developer experience – may have designated DPE as a standard “plank” in their platform that includes a curated set of tools (e.g. IDE, collaboration, CI services, etc.) for onboarding and on-going developer support.
TECHNOLOGY	<p>Proactive</p> <ul style="list-style-type: none"> Multiple acceleration technologies have been optimized for an increasing number of projects and are being combined for maximum effect in local, remote workstation, and CI environments Data and trend observability are used regularly to proactively manage performance and failure rate trends and take preventative measures Development platform teams integrate DPE technology as a standard component.
REPORTING & BUSINESS IMPACT	<p>Strategic</p> <ul style="list-style-type: none"> Developer productivity gains are tracked in a more sophisticated manner (e.g., with a richer set of metrics and comparison data). Productivity goals are in place, and improvements can be projected based on historical and other data. DPE data/metrics may be integrated with BI tools such as Tableau.
LEADERSHIP	<p>Advocating</p> <ul style="list-style-type: none"> Developer productivity is a priority and is led by dedicated professionals with senior leadership support. Decisions that directly impact developer experience are made using hard data.
PROCESSES	<p>Reliable</p> <ul style="list-style-type: none"> Dedicated teams formalize trusted processes for monitoring and optimizing performance and troubleshooting. DPE metrics and impact are being continuously provided to leadership.

Level 5: Scaling – DPE is being institutionalized and rolled out across the company	
PEOPLE	<p>EPMO</p> <ul style="list-style-type: none"> A formal dedicated Enterprise DPE PMO coordinates the DPE initiative enterprise-wide. Capacity needs for Developer Productivity Engineers are well understood and the business can predictably acquire talent based on project and engineering growth. DPE maturity is regularly mentioned during recruitment as a significant benefit to employment.
TECHNOLOGY	<p>Biz Insights</p> <ul style="list-style-type: none"> Every available and appropriate DPE-recommended acceleration, observation, and analytics technology is being utilized and optimized Teams are committed to being early adopters of new DPE technology innovations, such as AI/ML-based analytics, to gain maximum observability and detect anomalies.
REPORTING & BUSINESS IMPACT	<p>Transformational</p> <ul style="list-style-type: none"> DPE metrics are reported at the highest levels and across organizations. Executives regularly monitor DPE outcomes via an executive dashboard with a focus on direct impact to high-level business metrics (revenue/ TTM, quality, brand impact, etc.). Executives may comment or report results publicly (e.g. quarterly financial calls and annual reports).
LEADERSHIP	<p>Crusading</p> <ul style="list-style-type: none"> DPE has executive sponsorship and buy-in at all levels of software dev team leadership enterprise wide. Executives are out-spoken public champions of DPE.

PROCESSES	<p>Institutionalized</p> <ul style="list-style-type: none">• All processes are highly reliable and institutionalized and followed when “nobody is looking.”• The focus is on continuous improvement and optimization.• Integration with DPE tooling is automated and part of the standard platform rollout.
-----------	--