

## Comparison of Agile Methods by The Agile PrepCast

This document contains a comparison of the eight most popular Agile methods. For each method we show how it stacks up against ten criteria in a comparison table.

The goal is to help those who are preparing for their PMI Agile Certified Practitioner ([PMI-ACP](#))® Exam by providing a much needed “quick reference”. Use it as a cheat sheet in your exam preparation.

The eight methods were selected because they currently represent the most popular Agile methods used by practitioners around the world. Questions on the exam will most likely be focused on this group. And the ten criteria were selected because they represent the major differences between the methods. Knowing the differences will help you in answering exam questions. You can find a description and definition for each method and each criteria starting on page 5.

But this comparison can be used for other purposes as well. For example if you are chartering a new Agile project and you are not certain which method to use, then you can use the comparison table to get a high-level overview of the methods and use it as one of your methods selection criteria.

### What is The Agile PrepCast?

The Agile PrepCast™ is a PMI Agile Certified Practitioner (PMI-ACP)® Prep Video Workshop that you download to your smart phone, tablet or laptop computer and watch whenever and wherever you want: On your commute to and from work, while doing housework, in the gym, during your lunch-break or back home as you relax on your sofa.

**Watch the “Scrum Overview” lesson for free:**  
[www.youtube.com/watch?v=sROBERoQoXk&](http://www.youtube.com/watch?v=sROBERoQoXk&)

The course is your high-quality but low-cost alternative to attending a cumbersome classroom training. This is a complete PMI-ACP exam study approach that increases your chance of passing the Exam. Shorten your study time and still keep your focus. And you don't even have to read the recommended books to prepare. Watch over 50 hours of video from The Agile PrepCast and tackle the exam. It's that Agile!

Click the banner to learn more:



Please visit The Agile PrepCast at [www.agileprepcast.com](http://www.agileprepcast.com)

## Methods Comparison

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Scalability</b>	A wide range of software projects of various sizes; can also be adapted to non-software projects such as product/application development products.	Software development projects driven by need for quick delivery; also for large, complex projects where plan driven methods are ineffective.	Any technical or business environment where a system needs to be quickly developed and deployed; preferably any project that is a good fit with the DSDM Atern Method as outlined in the Project Approach Questionnaire.	Designed and first used for large, complex banking projects; good for any large, complex project with large teams.	Designed for project environments that are high-speed, high-change, and high-uncertainty; also good for organizations that are highly adaptive.	Based on Lean Manufacturing; used when project requires the use of a specific set of engineering practices or when cost and ROI are the main measurements of success.	Although Kanban is not considered an Agile development method, it can be used with any existing processes and projects to help reduce costs and increase efficiency.	Designed to be adaptable to projects ranging from small with low criticality to large with high criticality.
<b>Team size</b>	Small Teams (Should consist of at least 5 but no more than 9 team members)	Small teams (Generally 2-10 team members; emphasis is on highly-skilled and experienced team members)	Small team: (Generally 2-10 team members in a structured and disciplined environment)	Small to Large Teams (Generally 4-20 team members depending on complexity and length of project)	Small to Large Teams (Depends on project scope)	Not specified;	Not specified; (Normally used with existing team size)	Supports Any Team Size (Emphasis is on highly-skilled and experienced team members)
<b>Iteration length</b>	Medium (30-day iterations recommended as maximum length with preference for a shorter iterations)	Short (2-week iterations; never to exceed 3 weeks)	Not specified	Short (2-week iterations recommended as best practice)	Determined by project schedule and degree of uncertainty  (4-week iterations for small projects and 8-week iterations for medium to large projects)	Not specified	Short (1-week iterations)	Project Specific (Up to 4-month iterations for large, complex projects)

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Roles &amp; responsibilities</b>	Specifically defined	Specifically defined	Specifically defined	Specifically defined	Not defined	Not defined	Not specified; (normally uses existing organization and project roles)	Not defined
<b>Process centric vs. people centric</b>	People Centric (Process is important but secondary)	People centric	Process centric	Process centric	People centric	Process centric	Process centric	People Centric (Process is important but secondary)
<b>Virtual team support</b>	Somewhat Supported (not specifically designed for distributed teams but multiple teams could easily be effectively distributed)	Not Supported (Team must be co-located distributed teams.)	Not Supported	Somewhat Supported (Designed for multiple teams; should be adaptable to distributed teams.)	Loosely Supported (Can be applied to distributed teams but becomes more challenging during collaboration phase.)	Not Supported	Somewhat Supported (Can be adapted to distributed teams through virtual Kanban boards and other Agile tools.)	Fully Supported (Designed to support distributed teams.)
<b>Risk mitigation level</b>	High Risk Mitigation (Risk is identified and mitigated daily.)	Medium Risk Mitigation (Despite the fact that risks are decreased by Collocation & Pair Programming, risks are also increased by the slower delivery of features)	High Risk Mitigation (Risk mitigation is inherent in most of the principles and the roles and responsibilities that are basic to this method.)	Medium Risk Mitigation (Follows most basic Agile principles except for daily collaboration with customer.)	High Risk Mitigation (Risk-Driven is one of the six basic characteristics and each Component Cycle Plan is initiated by analyzing risks.)	Medium Risk Mitigation (Follows most basic Agile principles except for making decisions as late as possible, which increases negative risk.)	Medium Risk Mitigation (Follows most basic Agile principles except often considers estimation as waste, which may increase negative risk.)	High Risk Mitigation (Risk mitigation is inherent in just about all of the principles and techniques that are recommended when using this method.)
<b>Customer interaction</b>	Medium Interaction (monthly input)	High Interaction (daily input)	High Interaction (daily to weekly input)	Low interaction (input not specified)	Low interaction (input not specified)	Low Interaction (input not specified)	Low Interaction (input not specified)	High interaction (daily to weekly input)

