

# Part 3: Knowledge, Skills, Methodologies, & Standards

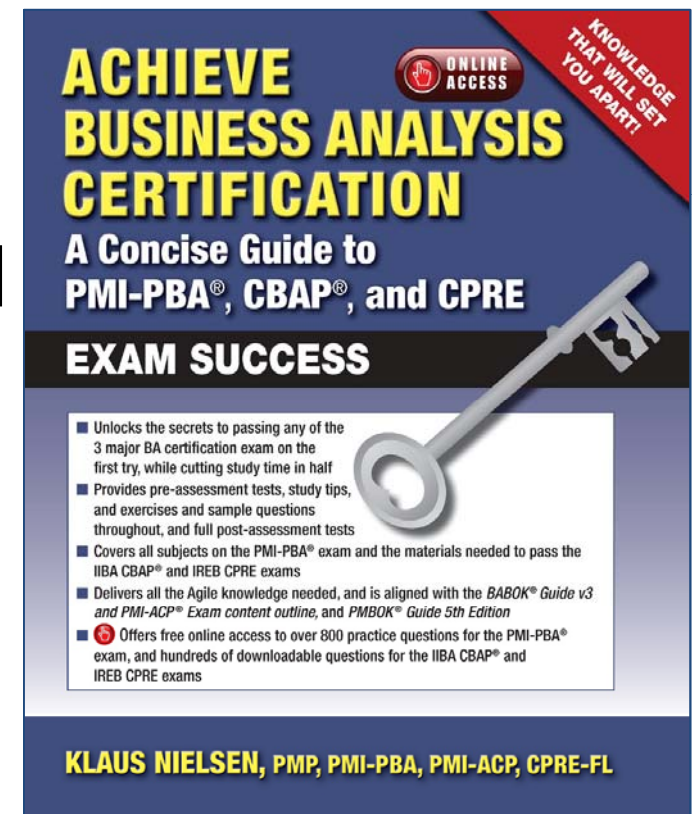
## Chapter 13: Agile Methodologies and Manifesto

1

# ACHIEVE BUSINESS ANALYSIS CERTIFICATION

## A Concise Guide to PMI-PBA®, CBAP®, and CPRE

By Klaus Nielsen



# Agile Software Development Manifesto

2

We are uncovering better ways of developing software by doing it and helping others do it too.

Through this process 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

# The Twelve Principles of Agile Software

**Table 13.1** The 12 principles of agile software

Principle	Practices	Requirements management
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Product backlog Whole team Incremental deployment Small releases Frequent delivery	Backlog management and the ongoing deliveries of identified, planned, and analyzed user stories.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.	Incremental deployment Small releases Sprint review Definition of done Acceptance tests	Small releases with incremental development of working software create a constant changing point of reference, which is the working software.
Working software is the primary measure of progress.	Incremental deployment Small releases Definition of <i>done</i> Acceptance tests	The measurement is working software as a result of collaboration with product owner and development team rather than individual input.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Sprint planning Planning game Product backlog Customer involvement	Backlog management with a constant increase of new user stories.
Business people and developers must work together daily throughout the project.	Real customer involvement Whole team Osmotic communication Daily Scrum	Day-to-day cooperation with the development team and product owner.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	Servant leadership Motivation	The business analyst must also act as servant leader to motivate the development team to get the work done. This is a team effort.

# The Twelve Principles of Agile Software (Continued)

4

**Table 13.1** The 12 principles of agile software

Principle	Practices	Requirements management
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Osmotic communication Servant leadership	The business analyst must be with the team and communicate face-to-face as much as possible as it is far more effective.
The best architectures, requirements, and designs emerge from self-organizing teams.	Test-driven development Refactoring Osmotic communication Servant leadership	Requirements are developed in close collaboration with the development team that is going to develop the solution.
Continuous attention to technical excellence and good design enhances agility.	Testing Sprint retrospective Pair programming Test-driven development Refactoring	The business analyst should use pair techniques to foster collaboration and take part in retrospective.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	Real customer involvement Motivation	The business analyst should not pressure the development team more than they can keep a constant speed.
Simplicity—the art of maximizing the amount of work not done—is essential.	Product backlog, refactoring Seeing waste	Backlog management, however, keeps it simple
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	Sprint retrospective Root cause analysis Seeing waste Value stream mapping	The business analyst must engage in activities to improve the process and collaboration.

