

Artículo extraído de LinkedIn:



La crisis del software y sus problemas vigentes

Gerardo Fernández

Analista Programador • Ingeniería de Software • Desarrollo de Aplicaciones • Innovación Empresarial

Fecha de publicación: 6 mayo 2023

Desde la década de 1960, el desarrollo de software ha sido un desafío constante para la industria de la tecnología de la información. En aquel entonces, se hablaba de la "crisis del software", que se refería a la incapacidad de los desarrolladores de software para cumplir con los requisitos y plazos del proyecto, y la baja calidad del software resultante (Brooks, 1975).

A pesar de los avances tecnológicos y metodológicos en el desarrollo de software, estos problemas persisten hoy en día.

Según un informe de Geneca (2019), el 75% de los proyectos de software empresarial fallan debido a la falta de comprensión de los requisitos del proyecto. Además, solo el 26% de los proyectos de software se completan a tiempo y dentro del presupuesto (The Standish Group, 2020).

Uno de los principales problemas en el desarrollo de software es la mala planificación y estimación de tiempos del proyecto. Según una encuesta realizada por Deloitte (2018), el 38% de los encuestados citaron la mala planificación y la falta de control del proyecto como una de las principales razones de los problemas en proyectos de software empresarial. Esto a menudo se debe a la complejidad inherente del desarrollo de software y la falta de comprensión de los requisitos del proyecto.

Otro problema importante en el desarrollo de software es la baja productividad y calidad del software resultante. Según un estudio de CAST Software (2018), el 69% del código de software en aplicaciones empresariales contiene vulnerabilidades de seguridad. Además, según un estudio de Capgemini (2019), solo el 29% de las empresas creen que sus proyectos de software cumplen con los estándares de calidad.

A pesar de estos problemas, la industria del software ha hecho avances significativos en el desarrollo de nuevas metodologías y herramientas para abordar estos problemas. La adopción de enfoques ágiles, la ingeniería de software orientada a objetos y la automatización del desarrollo de software son solo algunos ejemplos de estos avances.

Sin embargo, es importante que la industria del software siga invirtiendo en la investigación y el desarrollo de nuevas tecnologías para mejorar el proceso de desarrollo de software. Los desarrolladores y clientes de software también deben exigir

una mayor calidad y productividad en el desarrollo de software para avanzar hacia soluciones más efectivas.

En resumen, la crisis del software y sus problemas vigentes siguen siendo un desafío para la industria de la tecnología de la información. Si bien ha habido avances significativos en el desarrollo de nuevas metodologías y herramientas, aún queda mucho por hacer para mejorar la planificación, estimación de tiempos, calidad y productividad del software resultante.

Referencias:

- Brooks, F. P. (1975). *The mythical man-month: Essays on software engineering*. Addison-Wesley Publishing Company.
- Capgemini. (2019). World quality report 2019-20. <https://www.capgemini.com/wp-content/uploads/2020/01/World-Quality-Report-2019-2020.pdf>
- CAST Software. (2018). CRASH report: Application security edition. <https://www.castsoftware.com/research-labs/crash-reports/2018/application-security-edition>
- Deloitte. (2018). Deloitte digital innovation survey 2018. <https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/technology-media-telecommunications/ch-en-tmt-digital-innovation-survey-2018.pdf>
- Geneca. (2019). The state of IT project management in 2019. <https://www.geneca.com/state-of-it-project-management/>
- The Standish Group. (2020). CHAOS report 2020: The 10th Standish Group report on project performance. https://www.standishgroup.com/sample_research_files/CHAOSReport2020.pdf