

Manifesto for Agile Software Development

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.

Principles behind the Agile Manifesto:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

www.agilemanifesto.org

Lean Product Development

Sustainable shortest lead time, best quality and value (to people and society), most customer delight, lowest cost, high morale, safety

Respect for People	Product Development	Continuous Improvement
don't trouble your customer	long-term great engineers	Go See
develop people, then build products	mentoring from manager, engineer, teacher	kaizen
no wasteful work	cadence	spread knowledge
teams & individuals evolve their own practices and improvements	cross-functional	small, relentless
build partners with stable relationships, trust, and coaching in lean thinking	team room and visual management	retrospectives
develop teams	entrepreneurial chief engineer/product manager	5 Whys
	set-based concurrent development	eyes for waste
	create more knowledge	perfection challenge
		work toward flow
	14 Principles	
	long-term	
	flow	
	pull	
	less variability & overburden	
	Stop & Fix	
	master norms	
	simple visual management	
	good tech	
	leader-teachers from within	
	develop exceptional people	
	help partners be lean	
	Go See	
	consensus	
	reflection & kaizen	

Management applies and teaches lean thinking, and bases decisions on this long-term philosophy

www.leanprimer.com

www.chmdebeer.com/resources