CHAPTER 5

A Software Development Productivity Framework

Caitlin Sadowski, Google, USA
Margaret-Anne Storey, University of Victoria, Canada
Robert Feldt, Chalmers University of Technology, Sweden

Productivity is a challenging concept to define, describe, and measure for any kind of knowledge work that involves nonroutine creative tasks. Software development is a prime example of knowledge work, as it too often involves poorly defined tasks relying on extensive collaborative and creative endeavors. As in other areas of knowledge work, defining productivity in software development has been a challenge facing both researchers and practitioners who may want to understand and improve it by introducing new tools or processes.

In this chapter, we present a framework for conceptualizing productivity in software development according to three main dimensions that we propose are essential for understanding productivity. To help clarify productivity goals, we also propose a set of *lenses* that provide different perspectives for considering productivity along these three dimensions. We contend that any picture of productivity would be incomplete if the three dimensions and various lenses are not considered.

C. Sadowski and T. Zimmermann (eds.), *Rethinking Productivity in Software Engineering*,
https://doi.org/10.1007/978-1-4842-4221-6_5

Productivity Dimensions in Software Development

The three dimensions in the proposed productivity framework for software engineering are as follows:

- *Velocity*: How fast work gets done
- *Quality*: How well work gets done
- *Satisfaction*: How satisfying the work is

When trying to define productivity goals or measure productivity, it is important to consider all three of these dimensions because they work together synergistically. Even though productivity is often considered in terms of increased output (higher velocity), an increase in velocity may not correspond to an actual productivity improvement if there is a corresponding drop in the quality of that output. Velocity and quality taken together make up overall work efficiency and effectiveness, while velocity and quality may impact satisfaction in different ways. An increase in velocity may lead to reduced costs (and improve the satisfaction of managers), but at the same time it can lead to increased stress for developers (and reduce their satisfaction and in turn incur future costs). A detailed example of the perils of low satisfaction, even with high velocity and quality, can be found in Chapter 11.

Velocity

The velocity dimension captures how productivity is often conceptualized in terms of the time spent doing a task or the time taken (or cost) to achieve a given quantity of work. How one may conceptualize or measure velocity is highly task dependent, and the type of task needs to be considered, as well as the granularity, complexity, and routineness of a particular task. For example, developer velocity metrics could include the number of story points per sprint or the time taken to go from code to a release.

Quality

The quality dimension encapsulates doing a good job when producing artifacts (such as software) or the quality of provided services. Quality may be an internal consideration in a project (e.g., code quality) or external to a project (e.g., product quality from the perspective of the end users). Metrics for quality in a software project could include

counts of negative characteristics such as post-release defects or self-reported ratings of delays incurred by technical debt.

Satisfaction

Engineering satisfaction is a multifaceted concept, which makes it challenging to understand, predict, or measure. This dimension captures human factors of productivity and has several possible subcomponents, including physiological factors such as fatigue, team comfort measures such as psychological safety, and individual feelings of flow/focus, autonomy, or happiness. Learning or skill development that may positively impact long-term quality, developer retention, or velocity may manifest as an increase in satisfaction. For developers, satisfaction may be impacted by the real or perceived effectiveness of their personal work or their team's work.

Lenses

The three dimensions of productivity can be viewed through different lenses. These lenses may help to narrow a research goal and provide perspective on the subsequent methods we may use to understand or measure productivity. The following are the main types of lenses we feel are important to consider:

- *Stakeholders*: Different stakeholders (e.g., developer, manager, vice president, etc.) may have varied goals and interpretations of any sort of productivity measurement. Before trying to understand and measure productivity, it is essential to identify which stakeholders are of concern and what is important to those stakeholders. It may not be immediately obvious which stakeholders should be considered; a researcher or practitioner may need to carefully elicit which stakeholder perspectives are important.
- *Context*: Particular project, social, and cultural factors will change perceptions of productivity. For example, if developers feel that helping others is valued by their team, then they will feel that time spent answering questions is productive. The underlying development context (e.g., open source projects versus projects

focused on profits) affects productivity goals. Though context lenses are often implicit, sometimes it may be necessary to explicitly consider the impact of any norms, values, or attitudes.

- *Level*: Each lens in the level category represents a particular scale (in terms of group size) at which productivity is considered. Individual developers, teams, organizations and the surrounding community will lead to different perceptions of productivity, and productivity goals may also be in tension across these different groups. An intervention that may benefit one level may not hold at all levels. As a concrete example, interruptions that negatively impact the person who is interrupted may lead to a net gain from a team perspective. For an in-depth look at four different level lenses, see Chapter 6.
- *Time period*: Productivity perceptions vary greatly according to the period of time that is considered (shorter terms such as days, weeks, or sprints or longer terms such as months, years, or milestones). For example, a process change may slow down velocity in the short term but lead to enhanced team learning over time and thus speed up velocity over a longer time period. Similarly, short-term velocity enhancements may lead to fatigue and lower developer satisfaction over a longer period of time.

The Productivity Framework in Action: Articulating Goals, Questions, and Metrics

Given a particular high-level productivity goal, a common desire is to derive specific metrics that track such a goal. Unfortunately, going from goals to metrics is not trivial as metrics are typically proxies for specific aspects of a goal. One technique to bridge this divide is to have an intermediate state under consideration. For example, the goal-question-metric (GQM) approach for understanding and measuring the software process [1, 2] works by first generating "questions" that define goals and then specifying measures that could answer those questions. GQM suggests a systematic approach to do the following:

- **Conceptualize goals** aimed at understanding or improving software engineering tools and processes
- Specify **research questions** to operationalize those goals
- Define **metrics** for understanding or measuring tools and processes

Similar to GQM, the HEART framework is used for measuring usability in design projects [3]. HEART first decomposes a high-level usability goal (such as "my app is awesome") into subgoals, abstract "signals" that could measure those subgoals (e.g., time spent with app), and specific metrics for those signals (e.g., number of shares or number of articles read in app). In addition to this goals-signals-metrics breakdown, the HEART framework splits usability into five dimensions: happiness, engagement, adoption, retention, and task success.

