

FACTORS INFLUENCING SOFTWARE DEVELOPMENT PRODUCTIVITY

FINDINGS

- Software development productivity still depends on the capabilities of people and tools involved.
- The high importance of project constraints and project manager’s skills suggests project management as another key factor for project success.
- The internal (architecture, data) and external (interfaces) complexity of software is a significant determinant of development productivity.
- The studies presented in the literature provide evidence not only that factors influencing productivity vary across different application domains, but also that, in principle, the magnitude of productivity alone varies across different domains.
- Software complexity and programming language are clearly factors preferred in the context of project data repositories.
 - The impact of a programming language on development productivity is considered to be so large that some authors present average development productivities for certain languages, independent of other potential factor.
- The skills of software programmers and analysts, for instance, seem to play a more important role in enhancement/maintenance projects. Similarly, tool/method usage seems to be less important in new development projects. On the other hand, software development that is not a continuation of a previous product/release seems to significantly rely on the quality of project management and team motivation.
- Software complexity might have a positive impact on productivity in the context of new development (and thus is not perceived as a factor worth considering) and a negative impact in case of maintenance and enhancement.
- Software reuse is not as significant a productivity factor as commonly believed, although it can increase in productivity in both new development and software maintenance.

- Reuse is not for free and may, at least at the beginning, bring negative savings.
- Creating and maintaining rarely used, small repositories of small components tends to cost more than the reuse savings they generate.
- The complexity and quality of reusable assets are key success factors for software reuse.
- Process maturity is rarely identified directly as a factor influencing development productivity.
- Only a few studies selected the development life cycle model as a significant factor influencing software productivity.
- Both schedule over- and underestimation may have a negative impact on productivity.
- A schedule compression of 25% may lead to a 43% increase in development costs.
- Others show that schedule expansion also seems to have a negative impact on development productivity.
- The implementation of early quality assurance techniques such as reviews and inspections does not seem to have a deciding impact on development productivity. In only 6 out of 25 studies, this factor was stated explicitly as having a significant impact on productivity.
- While requirements volatility is considered to have a significant impact on development productivity, requirements management is not consistently regarded as having such an impact.
- Outsourced projects do not have significantly different productivity than in-source projects.
 - Outsourcing will only lead to significant cost savings when the outsourcing provider has access to significantly cheaper labor.
- Very mature organizations consider as very significant factors such factors as Clarity of development team roles and responsibilities, which are actually common for rather immature organizations.

Keywords:

Software productivity

Who is this briefing for?

Software engineers practitioners who want to make decisions about software development based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the systematic review conducted by Trendowicz and Munch

What is a systematic review?

cin.ufpe.br/eseg/slrs

What is included in this briefing?

The main findings of the original systematic review.

What is not included in this briefing?

Additional information not presented in the original systematic review.

Detailed descriptions about the studies analysed in the original systematic review.

To access other evidence briefings on software engineering:

cin.ufpe.br/eseg/briefings

For additional information about ESEG:

cin.ufpe.br/eseg