# Declaration of Interdependence for Modern Management Adaption

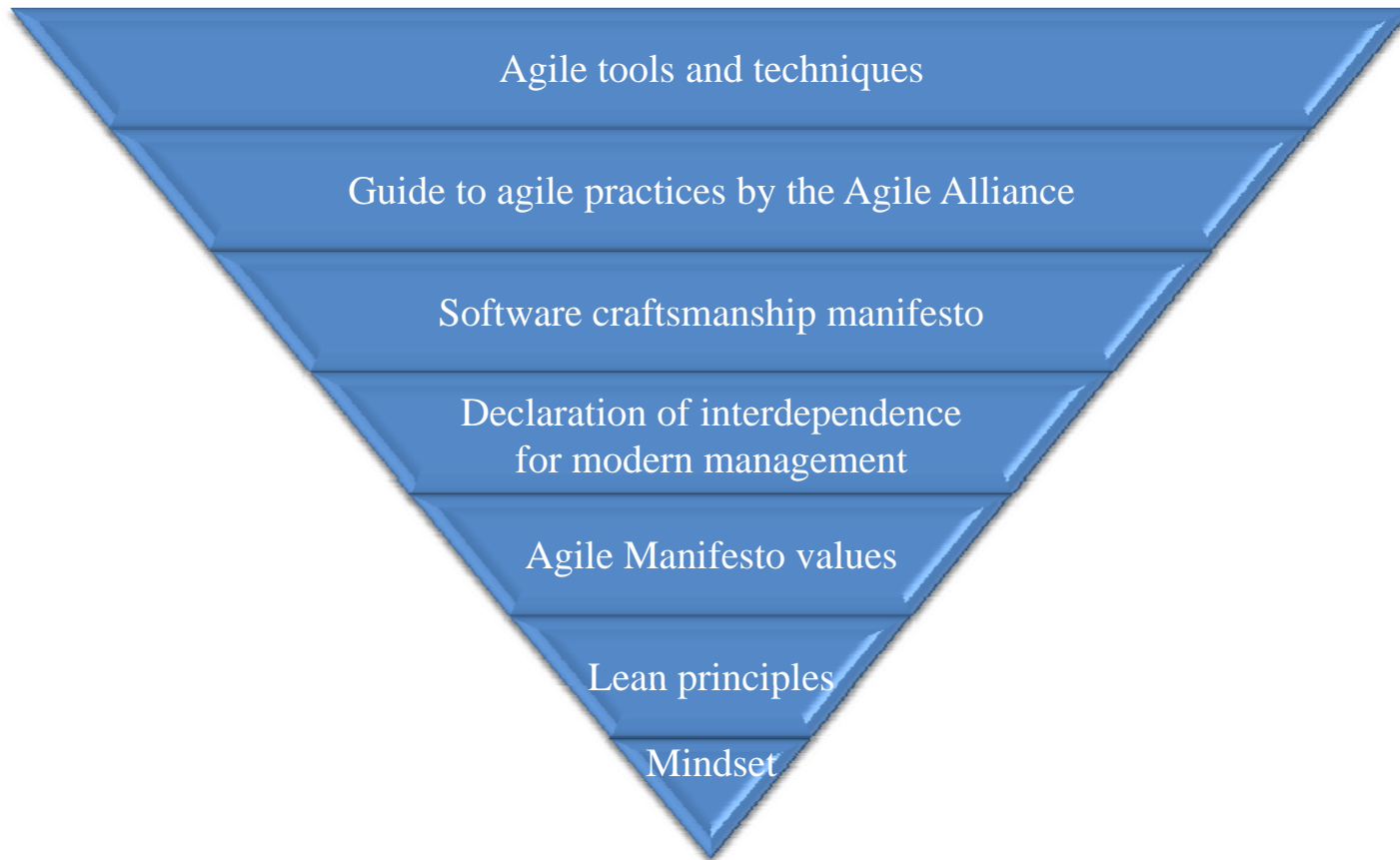
5

**Table 13.3** Declaration of interdependence for modern management adaptation

Accomplish this	by/through this	and	requirements management
Increased ROI	focusing on <i>flow of value</i> (e.g., not <i>tracking effort</i> )	continuous (one-piece) flow, preferably	Value-driven requirements
Reliable results	engaging customers in frequent interactions	shared ownership	Collaboration and communication with the development team and product owner daily
Unleash creativity and innovation	recognizing individual human beings as the ultimate source of value	creation of an environment where individual people can make a difference	The business analyst should consider the use of creative techniques and involvement
Manage uncertainty	iterations, anticipation, and adaptation	anticipation and adaptation (i.e., think ahead, plan, iter-ate, deliver, reflect, adapt)	PDCA
Improve effectiveness and reliability	situationally specific strategies, processes, and practices (i.e., no one answer, folks, get used to it)		The business analyst should embrace the agile practice and use retrospectives for improvements
Boost performance	group accountability for results (i.e., the whole group is singly accountable, no in-team blame)	shared responsibility for team effectiveness	Business analyst shares group accountability with the development team

