

Agile manifesto:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Some of the 12 agile principles:

- Working software is the primary measure of progress.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
 - Scrum - Sprints
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
 - Make wait time visible
 - Wait time = busy time / idle time e.g. 50%/50% = 1hr wait time; 90%/10% 9hrs of wait time
 - Everyone needs idle time otherwise work gets stuck in queues.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.
 - Sensei Mike Rother – It doesn't matter what you improve, as long as you're improving something. Because if you're not improving, entropy guarantees that you are getting worse.
 - Habits – Repetition creates habits and habits are what enables mastery.
 - Continually put tension into the system so we're continually reinforcing habits.
 - Resilience engineering tells us that we should routinely inject faults into the system doing them frequently, to make them less painful.

Others:

Technical debt:

- A concept in SD that reflects the implied cost of additional rework caused by choosing an easy(limited) solution now instead of using a better approach that would take longer.
- Unaddressed technical debt increases software entropy

Context switching /Multi-tasking

- Ensure the fast, predictable, and uninterrupted flow of planned work that delivers value to the business while minimising the impact and disruption of unplanned work.

Context Switching			
# of Tasks	Time Available	Time Lost to Context Switching	Time Available in a 10 HR Day
1	100%	0%	10
2	80%	20%	8
3	60%	40%	6
4	40%	60%	4
5	20%	80%	2

References:

The Phoenix Project

Scrum: Doing twice the work in half the time

The Culture Code