

Comparing Traditional, Agile, and Open Source Development

	<i>Traditional</i>	<i>Agile</i>	<i>Open source</i>
<i>Documentation</i>	Documentation is emphasized as a means of quality control and as a management tool.	Documentation is deemphasized.	All development artifacts are globally available, including code and information documentation.
<i>Requirements</i>	Business analysts translate users' needs into software requirements.	Users are part of the team.	The developers typically are the users.
<i>Staffing model</i>	Developers are assigned to a single project.	Developers are assigned to a single project.	Developers typically work on multiple projects at different levels of involvement.
<i>Peer review</i>	Peer review is widely accepted but rarely practiced.	Pair programming institutionalizes some peer review.	Peer review is a necessity and is practiced almost universally.
<i>Release schedules</i>	Large number of requirements bundled into fewer, infrequent releases.	Release early, release often.	Hierarchy of release types: "nightly," "development," and "stable."
<i>Management</i>	Teams are managed from above.	Teams are self-organized.	Individual contributors set their own paths.
<i>Testing</i>	Testing is handled	Testing is part of	Testing and QA

Comparing Traditional, Agile, and Open Source Development

	by QA staff, following development activities.	development.	can be performed by all developers.
Distribution of work	Different parts of the codebase are assigned to different people.	Anyone can modify any part of the codebase.	Anyone can modify any part of the codebase, but only committers can make changes official.

www.YouSigma.com