Inspired by the way that the HEART framework involves both splitting by dimensions and breaking down from goals to metrics, we propose splitting into goals, questions, and metrics in combination with the productivity dimensions and lenses. This technique can guide the development of specific questions and metrics toward the concrete productivity goals identified. Such goals include measuring the impact of an intervention, identifying anti-patterns or problem spots causing productivity losses, comparing groups, or understanding productivity for a particular context. To illustrate how the framework may be used, we sketch two hypothetical examples in the following sections.

Example 1: Improving Productivity Through an Intervention

A manager of a software development team (the stakeholder) in a large software company (the context) would like to *improve productivity through the introduction of a new continuous integration system* (the stakeholder's productivity goal). She hopes that productivity will be improved for both individual developers and the team overall (the levels) and intends to measure the change over the time frame of a few months (the time period).

A set of specific questions about productivity improvements arises from considering the productivity goal through the identified lenses along each dimension. Since these questions are specific, it is possible to identify a set of metrics that may help to answer them, as shown in Table 5-1. Note that productivity metrics are always proxies for what you really want to measure, and there is a many-to-one relationship between metrics and a specific question, as well as between a set of specific questions and one or more productivity goals.

Productivity Goal 1: Improve Productivity at the Individual and Team Levels Through the Introduction of a New Continuous Integration System

Table 5-1. *Breaking Down Productivity Goal 1 Along the Three Dimensions*

Productivity Dimensions	Questions	Example Metrics
Quality	Is the committed code of a higher quality?	Test coverage. Number of bugs post release.
Velocity	Are developers able to deploy their features more quickly?	Time from creating a patch to patch release. Time to reach team milestones.
Satisfaction	Are developers more satisfied with the engineering process using the new tool?	Developer ratings for the new system. Developer ratings of team communication enabled by tool.

Example 2: Understanding How Meetings Impact Productivity

For this example, we consider a situation where the stakeholder wants to understand rather than try to improve productivity (although improving it may be a longer-term goal). The scenario we present here is the case where developers (the stakeholders) working in a team that also collaborates with other teams at their large company (the context) would like to *understand how meetings impact productivity* (the goal). Here the developers are more interested in an exploratory approach to understanding the impact of meetings on productivity. The dimensions and the lenses help form research questions, as shown in Table 5-2. In this example, even though no metrics have been defined, research questions can help sharpen an exploratory analysis by making it more concrete. Since the needs and goals of individual developers might conflict with those of the team and/or organization an exploratory analysis can help clarify such conflicts and form a basis for later change. Note that in the table we show only a sample of possible relevant questions along each dimension.

Productivity Goal 2: Develop an Understanding of How Meetings May Impact Productivity

Table 5-2. *Breaking Down Productivity Goal 2 Along the Three Dimensions*

Productivity Dimensions	Questions
Quality	Which meetings prompt follow-up work? Which meetings feel like a waste of time? Were all meeting participants needed in the meeting?
Velocity	What characterizes meetings that are the right length? What is the right length for meetings?
Satisfaction	What characterizes meetings where people feel good after attending?

Caveats

The framework we propose is abstract by its nature and thus may not suit all studies of productivity, nor may it match every nuanced definition of productivity. Other researchers and practitioners may want to consider additional dimensions or lenses depending on their needs. For example, learning/education could be considered as an explicit fourth dimension if this is important to the productivity goals under consideration.

When the dimensions framework is used with GQM, it may not be immediately evident to the researcher or practitioner what should be framed as a goal and what should be framed as one or more questions, as a goal could be stated as a research question or vice versa. As mentioned earlier, the HEART framework offers an alternative of using signals instead of questions. We have found it useful in practice to iteratively break down productivity measures along these three dimensions, and GQM is one approach for this.

As we noted earlier, any metrics defined are proxies for the concepts being measured. It is important to choose metrics that adequately capture key aspects of measured concepts and to be aware that every metric has limitations. We also stress that measuring engineer satisfaction is challenging, as satisfaction is influenced by and refers to many different concepts. The lenses together with the research goal may help in identifying how satisfaction should be conceptualized or measured. When it comes to satisfaction in particular, we stress there is no one-size-fits-all solution.

Finally, identifying/focusing on the right goals is outside the scope of this framework. A researcher or practitioner may assume the work being done is the right work when in fact it may not be (that is, the wrong tasks may be worked on in a productive manner!).

Key Ideas

Here are the key ideas from this chapter:

- Productivity should be considered along three dimensions: quality, velocity, and satisfaction.
- These three dimensions complement each other but often are in tension with each other.
- The dimensions have several possible attributes; measuring them is highly task and situation dependent.
- Productivity goals may be refined by considering the three dimensions through a set of perspective lenses.
- The main lenses we suggest include the stakeholders, the development context, the levels, and the time scale.

References

[1] Victor R. Basili, Gianluigi Caldiera, and H. D. Rombach. The Goal Question Metric Approach. In Encyclopedia of Software Engineering (John J. Marciniak, Ed.), John Wiley & Sons, Inc., 1994, Vol. 1, pp.528–532.

[2] V. R. Basili, G. Caldiera, and H. Dieter Rombach. The Goal Question Metric Approach. NASA GSFC Software Engineering Laboratory, 1994. (ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf)

[3] HEART framework for measuring UX. https://www.interaction-design.org/literature/article/google-s-heart-framework-for-measuring-ux