# Building the Right Product

6



**Figure 13.1** Building the right product

# Pillars of Scrum

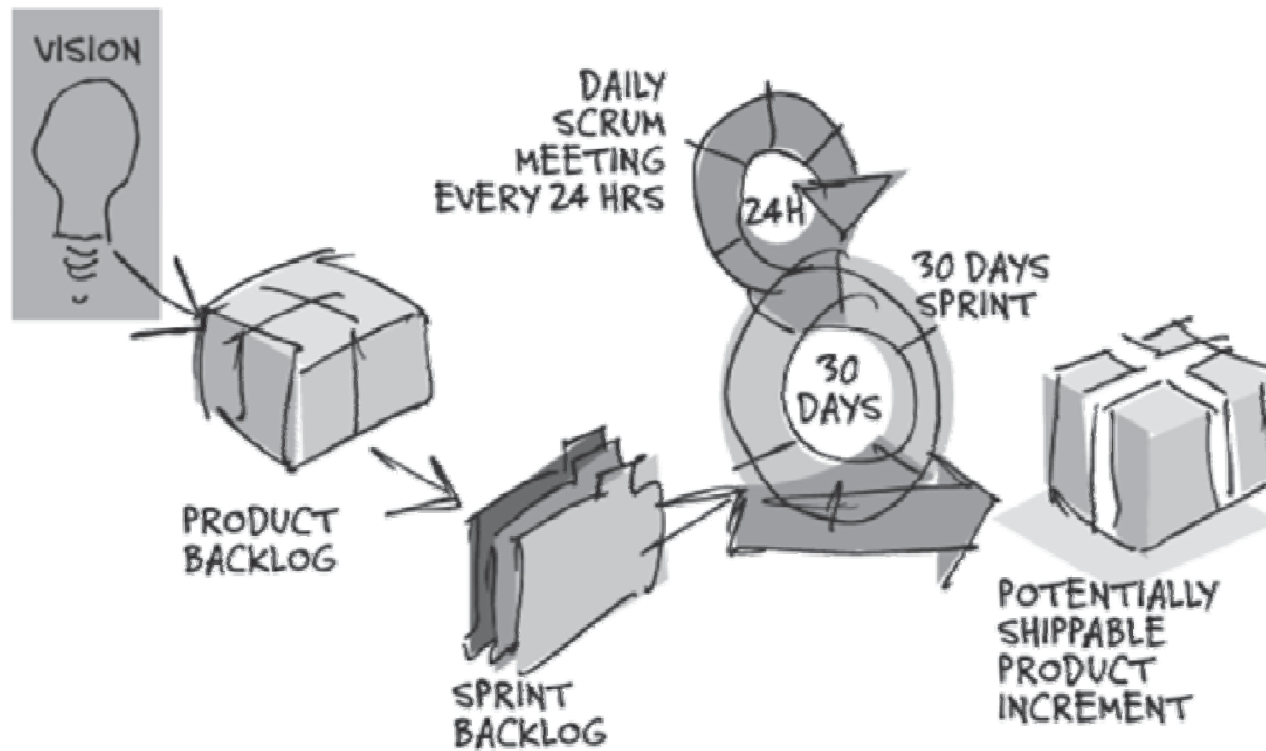
7

**Table 13.4** Pillars of Scrum

Theory of Scrum	Explanations	Requirement
Visibility	Visible outcome	Backlog management
Inspection	Timely checks Deviations or differences	Grooming of the backlog
Adaptation	Adjusting a process	Retrospective

# The Scrum Flow

8



**Figure 13.2** Components of Scrum adapted from the Agile Alliance in *I Am Agile* by Klaus Nielsen



# Key Practices of Scrum

**Table 13.5** Key practices of Scrum

Scrum methodology	Description
Product backlog (Artifact)	The prioritized list of all features and changes that have yet to be made to the system that were desired by multiple actors, such as customers, marketing and sales, and the project team. The product owner is responsible for maintaining the product backlog.
Sprint retrospective (Event)	Reflection on the process and room for improvements. Read Chapter 3 for more information on retrospectives.
Sprints (Event)	Sprints are 30 days in length; it is the procedure of adapting to the changing environmental variables (requirements, time, resources, knowledge, technology, etc.) and must result in a potentially shippable increment of software. The working tools of the team are sprint planning meetings, sprint backlog, and daily Scrum meetings.
Sprint planning meeting (Event)	A sprint planning meeting is first attended by the customers, users, management, product owner, and Scrum team where a set of goals and functionality are decided on. Next the Scrum Master and the Scrum team focus on how the product is implemented during the sprint.
Sprint backlog (Artifact)	The list of features that is currently assigned to a particular sprint. When all the features are completed, a new iteration of the system is delivered.
Daily Scrum (Event)	A daily meeting lasting approximately 15 minutes, which are organized to keep track of the progress of the Scrum team and address any obstacles faced by the team.
Team size	Development personnel are split into teams of up to seven people. A complete team should at least include a developer, quality assurance engineer, and a documenter.
