

# MANAGEMENT

**Oleksandr Bogolii, Postgraduate Student**

*"KROK" University*

*Kyiv, Ukraine*

*ORCID: <https://orcid.org/0000-0003-0253-667X>*

*DOI: <https://doi.org/10.30525/978-9934-26-495-5-26>*

## **THE ECONOMIC BENEFITS OF STANDARDIZING TEMPLATE FOR TECHNICAL DESIGN DOCUMENTS IN IT COMPANIES OPERATING IN A DISTRIBUTED ENVIRONMENT**

Today, software is frequently developed in globally distributed environments. Many organizations have started developing software remotely in order to save costs, gain access to skilled resources from other countries, and reduce the time to market by having round-the-clock development.

Documentation is an essential component of the software development process in such environment. Development teams, often spread across different time zones, rely on Technical Design Documents (TDD) to share knowledge and communicate strategies for implementing technical solutions.

A Technical Design Document is a document written by the development team that describes the solution to a specific technical problem with the aim of conveying the technical details of the upcoming work to team members.

Documenting Technical Design encourages engineers to proactively consider various issues they may encounter and to devise solutions for these challenges, as well as to analyze potential corner cases – scenarios that arise under atypical conditions. Additionally, TDD facilitates the early detection of issues in the proposed solution, at a stage when making necessary adjustments is still relatively easy.

Different development teams follow various standards and conventions for technical design templates, depending on their specific circumstances. Standardizing this document inside an organization has proven to be

helpful, enabling engineers to enhance their efficiency and improve the quality of their work [1].

The economic benefits of standardizing technical design documents arise from the optimization of human resource utilization and are attributed to the establishment of a consistent methodology for documenting technical solutions. This reduces the time spent on creating, reviewing, and correcting documents, thereby enhancing productivity and reducing the likelihood of errors resulting from inaccurate or incomplete documentation.

In natural units, the impact of standardizing the template of technical design documents is quantified in terms of time and is determined by calculating the difference in the duration required for software development before and after the adoption of the template.

In monetary units, the effect of standardizing TDD can be calculated as the total savings achieved at each phase of the software development process.

The software development life cycle is summarized in the international standard ISO/IEC/IEEE 12207 Systems and Software Engineering – Software Life Cycle Processes. This standard provides a theoretical foundation for the application of engineering approaches in software development.

The Software Development Life Cycle (SDLC) is a framework that outlines the different phases involved in developing software products. It includes comprehensive plans for each stage, encompassing the processes of building, deploying, and maintaining the software. It comprises phases – requirement gathering, technical design, coding, testing, deployment, and maintenance.

The economic benefits during the *technical design phase* are achieved by reducing the execution time through the standardization of documentation processes, the reusable nature of standard technical documentation, and the reduction of time required for design approval and agreement. Research indicates that in projects implementing standardized documentation practices, the time spent on analysis and documentation is typically reduced by 20–30% [2].

The economic benefits during the *coding* and *testing* phases arise from the reduction of costs associated with information retrieval by technical specialists and the decrease in the risk of errors related to inaccurate or incomplete documentation. A study by the Consortium for Information Software Quality (CISQ) revealed that poor documentation quality is

directly linked to an increased frequency of errors made by developers and the subsequent need for revisions [3].

When calculating the economic benefits during the *implementation* and *maintenance phase*, the reduction in the number of errors, along with the decrease in the time required to correct and refine identified defects in the software solution, are taken into account. The cost of correcting an error identified after the product's release is four to five times higher than the cost of fixing an error detected during the design phase, and it can be up to 100 times greater than the cost of addressing an error found during the maintenance phase [4]. Furthermore, standardized design documents significantly reduce the time required to locate necessary information during the maintenance phase. A study by Coveo [5] shows that 62% of IT workers spend 1 to 3 hours per day searching for the information they need. This highlights the significant effort involved in processing and finding relevant data essential for their work.

To summarize, standardizing the template of Technical Design Document in software development companies that operate in remote distributed environments brings significant economic benefits and leads to a reduction of a project's timespan.

### References:

1. Aghajani E. (2019) Software Documentation Issues Unveiled," in 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), Montreal, QC, Canada: IEEE, pp. 1199–1210. DOI: <https://doi.org/10.1109/ICSE.2019.00122>.
2. Krasner H. The cost of poor software quality in the us: a 2020 report. CISQ. Available at: <https://www.it-cisq.org/the-cost-of-poor-software-quality-in-the-us-a-2020-report/> (accessed: July 19, 2024.)
3. Charette R. N. (2005) Why software fails [software failure]. *IEEE Spectr*, vol. 42, no. 9, pp. 42–49. DOI: <https://doi.org/10.1109/MSPEC.2005.1502528>.
4. Dawson M., Burrell D., Rahim E., Brewster S. (2010) Integrating Software Assurance into the Software Development Life Cycle (SDLC). *Journal of Information Systems Technology and Planning*, vol. 3, pp. 49–53.
5. Workplace Relevance Report 2023. Coveo. Available at: [https://www.coveo.com/en/resources/reports/relevance-report-workplace?utm\\_source=press-release&utm\\_medium=organic&utm\\_campaign=relevance-report-2022-workplace](https://www.coveo.com/en/resources/reports/relevance-report-workplace?utm_source=press-release&utm_medium=organic&utm_campaign=relevance-report-2022-workplace) (accessed: July 19, 2024.)