



Hans Dockter on Developer Productivity

Giovanni Asproni 

From the Editor

In episode 628 of “Software Engineering Radio,” Hans Dockter, the creator of the Gradle build tool and founder of Gradle Inc., the company behind the developer productivity platform Develocity, joins Giovanni Asproni (<https://se-radio.net/team/giovanni-asproni/>) to talk about developer productivity. They start with some definitions and an explanation of the importance of developer productivity, its relationship with cognitive load, and the big impact that development tools have on it. Hans describes how to implement developer productivity metrics in an organization as well as warns about some pitfalls. The episode closes with a discussion of Hans’s views on the future of this discipline as well as some near-term developments and expectations. We provide summary excerpts next; to hear the full interview, visit <http://www.se-radio.net> or access our archives via RSS at <http://feeds.feedburner.com/se-radio>.—
Robert Blumen and Karthik Vaidhyanathan

Hans Dockter: Developer productivity is about how close the actual productivity is to the maximum productivity your development team could provide to you. It’s not just about how many features are developed per unit of time. Every metric we use to quantify productivity is a hypothesis about how close the actual productivity is to the maximum potential. These roadblocks keep actual productivity far from its potential. Addressing them can lead to transformational improvements.

It’s important because there’s a lot of waste in software development, and by investing in this area, organizations can achieve significant gains in their output and efficiency.

Giovanni Asproni: What is developer productivity engineering (DPE), and how is it different from traditional approaches to improving productivity?

Hans Dockter: DPE focuses on the toolchain, which I see as the biggest barrier to developer productivity, especially in enterprises. Traditionally,

work on continuous integration (CI) or build systems was viewed as niche, but DPE professionalizes this by removing roadblocks like slow feedback and build failures. It’s different from traditional approaches that might focus on broad issues like the work environment. DPE targets the biggest productivity blocker—the toolchain—and aims to fix it. This approach is transformational because it directly addresses what holds developers back from reaching their full potential.

Giovanni Asproni: What is the relationship between developer productivity

SOFTWARE ENGINEERING RADIO

Visit www.se-radio.net to listen to these and other insightful hour-long podcasts.

RECENT EPISODES

- 635—Stevie Caldwell, senior engineering technical lead at Fairwinds, joins host Priyanka Raghavan (<https://se-radio.net/team/priyanka-raghavan/>) to discuss zero-trust network reference architecture.
- 634—Jim Bugwadia, CEO of Nirmata and a committer to the Kyverno projects, joins host Robert Blumen (<https://se-radio.net/team/robert-blumen/>) to discuss policy-as-code and the open source Kyverno project.
- 632—Goran Petrovic, a staff software engineer at Google, speaks with host Gregory M. Kapfhammer (<https://se-radio.net/team/gregory-kapfhammer/>) about how to perform mutation testing on large software systems.
- 629—Emily Bache, founder of the Samman Technical Coaching Society and author of several books about technical agile coaching, talks with SE Radio host Sam Taggart (<https://se-radio.net/team/sam-taggart/>) about katas and the importance of practice.

UPCOMING EPISODES

- Steve Smith, founder and principal architect at Nimble Pros, joins host Jeff Doolittle for a conversation about software quality.
- Nick Tune and Jean-Georges Perrin join Giovanni Asproni to talk about the approach to modernizing legacy systems they propose in their book *Architecture Modernization*, published by Manning.
- Catherine Nelson, author of the new O'Reilly book *Software Engineering for Data Scientists*, discusses the collaboration between data scientists and software engineers, an increasingly common pairing of machine learning and AI projects, with host Philip Winston.
- Ganesh Datta, cofounder of cortex.io (<http://cortex.io/>), joins host Robert Blumen for a conversation about production readiness.

and developer experience? Further, how does the toolchain impact developer productivity and experience, especially in terms of cognitive fatigue?

Hans Dockter: The relationship between developer productivity and developer experience is closely tied to the toolchain. Developers write code, which is like a hypothesis, and need fast feedback from their

tools—whether it's a compiler, tests, or other systems. When this feedback loop is fast, developers stay in the flow. But when it's slow or unreliable, it takes the fun out of development, leading to frustration and burnout. This is where cognitive fatigue sets in. Developers get demotivated when they're constantly waiting for feedback or dealing with flaky tests or other toolchain

issues. The more unreliable the toolchain, the more mental energy gets wasted, leading to burnout and reduced productivity.

Giovanni Asproni: Do you have examples of how poor tool chains impact productivity?

Hans Dockter: A common example is when a pull request build fails, and the developer has no idea why. This happens in many companies where 5–10% of CI builds fail due to flakiness in the pipeline. Developers waste time trying to figure out if the failure is due to their code or the environment. Often, they just rerun the build, and if it passes, they move on. This causes major disruptions and frustration. Some companies even have teams dedicated to triaging failed CI builds, handling tens of thousands of tickets a year. The flakiness creates a lot of waste and uncertainty, and developers are stuck waiting, unable to proceed, which severely impacts productivity.

