PN-505-E
January 2018

Agile Project Management

Jaume Ribera

Abstract for promotional use only. Full version available at www.iesepublishing.com

Do Not Copy or Post

An Old "New" View on Project Management

It should be clear to anyone who has worked in project management in recent years that the agile methodology is not really a new concept, but it has been seen mostly as a methodology applicable to software development and related areas. It was not until recently that the leading project management organization the Project Management Institute published a document entitled Agile Practice Guide[1] to complement its traditional PMBOK Guide,[2] already in its sixth edition.[3]

Agile methodology was seen as an alternative to what was traditionally called waterfall methodology, even though different authors refer to it as "serial," "predictive" or "plan-driven." The agile methodology in the software industry was formalized with the publication of the Manifesto for Agile Software Development,[4] which states:

> 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.

[1] Agile Practice Guide, Project Management Institute, Inc., 2017. Note that this technical note incorporates some of the ideas of this document.

[2] A Guide to the Project Management Body of Knowledge, Project Management Institute, Inc., 2017.

[3] The first edition was published in 1996.

[4] Manifesto for Agile Software Development, available at agilemanifesto.org (accessed 12/12/17).

This technical note was prepared by Professor Jaume Ribera. January 2018.

Copyright © 2018 IESE. To order copies contact IESE Publishing via www.iesepublishing.com. Alternatively, write to publishing@iese.edu or call +34 932 536 558.
No part of this publication may be reproduced, stored in a retrieval system, used in a spreadsheet, or transmitted in any form or by any means - electronic, mechanical, photocopying, recording, or otherwise - without the permission of IESE.

Last edited: 30/8/23

Moreover, it develops the concept along 12 principles:

1. The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from every couple of weeks to every couple of months, with a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In practice, you can find these principles applied through many different practices or approaches, such as Scrum, Crystal, XP, Kanban, etc., all of them considered to be the lean thinking concepts of manufacturing adapted for project management.

The characteristics of a project that define whether the traditional methodology or the agile one is more suitable are its level of uncertainty and complexity.

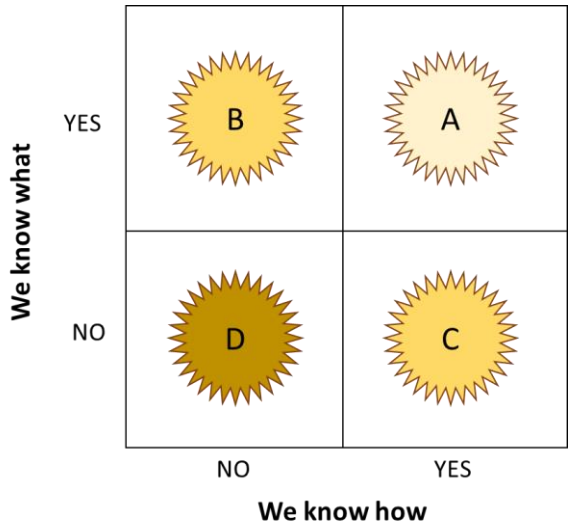
Two frameworks can help us understand these two characteristics. The first one comes from a book[5] by E. Obeng on strategy, which suggests classifying projects according to their uncertainty on *what* they plan to achieve and on *how* they are going to do it. Using these two dimensions, we could classify projects in four quadrants:

[5] Obeng, E., Putting Strategy to Work, Financial Times Pitman, 1995.

Abstract for promotional use only. Full version available at www.iesepublishing.com

Do Not Copy or Post

Figure 1



Projects in the A quadrant have a low level of uncertainty and can be planned with a lot of detail before starting their execution. This makes them suitable for a linear approach, with a lot of effort spent in the planning stage so that the execution would consist mostly of deploying the plan, not expecting major changes. Projects in the other three quadrants exhibit an important degree of uncertainty so will be much less stable and will involve a lot of learning during the project execution, which will require updating the plan quite often. For these projects, agile methodology will be more suitable.

Another interesting framework to describe the complexity of a project comes from the work of D.J. Snowden *et al.*,[6] which developed from the areas of knowledge management, cultural change and community dynamics to evolve towards organizational strategy by questioning assumptions such as order, rational choice and intentional capability. The central element of his approach is summarized in the Cynefin framework.[7]

Abstract for promotional use only. Full version available at www.iesepublishing.com

Do Not Copy or Post

[6] C.F. Kurtz and D.J. Snowden, The new dynamics of strategy: Sense-making in a complex and complicated world, IBM Systems Journal, 2003 & D.J. Snowden and M.E. Boone, A leader's framework for decision making, HBR, 2007.

[7] As described in Kurtz and Snowden's paper (cited above), Cynefin is a Welsh word whose literal translation into English as "habitat" or "place" fails to do it justice; it is more properly understood as the place of our multiple affiliations. We may not know all of these affiliations, but they profoundly influence what we are.