Iteration length	While Schwaber originally suggested sprint lengths from 1 to 6 weeks, durations are commonly held at 4 weeks.
Support for distributed teams	While Scrum's prescription does not explicitly mention distributed teams or provide built-in support, a project may consist of multiple teams that could easily be distributed.
Definition of <i>done</i> (Artifact)	Everyone must agree on the meaning of <i>done</i> . Read Chapter 9 for more information on the definition of <i>done</i> .
System criticality	Scrum does not explicitly address the issue of criticality.

# Scrum Roles

**Table 13.6** Scrum roles

Role	Description
Product owner	Maximizes the value of the product, represents the stakeholders, is the <i>voice of the customer</i> , communicates the vision, prioritizes the product backlog, and is responsible for the outcome.
Development team/team member	Delivers the product and is self-organizing and determines how it will accomplish the work.
Scrum Master	Liaison between the product owner and the team, advises the product owner, supports the development team, and enforces rules and processes.

# Questions?

11



# Thank You

12

Thank you for using the slides!

As the demand for new professional certifications courses grow, at Global Business Development, a PMI® Registered Education Provider (R.E.P.) and SHRM training company, we do our best to meet this need.

We welcome you to visit our Global Business Development website at [www.gbd.dk](http://www.gbd.dk), for a complete listing of programs, products, high-profile clients, and speaking schedule.

Please contact me if you'd like to discuss my participation in any upcoming conferences, events, or in-house training.

I'd also love to hear from you if you have any feedback on my book, ideas on how to improve the material, or would like to share success you've had using the book.

Klaus Nielsen



**Global Business Development**  
*Knowledge that sets you part*  
[www.gbd.dk](http://www.gbd.dk)

Klaus Nielsen, PMP, PMI-PBA, PMI-ACP, CPRE-FL  
Global Business Development  
Email: [kni@itu.dk](mailto:kni@itu.dk) or connect at LinkedIn