Giovanni Asproni: Besides the toolchain, what other factors, like cognitive load or code quality, affect developer productivity?

Hans Dockter: Besides the toolchain, cognitive load and code quality play significant roles in developer productivity. Cognitive load comes into play when developers face tasks that require cognitive control—where the goal is clear, but the path to achieving it isn't. This leads to cognitive fatigue, which drains mental energy. Ideally, all cognitive fatigue would convert into productive work, but often, it's wasted on nonproductive tasks like unclear requirements or poor code quality. For example, legacy code without tests



ABOUT THE AUTHOR



GIOVANNI ASPRONI is the chief technology officer and cofounder of Launch Ventures and a software engineering consultant based in London, U.K. Contact him at aspro@computer.org.

or documentation makes work much harder. Microsoft¹, for instance, measures productivity by tracking how quickly a new developer makes their first pull request. They found that the earlier the first pull request, the faster subsequent ones follow. This highlights how cognitive load, especially during onboarding, can impact productivity. Reducing cognitive fatigue, improving code quality, and streamlining onboarding are key factors that go beyond the toolchain in boosting developer productivity.

Giovanni Asproni: What are some of the key metrics or indicators teams can use to measure developer productivity effectively?

Hans Dockter: A key metric is the number of feedback cycles a developer initiates daily—how often they request feedback from the toolchain. Faster and reliable feedback loops lead to smaller and more frequent commits, which is a sign of healthy workflows. Build times, both for local builds and pull requests, are also crucial. Teams

with 1-min build times often run twice as many feedback cycles compared to those with 4-min build times, which act as a productivity tax. Observability is essential—measuring build times, dependency downloads, and test failures helps identify where productivity is lost, allowing teams to prioritize fixes effectively.

Giovanni Asproni: What are the first steps an organization should take to improve developer productivity, and who is responsible for owning these metrics?

Hans Dockter: The first step is observability—measure where bottlenecks exist, like build times or flaky tests. Without data, you're guessing, and it's hard to prioritize fixes. Both management and developers must own this process. C-level involvement is essential to treat productivity as a strategic priority. Developers need to be part of the solution, using metrics to improve the toolchain, not for individual evaluation. If metrics are used to punish developers, they'll game the system, undermining the goal.

Giovanni Asproni: What are some common pitfalls teams encounter when trying to improve productivity?

Hans Dockter: A common pitfall is delaying easy fixes in favor of long-term strategic changes. Teams often hold smaller improvements hostage, waiting for a bigger overhaul that can take years. Another issue is focusing too much on new tools rather than optimizing the existing infrastructure. Developers sometimes push for shiny new technologies while neglecting low-hanging fruit that could immediately improve productivity. It's important to address what's blocking productivity now instead of waiting for an ideal future solution. Organizations need to balance both short-term gains and long-term goals to avoid getting stuck in a cycle of delays and missed opportunities.

Giovanni Asproni: How do you see the future of DPE, and what role will generative AI play in shaping it?

Hans Dockter: The future of DPE will involve deep academic research, similar to chemical process engineering. There's a lot of cognitive science that could be applied to software development, but we're only scratching the surface. Generative AI will play a big role by increasing the pressure on the toolchain. If developers write twice as much code with AI, we need tools to handle the load. AI could also help tackle complex problems like modularizing code, which would improve build times and overall productivity. The key will be making sure the toolchain keeps pace with these advancements. 🌀

¹<https://www.microsoft.com/en-us/research/publication/the-space-of-developer-productivity-theres-more-to-it-than-you-think/>.

IEEE Software (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; +1 714 821 8380; fax +1 714 821 4010. IEEE Computer Society headquarters: 2001 L St., Ste. 700, Washington, DC 20036. Subscribe to *IEEE Software* by visiting www.computer.org/software.

Postmaster: Send undelivered copies and address changes to *IEEE Software*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8, Canada. Printed in the USA.

Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any

third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version that has been revised by the author to incorporate review suggestions, but not the published version with copyediting, proofreading, and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2024 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

ITProfessional

TECHNOLOGY SOLUTIONS FOR THE ENTERPRISE

CALL FOR ARTICLES

IT Professional seeks original submissions on technology solutions for the enterprise. Topics include

- emerging technologies,
- cloud computing,
- Web 2.0 and services,
- cybersecurity,
- mobile computing,
- green IT,
- RFID,
- social software,
- data management and mining,
- systems integration,
- communication networks,
- datacenter operations,
- IT asset management, and
- health information technology.

We welcome articles accompanied by web-based demos. For more information, see our author guidelines at www.computer.org/itpro/author.htm.

WWW.COMPUTER.ORG/ITPRO

Digital Object Identifier 10.1109/MS.2024.3493555



IEEE
COMPUTER
SOCIETY

