Some Laws(?) of Software Engineering

KochiTechgroup

Courtesy

The 13 software engineering laws - by Anton Zaides (manager.dev)

13 laws of Software Engineering

1. Parkinson’s law
2. Hofstadter’s law
3. Brooks’ law
4. Conway’s law (and the Inverse Conway's law)
5. Cunningham’s law
6. Sturgeon’s law
7. Zawinski’s law
8. Hyrum’s law
9. Price’s law
10. The Ringelmann effect
11. Goodhart’s law
12. Gilb’s law
13. Murphy’s law

(Cyril Northcote) Parkinson’s law

- *Work expands to fill the available time (for it’s completion)*

(Douglas) Hofstadter’s law

- *It always takes longer than you expect, even when you take into account Hofstadter’s Law.*

(Fred) Brook’s law

- *Adding manpower to a late software project makes it later.*

(Melvin) Conway’s law

- *Organizations produce designs which are copies of the communication structures of these organizations.*

(Ward?) Cunningham's law

- *The best way to get the right answer on the internet is not to ask a question, but to post the wrong answer.*

(Theodore) Sturgeon’s law

- *90% of everything is crap.*

(James | Jamie?) Zawinski’s Law

- *Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones that can.*

Hyrum(Wright)’s law

- *With a sufficient number of users of an API, it does not matter what you promise in the contract: all observable behaviors of your system will be depended on by somebody.*

Price's law

- *In any group, 50% of the work is done by the square root number of people.*

Ringelmann effect

- *The tendency for individual members of a group to become increasingly less productive as the size of their group increases.*

Goodhart’s law

- *When a measure becomes a target, it ceases to be a good measure.*

Glib’s law

- *Anything you need to quantify can be measured in some way that is superior to not measuring it at all.*

Murphy’s Law

- *Anything that can go wrong will go wrong.*

Questions

- If,any