

# Agile Software Development

## Customer Focus

Write user stories to capture customer's need for a feature. Have customer involvement if at all possible. Write acceptance tests. Regularly demo and/or release working software to customers. Have frequent releases.

## Software Focus

Demo working software after each sprint. Write tests along with code and use automated testing (TDD). Use some sort of Continuous Integration platform. Never skip refactoring, do not allow technical debt to overwhelm the project.

## Team Focus

Enable team collaboration and communication. Use as short of sprints as reasonable to keep team effective. Avoid interference---do not modify sprint backlog. Allow team to work at a sustainable pace.

## No Wasted Time Focus

Keep estimation and planning simple. Use unitless story point estimates from the Fibonacci series. Use planning poker to avoid bias. Avoid detailed up-front design but do not skip refactoring. Track project velocity. Keep sprints short. Use a burndown chart to track progress towards a release.

## Value Focus

Welcome changing requirements to keep your product relevant. Use test-driven development to help create reliable and effective software. Use static analysis tools.

Copyright 2024 Jonathan Cook. This is a work in progress for good agile development ideas.



**BE BOLD.** Shape the Future.

**New Mexico State University**  
**[computerscience.nmsu.edu](http://computerscience.nmsu.edu)**