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Pros</b>	Maximizes personal communication and informal knowledge sharing; breaks project into manageable pieces; progress is made even if requirements are not stable	Simple; iterative; values communication; based on best practices; puts emphasis on design	Highly dependable; provides well-defined guidelines for the business system, concept development, and requirements	Strong modeling features; provides detailed guidelines for multi-team projects	Very strong in non-technical aspects of software development	Seeks to change companies from the top down; guidelines for business enterprise very defined	Allows teams to visualize their work and eliminate bottlenecks; can lead to exponential improvements in operational efficiency and quality	Strong on communication; well-defined guidelines for project teams; well-defined technical practices
<b>Cons</b>	Weak on measurement practices; weak on business system, technical, and concept development practices	Lacks design documentation; highly prescriptive; lacks measurement processes; does not address deployment	Details and white papers available only to DSDM Consortium members; may tend towards bureaucracy	Only addresses design and implementation; requires highly-experienced experts for modeling	Does not provide guidelines for individual development projects; does not address technical aspects; weak on metrics	Allows little change in requirements; literature for applying in a software environment is limited; poorly-defined technical and measurement practices	Using JIT delivery will inevitably lead to delays at some point when JIT turns in to "Not In Time"; doesn't capture or show dependent tasks very well; hard to see overall project status	Largely theoretical; does not define guidelines for the business enterprise

## Methods Description

Method	Description
<b>Scrum</b>	Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development. Its focus is on "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach". Scrum enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication between all team members and disciplines in the project. A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an empirical approach - accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly and respond to emerging requirements.
<b>XP</b>	Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. Other elements of Extreme Programming include: programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels
<b>DSDM Atern</b>	Dynamic systems development method (DSDM) is an agile project delivery framework, primarily used as a software development method. First released in 1994, DSDM originally sought to provide some discipline to the rapid application development (RAD) method. In 2007 DSDM became a generic approach to project management and solution delivery. DSDM is an iterative and incremental approach that embraces principles of Agile development, including continuous user/customer involvement. DSDM fixes cost, quality and time at the outset and uses the MoSCoW prioritization of scope into musts, shoulds, coulds and won't haves to adjust the project deliverable to meet the stated time constraint. DSDM is one of a number of Agile methods for developing software and non-IT solutions, and it forms a part of the Agile Alliance.
<b>FDD</b>	Feature-driven development (FDD) is an iterative and incremental software development process. It is one of a number of Agile methods for developing software and forms part of the Agile Alliance. FDD blends a number of industry-recognized best practices into a cohesive whole. These practices are all driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner.

Method	Description
<b>ASD</b>	Adaptive Software Development (ASD) is a software development process that grew out of rapid application development work by Jim Highsmith and Sam Bayer. It embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs. Adaptive Software Development replaces the traditional waterfall cycle with a repeating series of speculate, collaborate, and learn cycles. This dynamic cycle provides for continuous learning and adaptation to the emergent state of the project. The characteristics of an ASD life cycle are that it is mission focused, feature based, iterative, timeboxed, risk driven, and change tolerant.
<b>LSD</b>	Lean software development - is a movement dedicated to reducing errors and wasted time while maximizing education and efficiency. Its principles were originally used in IT and manufacturing, and it has since been adopted by the programming community. The guiding philosophy of lean development is a primary commitment to the value being created for the end user while intelligently conserving resources. It has large support among the Agile development community, which it has much in common with.
<b>Kanban</b>	Kanban (かんばん(看板)) (literally signboard or billboard) is a scheduling system for lean and just-in-time (JIT) production. Kanban is a system to control the logistical chain from a production point of view, and is not an inventory control system. Kanban was developed by Taiichi Ohno, at Toyota, to find a system to improve and maintain a high level of production. Kanban is one method through which JIT is achieved. Kanban became an effective tool in support of running a production system as a whole, and it proved to be an excellent way for promoting improvement. Problem areas were highlighted by reducing the number of kanban in circulation.
<b>Crystal family</b>	Crystal Methods - is a collection of Agile software development approaches, focuses primarily on people and the interaction among them while they work on a software development project. There is also a focus on business-criticality and business-priority of the system under development. Unlike traditional development methods, Crystal doesn't fix the tools and techniques of development, but keeps people and processes at the core of the development process. However, it is not only the people or the processes that are important, rather the interaction between the two that is most important.

