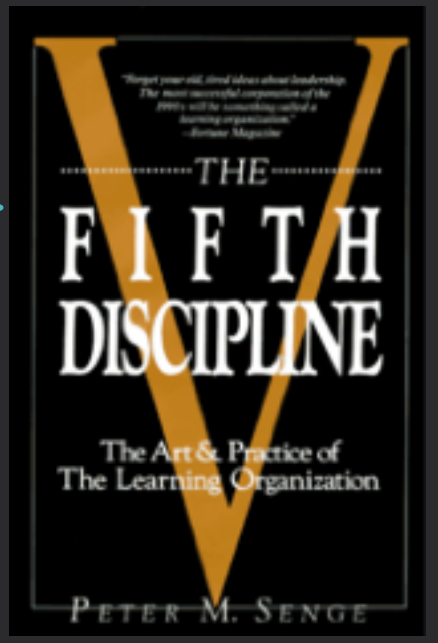
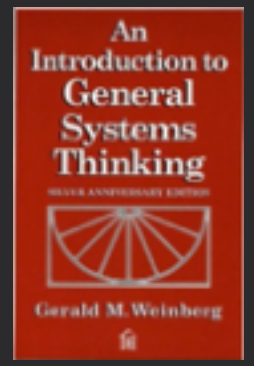
Laws of Systems Thinking

1. Today's problems come from yesterday's 'solutions.'
2. The harder you push, the harder the system pushes back.
3. Behavior will grow worse before it grows better.
4. The easy way out usually leads back in.
5. The cure can be worse than the disease.
6. Faster is slower.
7. Cause and effect are not closely related in time and space.
8. Small changes can produce big results…but the areas of highest leverage are often the least obvious.
9. You can have your cake and eat it too—but not all at once.
10. Dividing an elephant in half does not produce two small elephants.
11. There is no blame.





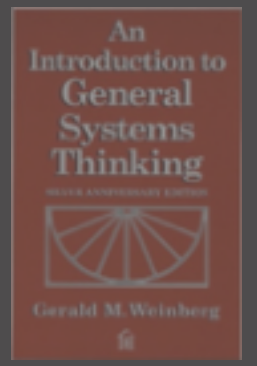
Weinberg-Brooks' Law: More software projects have gone awry from management's taking action based on ***incorrect system models*** than for all other causes combined.

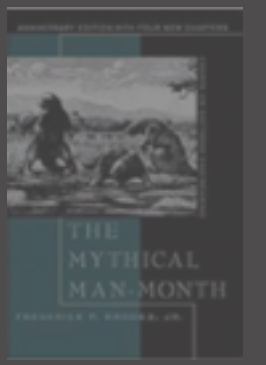
Causation Fallacy: Every effect has a cause… and we can tell which is which

LEVEL UP

Laws of Systems Thinking

1. Today's problems come from yesterday's 'solutions.'
2. The harder you push, the harder the system pushes back.
3. Behavior will grow worse bef
4. The easy way out usually lea
5. The cure can be worse than t
6. Faster is slower.
7. Cause and effect are not clos space.
8. Small changes can produce b highest leverage are often th
9. You can have your cake and e once.
10. Dividing an elephant in half elephants.
11. There is no blame.





What are we applying this week?

- Whole feature team
- Prioritize learning
- Avoid local optimization - *watch the baton, not the runner*
- Understand system forces and how they impact you (this is not about developer productivity)

ore software
om
n based on
an for all other

Causation Fallacy: Every effect has a cause… and we can tell which is which

LEVEL UP