

11-4-2019

INGENIERIA DE SOFTWARE

Leyes famosas del desarrollo
de software

MAESTRO: Eduardo Flores
Gallegos

ALUMNA: Sierra Lynn Brown



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga

ATEC

Como cualquier otro campo, el mundo del desarrollo de software tiene algunas reglas, principios y leyes interesantes y famosas. Los programadores, desarrolladores, gerentes y arquitectos a menudo los usan en conversaciones, reuniones.

Estas leyes consisten en reglas, principios o palabras famosas de personas grandes e inspiradoras en el mundo del desarrollo.

Estas son las leyes más famosas y más utilizadas en el desarrollo de software:

Ley de Murphy

Probablemente una de las leyes más famosas, sobre todo porque no solo es aplicable al desarrollo de software.

- 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.
- Conclusión: una computadora hará lo que escribas, no lo que quieres.

Ley de Brook

La adición de mano de obra a un proyecto de software tardío lo hace más tarde.

Si un proyecto se está retrasando, es mejor no incluir más personas ya que se atrasaría más el proyecto porque no tienen conocimiento sobre el y sería una pérdida de tiempo tener que regresarse a explicarles paso a paso a como desarrollarlo.

Ley de Hofstadter

Siempre lleva más tiempo de lo que espera, incluso si tiene en cuenta la Ley de Hofstadte.

Es una declaración sobre la dificultad de estimar con precisión el tiempo que tomará completar tareas de una manera más compleja.

Ley de Conway

Cualquier pieza de software refleja la estructura organizativa que lo produjo.

Las organizaciones cuyos sistemas de diseño están obligados a producir diseños que sean copias de las estructuras de comunicación de estas organizaciones.

El principio de robustez de la ley de Postel aka

Sea conservador en lo que envía, sea liberal en lo que acepta.

Principio de Pareto también conocido como la regla 80-20

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

El código te demuestra que no lo has desarrollado bien al momento de querer corregir el error, puedes causar más errores.

El principio de Peter

Una ley bastante deprimente y, a veces, frustrante, ciertamente si tiene experiencia de primera mano.

En una jerarquía, cada empleado tiende a elevarse a su nivel de incompetencia y decide hasta dónde quiere llegar.

Principio de Kerckhoff

En la criptografía, un sistema debe ser seguro incluso si todo lo relacionado con el sistema, excepto una pequeña parte de la información, la clave, es de conocimiento público.

La ley de linus

Dados suficientes globos oculares, todos los errores son superficiales.

- El modelo donde el código fuente está disponible con cada versión de software, pero el código desarrollado entre versiones está restringido a un grupo exclusivo de desarrolladores de software.
- El modelo Bazaar, en el que el código se desarrolla a través de Internet en vista del público.

Ley de Moore

El poder de las computadoras por unidad de costo se duplica cada 24 meses.

La versión más popular dice:

El número de transistores en un circuito integrado se duplicará cada 18 meses

La velocidad de procesamiento de las computadoras se duplica cada 2 años.

Ley de wirth

El software se ralentiza más rápido que el hardware.

Regla del noventa y noventa

El primer 90% del código toma el 10% del tiempo. El 10% restante toma el otro 90% del tiempo.

Principio de optimización de Knuth

La optimización prematura es la fuente de todos los males.

Primero escribe el código, luego identifica los problemas y luego soluciona los problemas.

La ley de norvig

Cualquier tecnología que supere el 50% de penetración nunca volverá a duplicarse (en cualquier número de meses).

Referencia:

Tim Sommer. (2017). Famous Laws of Software Development. 11/04/2019, de Inspired Web Development Sitio Web:
<https://www.timsommer.be/famous-laws-of-software-development/>