12-04-2019

Resumen

Leyes Famosas de Arquitectura de Software

Maestro:

Eduardo Flores Gallegos

Alumno:

Carlos Romo Padilla

Materia:

TIC´S 4 semestre

12-04-2019

Ley de Arquitectura de desarrollo

En el mundo del desarrollo del software tiene algunas reglas, principios y leyes interesantes y famosas. Los programadores, desarrolladores, gerentes y arquitectos a menudo lo usan en conversaciones chats y entre otros casos. Pensaremos sin querer que nuestro interlocutor jamás a escuchado hablar d estos personajes (en el caso de si saber de alguno de los nombrados). Este tipo de principios, reglas son parte de, ya que son muy inspiradoras en el mundo del desarrollo, tanto interesantes, cómicas, divertidas vale la pena conocerlas y con su excelente historia de fondo para leer. Así que a continuación estarán presentes algunas de estas leyes.

Ley de murphy

Probablemente una de las leyes más famosas, sobre todo porque no solo es aplicable al desarrollo de software. **Si algo puede salir mal, lo hará.**

-Primera derivación: si funciona, es probable que no lo hayas escrito. **Segunda derivación:** la maldición es el único lenguaje que todos los programadores hablan con fluidez. A fin de cuentas una computadora hará lo que escribas, no lo que quieres. La programación defensiva, el control de versiones, los escenarios de fatalidad (para esos malditos ataques de zombies-servidor), TDD, MDD, etc. son todas buenas prácticas para defenderse contra esta ley.

Ley de Brook

La mayoría de los desarrolladores, ya sea a sabiendas o sin saberlo, tendrán experiencia con la ley de Brook, que establece: **La adición de mano de obra a un proyecto de software tardío lo hace más tarde.** Si un proyecto se está retrasando, simplemente agregar mano de obra probablemente tendrá resultados desastrosos. Buscar y revisar el nivel de eficiencia de la programación, la metodología del software, la arquitectura técnica, etc. casi siempre tendrá mejores resultados. O no, lo que probablemente significa que la Ley de Hofstadter también está tomada en cuenta.

12-04-2019

Ley de Hofstadter

La ley de Hofstadter fue escrita por Douglas Hofstadter y lleva su nombre

Esta ley es, por supuesto, no debe confundirse con Leonard Hofstadter de la teoría del Big Bang. A pesar de que su cita tendrá sentido para algunos de ustedes.

La regla establece: **Siempre lleva más tiempo de lo que espera, incluso si tiene en cuenta la Ley de Hofstadter.** La "ley" es una declaración sobre la dificultad de estimar con precisión el tiempo que tomará para completar tareas de complejidad sustancial. La naturaleza recursiva de la ley es un reflejo de la dificultad ampliamente experimentada de estimar tareas complejas a pesar de todos los esfuerzos, incluso saber que la tarea es compleja. Es por eso que siempre debe tener un búfer antes de dar cualquier tipo de estimación. Si desea saber más sobre cómo proporcionar mejores estimaciones, lea mi publicación sobre el tema: Asistente de estimación.

Principio de Pareto también conocido como la regla 80-20

Para muchos fenómenos, el 80% de las consecuencias provienen del 20% de las causas.

Este es el principio detrás de la dolorosa verdad de que el 80% de los errores en el código surgen del 20% del código. De lo contrario, el 80% del trabajo realizado en una empresa lo realiza el 20% del personal. El problema es que no siempre tienes una idea clara de cuál es el 20%. Así que eso dificulta un poco al momento de hacer las cosas.

Tomando en cuenta algunas de las leyes de arquitectura de desarrollo de software se puede dar una idea clara que tan difícil eh interesante es el desarrollo de software, y como es que uno puede estimar o por lo menos mantiene un hilo de como comenzar de una manera estructurada cuando desarrolla el software ya sea propio o para alguien en específico ya sea una empresa o alguna persona que haya requerido de nuestra disposición para un software propio (hablando de la persona que no lo encargo).