

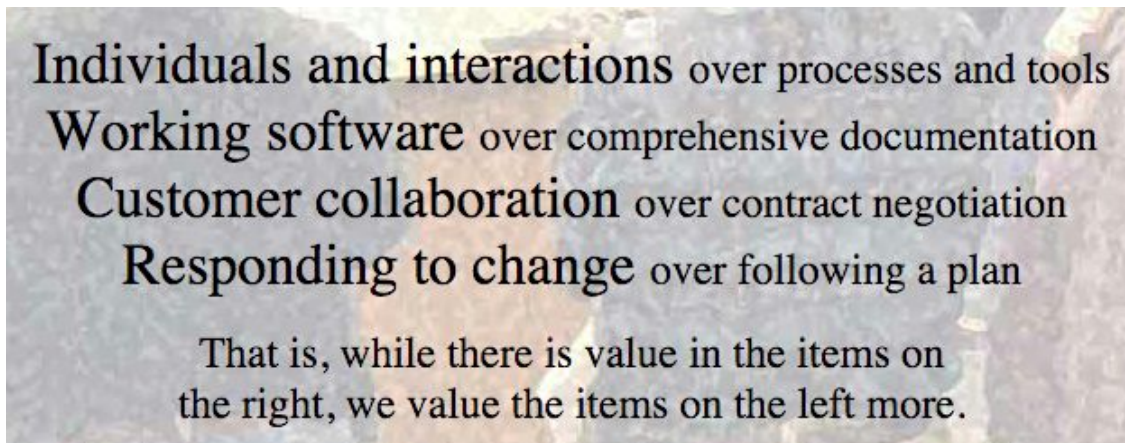
# tech resources - agile

*Tech notes describing an agile development methodology and Scrum.*

## Agile and Scrum

*Agile software development is a set of principles for software development as defined by the Agile Manifesto. Scrum is one example of an agile methodology.*

Agile is a development process defined by its manifesto, [www.agilemanifesto.org/](http://www.agilemanifesto.org/):



[www.agilemanifesto.org/principles.html](http://www.agilemanifesto.org/principles.html) - there are also 12 lower-level agile principles; these are worthy!

[www.amzn.com/B007P5N8D4](http://www.amzn.com/B007P5N8D4) - "Scrum: a Breathtakingly Brief and Agile Introduction" by Chris Sims and Hillary Louise Johnson; I have it (\$1 Kindle version); it's excellent and only 54 pages, as it should be!

[www.amzn.com/B015QA141W](http://www.amzn.com/B015QA141W) - "Scrum for Newbies" by Jeremy Wilson; I have it (Kindle \$3), but haven't read it yet

Other book possibilities:

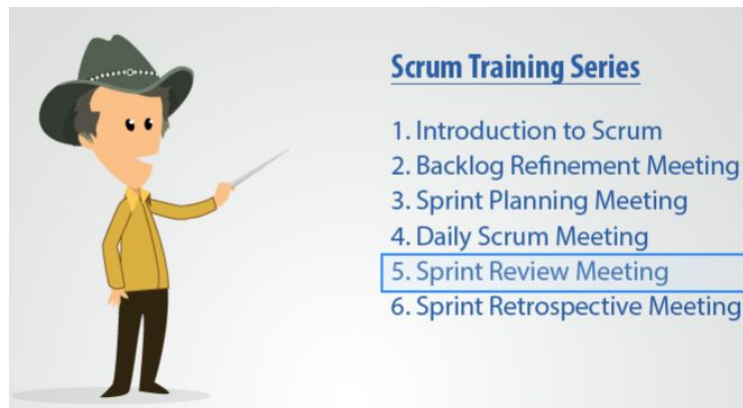
- [www.amzn.com/B008NAKA5O](http://www.amzn.com/B008NAKA5O) - "Essential Scrum" by K Rubin; looks good and all, but why is it 500 pages to describe an agile process?
- [www.amzn.com/B00JI54HCU](http://www.amzn.com/B00JI54HCU) - Scrum: The Art of Doing Twice the Work in Half the Time by Sutherland, 2014. Only 258 pages, but dinged in reviews for not actually covering Scrum?!?

[www.scrumguides.org](http://www.scrumguides.org) - the “official” definition of Scrum

[www.scrummethodology.com/](http://www.scrummethodology.com/) - nice overview; watch these videos when I get a chance

[www.scrumtrainingseries.com/](http://www.scrumtrainingseries.com/) - a good video series; watched these and here are my notes:

1. **Introduction to Scrum** - Strong intro! “Today is the dumbest day of the rest of your project”, desired team size is (7 +/- 2), “a PowerPoint presentation about hypothetical things”
2. **Backlog Refinement Meeting** - INVEST= {Independent, Negotiable, Valuable, Estimable, Small, Testable}, timeboxed meetings, “potentially shippable form”, user story/use case, very funny!
3. **Sprint Planning Meeting** - 2 hours, plan tasks for 2 week sprint
4. **Daily Scrum Meeting** - No phones, standing up, ideal sprint task is 1 man-day or less
5. **Sprint Review Meeting** - agenda={demo, product owner declares what’s done, measure velocity (optional), stakeholder feedback}, no “partially done” tasks
6. **Sprint Retrospective Meeting** - team growth, “a learning team”,



[www.scrumreferencecard.com/](http://www.scrumreferencecard.com/) - 6 page reference card? Not good

[www.agilenutshell.com/](http://www.agilenutshell.com/) - this guy has a book: The Agile Samurai,

[www.amzn.com/1934356581](http://www.amzn.com/1934356581); I have it but haven’t read it yet

[en.wikipedia.org/wiki/Pair\\_programming](http://en.wikipedia.org/wiki/Pair_programming) - Pair programming can be a part of Agile, “Pairs spend about 15% more time on programs than individuals. However, the resulting code has about 15% fewer defects.” and “survey: 95% of programmers enjoyed working in pairs, 95% were more confident in their solutions”; So, the question is: try something new/fun/different because that’s what school is for? Let teams decide?