## Criteria Definition and Description

Criteria	Definition / Description
<b>Scalability</b>	This is the degree to which this Agile method can be used to support different types of product development and, different types of projects, or different types of industries.
<b>Team size</b>	This is the ideal number of members on a development team required for maximum effectiveness of this Agile method.
<b>Iteration length</b>	This is the recommended length of time (iteration) to release a product increment to the customer for this Agile method.
<b>Roles &amp; responsibilities</b>	This is the degree to which specific roles & responsibilities are defined for this Agile method. (Specifically Defined/Somewhat Defined/Loosely Defined/Not Defined)
<b>Process centric vs. people centric</b>	This refers to whether or not this Agile method is more process centric, more people centric, or a combination of both.
<b>Virtual team support</b>	This is the degree to which this Agile method supports virtual team communication and coordination. (Fully Supported/Somewhat Supported/Loosely Supported/Not Supported)
<b>Risk mitigation level</b>	This is the level of Organic Risk Mitigation inherent on a project simply by implementing this Agile method. (High Risk Mitigation/Medium Risk Mitigation/Low Risk Mitigation)
<b>Customer interaction</b>	This is the level of customer interaction (High Interaction/Medium Interaction/Low Interaction) and the recommended frequency of customer interaction (Daily/Weekly/Monthly) required for maximum effectiveness of this Agile method.
<b>Pros</b>	This is the most appealing aspects of this Agile method. Organizations can use this information to determine those Agile methods that should be implemented on their specific projects..
<b>Cons</b>	This is the least appealing aspects of this Agile method. Organizations can use this information to determine those Agile methods that should not be implemented on their specific projects..

## Sources

### Comparison Table

- A Survey of Agile Methodologies, Mkandla and Dwolatzky, The Transactions of the SA Institute of Electrical Engineers, December, 2004
- OSP International LLC - The Agile Prepcast - [www.agileprepcast.com](http://www.agileprepcast.com)
- <http://www.boost.co.nz/blog/development/scrup-and-kanban-a-developers-perspective/>

### Methods Description

- <http://blog.airbrake.io/internet/lean-forward-how-lean-software-development-is-shaping-our-virtual-world/>
- <http://codebetter.com/darrellnorton/2005/02/02/lean-software-development-overview/>
- <http://scrumstudy.com/blog/?p=223>
- <http://www.wikipedia.org>



---

## Copyright and Disclaimer

Published by:

OSP International LLC, P.O. Box 863, USA - Silverado, CA 92676

Contact: [info@osp-international.com](mailto:info@osp-international.com) - [www.osp-international.com](http://www.osp-international.com)

OSP International LLC has been reviewed and approved as a provider of project management training by the Project Management Institute (PMI)®. As a PMI Registered Education Provider (R.E.P.), OSP International LLC has agreed to abide by PMI established quality assurance criteria.

Copyright © 2013 OSP International LLC. All rights reserved.

The Agile PrepCast, The PM PrepCast, The PM Podcast, The Project Management Podcast, The PDU Podcast and The PM Exam Simulator are trademarks of OSP International LLC.

PMI, PMP, CAPM, PgMP, PMI-ACP, PMI-SP, PMI-RMP and PMBOK are trademarks of the Project Management Institute, Inc. PMI has not endorsed and did not participate in the development of this publication. PMI does not sponsor this publication and makes no warranty, guarantee or representation, expressed or implied as to the accuracy or content.

Every attempt has been made by OSP International LLC to ensure that the information presented in this publication is accurate and can serve as basic information in preparation for the PMI-ACP® certification exam. However, OSP International LLC accepts no legal responsibility for the content herein. This document or the course that it is part of should be used only as a reference and not as a replacement for officially published material. Using the information from this document does not guarantee that the reader will pass the PMI-ACP certification exam. No such guarantees or warranties are implied or expressed by OSP International LLC.

This publication is not intended for use as a source of legal, financial or human resource management advice.

Reproduction or translation of any part of this work without the permission of the Publisher is against the law. No part of this work may be reproduced or transmitted in any form or by any means, electronic, manual photocopying, recording, or by any information storage or retrieval system, without prior written permission by the Publisher.

Version:1.00

Date: 2013-August-15