



“Lone Star Ruby Conference 2010

Real Software Engineering by Glenn Vanderburg”

Metodología de desarrollo

Facultad de ingenierías

Karen Jasbleidy Lopez Ruiz

Universidad Alexander Von Humboldt

Ingeniería de software

Armenia-Quindío

2024

La ineficiencia de la ingeniería de software

Glenn Vanderburgh habla sobre la ingeniería de software de manera crítica, cuestionando su eficacia y cómo se enseña y practica tradicionalmente. Según Vanderburgh, a pesar de ser llamado "ingeniería", este campo no cumple con los estándares de confiabilidad y eficacia que distinguen a otras ingenierías.

Vanderburgh destaca desde el principio que las técnicas enseñadas en las universidades y los programas de capacitación corporativa no son confiables. Estas técnicas, denominadas "ingeniería de software", no logran controlar los costos y generar software de alta calidad de manera constante. Muchas veces no logran producir software funcional, incluso cuando son practicadas rigurosamente por profesionales capacitados. Muchos programadores comparten esta idea de que la ingeniería de software no funciona bien, lo que indica que el problema es más profundo.

En otros campos de la ingeniería, las prácticas y técnicas que han demostrado funcionar de manera confiable a lo largo del tiempo, señala Vanderburgh. No obstante, en el ámbito del software, las técnicas conocidas como "ingeniería" no cumplen con esta condición. Esto provoca una discusión sobre si el desarrollo de software debe llamarse "ingeniería" o no. Vanderburgh propone que el software puede ser considerado un arte, un oficio e incluso una ciencia, pero no ve razones para que no también pueda ser una ingeniería. Él cree que redefinir el concepto de ingeniería de software es fundamental.

En 1968, la OTAN popularizó el término "ingeniería de software" en una conferencia. Durante ese tiempo, la industria del software se encontraba en medio de la "crisis del software", que consistía en proyectos inseguros, inestables, susceptibles a errores y que generaban costos excesivos. Aunque eran inteligentes y sabios, admitieron que sabían muy poco sobre cómo gestionar proyectos de software de manera efectiva. Aunque la conferencia tenía buenas intenciones y algunas observaciones sensatas, la implementación posterior de la ingeniería de software tomó un rumbo problemático.

La adopción del modelo de proceso definido, que requiere que cada etapa del trabajo se entienda completamente antes de comenzar, es uno de los problemas identificados por Vanderburgh. Debido a su aparente simplicidad, este modelo ha prevalecido, aunque no es adecuado para el desarrollo de software. La metodología en cascada, desarrollada por Winston Royce en 1970, es un ejemplo claro. A pesar de que Royce no recomendaba este modelo, su artículo fue malinterpretado y, a pesar de sus numerosas fallas, la cascada se convirtió en un estándar en la industria.

Otro punto crítico es el sesgo hacia el modelo de proceso definido, que indica que cada tarea debe estar completamente comprendida y registrada antes de avanzar a la siguiente fase, es otro punto importante. Este método, aunque bien intencionado, ignora el mundo real del desarrollo de software, donde los errores son inevitables. Berry Beam demostró en 1981 que el costo de corregir errores aumenta significativamente cuanto más avanzado está un proyecto en el ciclo de vida. Sin embargo, Vanderburgh sostiene que estos datos eran sesgados porque se basaban en proyectos que utilizaban el modelo en cascada.

El costo de corregir errores es significativamente menor en un enfoque más iterativo y ágil, donde el software funcional se entrega y prueba en ciclos cortos. Esto se debe a que los errores se detectan y corrigen rápidamente, lo que reduce significativamente el tiempo y el esfuerzo necesarios para solucionarlos.

Vanderburgh cuestiona lo que denomina "envidia matemática", en la que se ha intentado emplear modelos matemáticos precisos en el desarrollo de software. A pesar de que el diseño y la documentación son cruciales, un excesivo enfoque en las técnicas matemáticas y de modelado ha causado una desconexión entre la teoría y la realidad del desarrollo de software. La ingeniería de software requiere una combinación de innovación y adaptación constante, algo que los modelos rígidos no pueden proporcionar.

Para finalizar, la forma en que se enseña y practica la ingeniería de software necesita una revisión fundamental. Según Vanderburgh, el problema radica en una falta de comprensión de qué es realmente la ingeniería y cómo se debe aplicar al desarrollo de software. La ingeniería de software debe adoptar un enfoque más adaptable y flexible, basado en ciclos cortos de retroalimentación y corrección de errores, en lugar de seguir metodologías rígidas y procesos definidos. Solo así podrá aspirar a convertirse en una verdadera disciplina de ingeniería, produciendo software de alta calidad de manera confiable y eficiente.