

# **If not Scrum, then what?**

**Five laws of software  
development with  
process suggestions**



**GOOD  
NEWS  
IS COMING**



# Building Software is Unpredictable

- Pick a release cadence, like once a month, and ship whatever you have.
- Pick a set of features and ship whenever it is finished.
- Do not rely too heavily on long term road-maps, as they are likely to change.
- Do not commit programmers to complete features by a certain date.

# Process is not free

It will affect productivity and morale – often dramatically

- Resist the desire to track everything, it's intrusive.
- Allow individual programmers to track their tasks with whatever tool they prefer. Post-It notes, pen and paper, or software - whatever makes them productive and does not get in their way.
- Programmers thrive in a flat organizational structure.

# Software is never “done”

**Done** is a decision, not a definition

- Do not attempt to collect metrics or enforce standards to automate this decision.
- Trust the intuition of your programmers in this matter. Pride and prestige will drive the product to a sufficient quality level without interference.

# **Programmers need long periods of uninterrupted concentration**

- Ideally, programmers will work remotely or have private offices, or both.
- Collaboration should be asynchronous and online whenever possible.
- Avoid open office layouts like the plague.

# **Coding is more like design than manufacturing**

- Assign areas of responsibility. Don't waste time estimating, planning or designing tasks that lie within someone's area of responsibility.
- If more than one person works inside an area of responsibility, assign one to be in charge. Others contribute by submitting suggested changes that are reject or accept based on value. This model used in open source projects.

# References

**If not Scrum, what then?**

**Why Scrum is the Wrong Way to Build Software**

**Three Ways Agile has Gone Astray**